
ARMS: AUTONOMOUS ROBOTS FOR MILITARY SYSTEMS

A Survey of Collaborative Robotics Core
Technologies and Their Military Applications

CMU-RI-TR-01-16

Surya P. N. Singh and Scott M. Thayer



The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania

Author Contact Information:

Surya P. N. Singh: spns+@ri.cmu.edu

Scott M. Thayer: sthayer@ri.cmu.edu

Originally Prepared For:

DARPA Contracts Management Directorate
Defense Advanced Research Projects Agency
Washington, District of Columbia

Copyright (©) 2001 Carnegie Mellon University Robotics Institute

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of Carnegie Mellon University.

Abstract

The military forces of the future *will* use multi-agent robotic workforces for reconnaissance and surveillance, logistics and support, communications infrastructure, forward-deployed offensive operations, and as tactical decoys to conceal maneuver by manned assets. Towards this end, there is a clear and definite need for optimal, or pareto-optimal, multi-robot control strategies in the synthesis, design, implementation, and fielding of autonomous and semi-autonomous teams of combat robots for military systems. Proven coordination methods are essential to enable interactions with in dynamic and hostile environments, synchronized maneuvers, sensible and robust rules of engagement (ROE), and reliable field behavior.

This paper surveys the state-of-the art in autonomous multi-robot work systems and investigates the relative strengths and weaknesses of each of approach with respect to military applications. Further, it outlines guidance for future research directions and new capabilities that could provide the scalability, robustness, and flexibility necessary for the production of combat-worthy robotic forces. From this survey, we have concluded that current multi-robot coordination and cooperation architectures alone are not fully sufficient to handle the scale and scope of future combat – that new methods and design approaches (potentially ones currently under development) will be required to provide the foundation for the realization of military-grade robotic workforces needed for combat by 2010 FCS field deployment deadline.

Table of Contents

1	<i>Introduction</i>	1
1.1	Architectures	
1.2	Implementation Issues	
2	<i>Traditional AI Based Techniques</i>	3
2.1	Predicate Calculus Logic	
2.2	Process Algebras	
2.3	Path-finding/Deterministic Search	
2.4	Markov Machine Learning	
2.5	Neural Networks	
2.6	Genetic Algorithms	
2.7	Summary	
3	<i>Centralized Architectures</i>	8
3.1	Overview: “Let some ‘omniscient’ server dictate”	
3.2	Description	
3.3	Comparative Advantages/Disadvantages of Centralized Control	
4	<i>Deliberative / Hierarchical Control Model</i>	11
4.1	Overview: “Think through the situation fully at various levels, then respond”	
4.2	Description	
4.3	Comparative Advantages/Disadvantages of Deliberative/Hierarchical Control	
5	<i>Reactive Architectures</i>	14
5.1	Overview: “Think maybe, use actions and directly respond to the stimulus”	
5.2	Reactive Control Theory	
5.3	Comparative Advantages/Disadvantages of Reactive Architectures	
6	<i>Behavioral Architectures</i>	17
6.1	Overview: “Let your available actions dictate your thinking and options”	
6.2	Subsumption Assumption	
6.3	Description of behavior-based robots	
6.4	Motor Schemas	
6.5	Multi-Robot Teaming	
6.6	Comparative Advantages/Disadvantages of the Behavior Architecture	
7	<i>Hybrid</i>	23
7.1	Overview: “Combine thinking and action (independently) in parallel”	
7.2	Hybrid Model Description	
7.3	AuRA	
7.4	Servo-Subsumption-Symbolic (SSS)	
7.5	Multi-Valued Logic	
7.6	Comparative Advantages/Disadvantages of Hybrid Architectures	
8	<i>Free Market Architecture</i>	27
8.1	Overview: “Trade and act to maximize your profit”	
8.2	Microeconomic Model	
8.3	Free Market System Employed	
8.4	Economy Based Robotic Coordination	
8.5	Comparative Advantages/Disadvantages of Free-Market Architectures	

9	<i>Immunity Based</i>	31
9.1	Overview – “Specific, dynamic brute force assemblages”	
9.2	Immune System ‘In a nutshell’	
9.3	Artificial Immune System Model and Immunity Architecture	
9.4	IDDARA and other Architecture Implementations	
9.5	Comparative Advantages/Disadvantages of Immunity Based Architectures	
10	<i>Technology Summary for Future Systems/Applications</i>	36
10.1	Summary of Robot Workforce Issues	
10.2	Comparisons	
11	<i>Military / Defense Implementations</i>	40
11.1	Overview	
11.2	Robot Team Software Developments and Technologies	
11.3	Robot Team Hardware Developments and Technologies	
11.4	DARPA TMR	
11.5	Summary	
12	<i>Potential Impact and Bearing of Collaborative Robotics</i>	49
12.1	Overview	
12.2	Coordination Level Description and Illustration	
12.3	DARPA / Army Impact	
12.4	Sample Collaborative Applications	
12.5	Summary	
13	<i>Conclusions</i>	58
14	<i>Acknowledgments</i>	60
15	<i>References</i>	60
16	<i>Appendix 1: Current DARPA Projects Impacted by Collaborative Robotics</i>	63

Table of Figures

Figure 1: Simple example combinations of atomic operations	5
Figure 2: Sample FSA for competitive robot task	6
Figure 3: Centralized architecture model	8
Figure 4: Descriptive overview of the centralized architecture	10
Figure 5: Deliberative / Hierarchical Control Model overview	11
Figure 6: Sample Deliberative/Hierarchical architecture	12
Figure 7: Sensor-Perception-Action Node in 4-D/RCS	13
Figure 8: Robot control spectrum	15
Figure 9: Reactive architecture overview	16
Figure 10: Behavioral Architecture overview	17
Figure 11: Subsumption's paradigm shift	18
Figure 12: Behavior based vs. traditional architectures	19
Figure 13: Robot formations	22
Figure 14: Macro level component structure for AuRA [12]	24
Figure 15: Reactive layer of the AuRA model	25
Figure 16: The TJ robot	26
Figure 17: Illustration of Free Market Architecture bidding/negotiation	29
Figure 18: Initial assignments and final tours for 2 robots and 8 cities	30
Figure 19: Free Market Architecture final tours for four robots and 20 cities	30
Figure 20: Immune system response structure	32
Figure 21: Model of human innate immunity	32
Figure 22: IDARA Software Architecture	34
Figure 23: IDARA in a Noisy Environments	34
Figure 24: IDARA Energy Consumed	35
Figure 25: Spectrum comparing the prime application areas for various architectures ..	39
Figure 26: JAUGS General Structure	41
Figure 27: MDARS (External) and Demo III UGV	43
Figure 28: MDARS user environment	44
Figure 29: BUGS and RECORM robots in the field	45
Figure 30: Scout robots	46
Figure 31: Ranger robot (and launcher)	47
Figure 32: Illustration of a hypothetical FCS "unit cell"	49
Figure 33: Envisioned FCS Coordination Echelons	50
Figure 34: Comparative Visitation Maps	53
Figure 35: Hybrid architecture and mapping	53
Figure 36: Distributed Intermittent Jamming	56
Figure 37: Collaborative robotics as a force multiplier and catalyst	58

List of Acronyms

4-D/RCS.....	Four Dimensional/Real-time Control System	MIT.....	Massachusetts Institute of Technology
A*	A-Star search	MLE.....	Maximum likelihood estimation
AI.....	Artificial Intelligence	MOUT	Military Operations in Urban Terrain
AMRDEC..	Aviation and Missile Research, Development, and Engineering Center	MRHA.....	Multiple Robot Host Architecture
APL.....	Applied Physics Laboratory	MVL.....	Multi-valued Logic
AuRA.....	Autonomous Robot Architecture	NASA.....	National Aeronautics & Space Administration
BBR	Behavior-based robotic	NASREM .	NASA/NIST standard reference model
BG.....	Behavior Generation	NIST	National Institute of Standards and Technology
BUGS	Basic UXO Gathering System	NP	Nondeterministic Polynomial
C2	Command and Control	NSA.....	National Security Agency
C4ISR.....	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance	OTS	Off the Shelf (technology)
CMU/RI.....	Carnegie Mellon University / Robotics Institute	ORNL.....	Oak Ridge National Laboratory (DOE)
DARPA	Defense Advanced Research Projects Agency	N-PGS	Naval - Post Graduate School
DIJ	Distributed Intermittent Jamming	POMDPs ...	Partially Observable Markov Decision Processes
DoD	Department of Defense	RAD	Rapid Application Development
DOE.....	Department of Energy	R & S.....	Rangers and Scouts (project)
EM.....	Expectation-Maximization	ROE.....	Rules of Engagement
EOD.....	Explosives ordnance	SMART	Sandia's Modular Architecture for Robotics and Teleoperation
FCS.....	Future Combat Systems	SP	Sensory Processing
FSA.....	Finite State Acceptor	SRI	Stanford Research Institute (International)
FSMs.....	Finite State Machines	SSS	Servo-Subsumption-Symbolic
GAs.....	Genetic Algorithms	TCP/IP.....	Transmission Control Protocol Internet Protocol
GDRS	General Dynamics Robotics Systems	TSP.....	Traveling Salesman Problem
GIT.....	Georgia Institute of Technology	UAV	Unmanned Air Vehicle
GPS.....	Global Positioning System	UGV	Unmanned Ground Vehicles
IBA	Immunity Based Architectures	UXO	Unexploded Ordnance
IBM.....	International Business Machines	VaMoRs	Versuchsfahrzeug futornome Mobiliaund Rechnersehen (an autonomous 5-ton van)
IDDARA....	Immunology Derived Distributed Autonomous Robotics Architecture	VJ	Value Judgment
JAUGS.....	Joint Architecture for Unmanned Ground Systems	WM.....	World Modeling
JPL.....	NASA Jet Propulsion Laboratory	XUV	Experimental Unmanned Vehicle
JRP	Joint Robotics Program		
MDARS.....	Mobile Detection Assessment Response System		
MEMS	MicroElectroMechanical Systems		

1 Introduction

Effective and reliable coordination has been and will continue to be an essential element in the success of combat missions with both unmanned and manned assets. It comes as no surprise that teams of robots generally can accomplish tasks more quickly and effectively than a single agent working independently on a task. Future combat will require multiple layers of interactions between warfighters and the autonomous and semi-autonomous systems that will be an indispensable part of the future force. Thus, there is a clear and definite need for mature design and configuration strategies for multi-robot work systems [79].

Robot coordination strategies are the governing principles that impel robot workforces to maximize their impact by working as a integrated team instead of as loose aggregate of individuals. These multi-robot strategies act as a keystone that directly affects (and to some extent limits) the performance of the assemblage. Just as building's architecture defines a structure before construction commences, so does the central architecture define the operation, scope, and utility of the coordination and control system.

1.1 Architectures

Coordination architectures are of key importance to robotics workforces because they serve as the foundation for the resulting distributed robotics system. The coordination architecture is the general form by which communication and successive member interactions are performed [9]. Architecture design not only specifies the underlying foundation/structure for a robot's control and coordination, but also imposes constraints on the controller and its solutions [49].

Robotic coordination (a central part of these architectures) can be a double-edged sword because it defines the medium and method by which cooperation and its synergistic effects take place. Therefore, it can aid certain operations while hindering performance or, in certain scenarios, potentially propagate errors. While important to any implementation, the servo-level control and mechanical details of the sensors/actuators on an individual robot are assumed to be independent and decoupled from the more general coordination system(s) associated with the individual robot. It is generally assumed that in a robot this control is moved into the background and, in the case of teams of robots, a secondary mechanism serves to guide the actions of these robots [76].

As described in detail in later sections, there are a variety of methods to approaching the robot coordination problem. As may be expected, these architectures have particular characteristics and advantages that make them individually suitable for certain classes of coordination problems; however, there is no general, best solution. Still a vigorous area of research, these architectures have been initially researched, developed, and demonstrated and are succinctly described as follows [53]:

- Centralized Coordination – “Let some ‘omniscient’ server dictate”
- Deliberative Control – “Think through the situation fully at various levels, then respond”
- Reactive Control – “Think maybe, use actions and directly respond to the stimulus”
- Behavior-Based – “Let your available actions dictate your thinking and options”
- Hybrid Control – “Combine thinking and action (independently) in parallel”
- Free Market – “Trade and act to maximize your profit”
- Immune Complex Interaction – “Specific, dynamic assemblages with stochastic actions”

The applicability of these approaches is governed by the environment and some generally unavoidable assumptions and generalizations that are associated with complex system design. In particular these include the following [53]:

- Thinking is generally slow and improves with time
- Actions need to be as fast as possible, especially reactions to perceived threats
- Planning needs thinking, but avoids looking ahead and avoiding bad actions
- Either extreme is hazardous. For example, thinking too long [only to be attacked by an on-coming (but known) threat] or acting in an irrational manner (e.g., falling of a cliff or running into a hazardous situation).
- Thinking needs actual or real time information. Reaction results in changes to information in addition to increased noise. Further, because the environment is changing, the robot's thinking may result in actions for an expired state.

The selection of the coordination method and its design is greatly influenced by these trade-offs and their consequences on the performance of the task. Further, there are trade-offs in redundancy and the underlying computational algorithms being used. Finally, implementation and serviceability of the coordination method and the robots is also an influencing factor. The later sections detail how the aforementioned approaches best address these trade-offs and thus suggest the best use of certain approaches [60].

1.2 Implementation Issues

Autonomous multi-robot coordination research investigates the use of various strategies and approaches to enable efficient and meaningful interactions and exchanges between robotic systems. Classic robot control strategies have been derived from Artificial Intelligence, Biological, and Sociological research methods. Newer algorithms are often based from epistemology methods (e.g., Albus's Intelligence Model), biological (e.g., behavior-based strategies based on the entomology of ant colony activities) and from natural science [3],[14]. However, many of these research methods are designed in ideal environments and experimental scales (less than twenty agents, often without large variance). Future Combat System (FCS) robots and systems, on the other hand, will be deployed on a large-scale in difficult, non-ideal, hostile, and combat environments [79].

Some of the most significant research issues affecting FCS are scaling issues, sub-optimal control strategies, and heterogeneous systems operation. This is an ongoing area of research which is being developed on several fronts ranging from extensions of computer science to research in control systems. Thus, the following research issues are important for FCS applications [79]:

1. **Order of expansion** – Provably optimal coordination architectures exhibit an exponential growth in computational complexity. , e.g. centralized coordination systems.
2. **Scaling** – Very large-scale command and control (that may consist of a spectrum of robotic platforms) have not been modeled fully. The key issue in scaling is the order of computation necessary to converge to a sufficiently acceptable solution and in a manner that does not flood the communication or computational sub-systems.
3. **Intelligent Adversaries** – An opposing force will exploit any “obvious” patterns in the control and coordination strategy; therefore, the coordination system cannot be rigid or ‘mechanical’ in scope. The system must thwart countermeasures and be capable of overwhelming a numerically superior force.

2 Traditional AI Based Techniques

Classic Artificial Intelligence (AI) techniques present methods for determining and processing logical inference from perceived datasets and then systematically applying this knowledge to derive optimal actions or responses to the environment. Classical AI methods are utilized to some extent by almost all the architectures under consideration; thus, a common issue with almost all of the architectures is the selection and design of artificial intelligence algorithms.

Classical AI algorithms can often be extended to cope with the issues of multi-robot coordination directly. However, these implementations are generally symbol-based and require a good model of the environment and may suffer from the symbol-grounding problem [18]. The coordination performance is constrained by the assumptions or restrictions associated with the underlying algorithm and therefore these methods generally yield coordination schemes that are inflexible outside the constraints of their underlying model (i.e., Bayesian, Gaussian, Predicate Calculus) [65].

In addition to logical inference, statistical and pattern classification methods have been used successfully in the robot coordination problem. Often exploited as underlying algorithms by more general architectures for coordination, these methods include regression, maximum likelihood estimation (MLE), gradient decent, batch perception algorithm, Gaussian perceptrons, sigmoid functions, and neural networks [58]. As detailed in the following sections, these fundamental AI methods range from symbolic methods (e.g., Predicate Calculus and Process Algebra) to data extrapolation methods such as Partially Observable Markov Decision Processes (POMDPs) and Neural Networks.

2.1 Predicate Calculus Logic

Predicate calculus logic, a fundamental basis of symbolic AI, allows for the solution of generalizations from given logic statements. Thus, it is possible for a robot team to coordinate itself by updating its “knowledge base” (i.e., a representation of logical facts about the world) corresponding to changes in the environment. The key issue with this method is that the system collapses if an inconsistency enters the knowledge base. Further, symbolic AI systems often cannot easily adapt to complex situations (i.e., where an input may need to be resolved via two or more different actions).

2.2 Process Algebras

A second approach in traditional AI is to extend the use of symbolic logic and its operations so that they can be manipulated, combined, and treated mathematically. In a nutshell, Process Algebra Theory combines this principal notion along with axiomatic theories (e.g., Algebra of Communicating processes (ACP) to yield an fundamental model that builds an algebra around these process and provides a strong method for manipulating and coordinating process execution [11].

This strong mathematical foundation not only serves to abstract process step coordination, but also allows for the concrete expressions and specification of complex processes and their interactions. This has several obvious, and some subtle, implications for robotics and, in particular, robot coordination, such as:

- Process actions can be modeled and treated as compositions simple atomic steps
- The interactions between processes and their relative execution can be integrated into the development of the algebra, allowing for a compact representation of the execution model.

- Non-deterministic, concurrent processes can be easily modeled [25].
- Optimal or near optimal solutions can be found given that optimality is a function of the processes involved (i.e., factors effecting the optimal solution can be modeled).
- Coordination can be viewed as the solution of these steps at critical junctures [11].
- Resulting process are such that the notion of correctness of implementation with respect to a specification is almost automatic being simply that of showing equivalence [25].

2.2.1 Basic Constructs

A key feature of process algebras is not only that they are able to represent complex motions as functions of atomic steps, but that these actions are combined by a relatively small set of definite basic operators and logical processes. These are briefly summarized as follows [11]:

- Combination: The operators $+$ (or) and $\cdot$ (and) are the basic combining operators. For example, the line $(a+b) \cdot c$ can be interpreted as first choose process a or b and, second, perform action c .
- Deadlock: The δ operator represents stagnation. This is different from a stop as the algebra wants to perform a proper action but it cannot.
- Merge: represented as $||$ or $(\downarrow)$ for left merge) this signifies parallel composition. That is the notion of doing either of the actions and then performing first the remaining actions after this. For example, $x || y$ may perform action y first and then would do complete action x .
- Fixed points: Methods for the representation and differentiation between finite and infinite processes.
- Communication: Symbolized $|$, this represents the concept that some processes may need an action in another processes for expression. This concept is of particular importance in coordination as it encapsulates the idea of execution order and allows for the integration of specificity. For example, in the case of a robot relay (i.e., of a part or even of an information packet) before the exchange can occur both robots need to be present and in a configuration where they can recognize each other and perform the necessary exchange.
- Projection theorems: These theorems are analogous to the fundamental theorems of standard arithmetic algebra (e.g, $a \cdot (b+c) = a \cdot b + a \cdot c$).

2.2.2 Sample Applications

Figure 1 shows (via process graphs) the application of the aforementioned theory to simple process linking cases. This same logic can be combined to yield far more complex results. As such, process algebras methods are amenable to several applications in process control and coordination. Some of the multi-robot coordination aspects that benefit from this technology include [11]:

- Log-time order sorting of operations and processes.
- Offers modular computation and execution of processes.
- Operation communication based on ordering between formulas (i.e., efficient method of data communication).
- Provable correctness with respect to failure semantics in the ordering of systolic systems (i.e., systems built-up from arrays of “cells”).

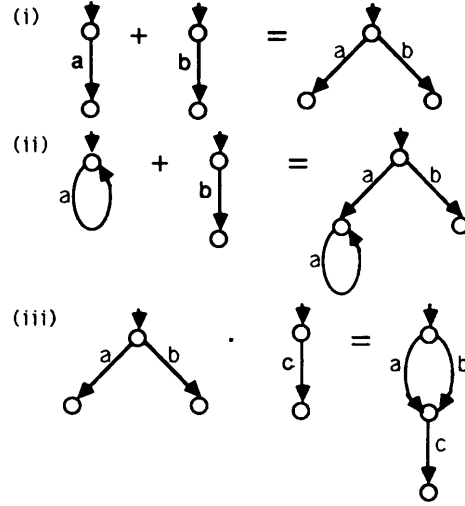


Figure 1: Simple example combinations of atomic operations

Where: (i) simple addition $[a+b]$ (ii) addition with deadlock $[\delta a+b]$ (iii) combination $[(a+b) \cdot c]$ [11]

2.3 Path-finding/Deterministic Search

While not directly applicable to the coordination problem, many of the coordination methods make use of deterministic search methods to solve the optimal action problem (i.e., determine which action(s) available should be executed). In particular, the methods are estimating the path with the lowest associated cost, given the know information/model of the environment. Depth First Search and A* (with a good [i.e., optimistic] heuristic function) are popular algorithms. In a dynamic environment, D* is often used for path planning evaluation calculations [77].

A special form of the search problem when structural constraints are known (i.e., from the world model) is constraint satisfaction propagation [63]. In this case the structural constraints are propagated via a backtracking search to constrain the search space and reduce the sample set. This subset is then processed to find an optimal path. For example, in minesweeper (the game) the structural properties of the game are well known and thus given some perceived information the search (and thus solution) set can be rapidly and easily restricted [58].

2.4 Markov Machine Learning

Machine Learning is a rich and deep field within computer science that gives many insights on how to structure, design, and build algorithms that enables individual and team learning and application of this updated knowledge to a new, uncertain stimulus. Tomes have been written on this subject; however, [57] suggests two principal techniques from this area on how best to proceed, namely: inductive, analytical methods (e.g., Markov model trees) and self-directed autonomous methods (e.g., POMDPs). Further, the coordination task can be modeled as an unsupervised learning task, which can be analyzed using Bayesian classifiers or Expectation-Maximization (EM) methods to generate and refine a model of the environment. The common issues with these methods for coordination are that they can be computationally expensive, especially if a random model of the environment is being used as a basis for solving the problem [65].

2.4.1 Markov Models

The Markov property is the fundamental and key assumption underlying Markov models and Markov decision processes. This property states that probability for future events is only a function of the present state and not the previous (historical) states that have lead up to it [58]. This greatly simplifies the calculation of probability of future events (often symbolized as α) by allowing for cancellation of previous steps (see Equation set 1). This then allows for the calculation of future probabilities by iterations of Bayes rule for the known probabilities of the future states [81].

$$\text{Belief}(x_t) = \alpha(t) = p(x_t | o_t, a_{t-1}, a_{t-2}, \dots, o_o) \quad (1.1)$$

Applying the Markov assumption gives:

$$\text{Belief}(x_t) = \alpha(t) = p(x_t | o_t) \quad (1.2)$$

Markov models have become increasingly popular due to their wide range of applications and their rich mathematical foundation [64]. Some examples of the applications of these methods in coordination include coordinated elevator scheduling and modeling tools for macroeconomic operations between multiple parties [58]. Finally, it is often useful to visualize these processes via finite state acceptor (FSA) diagrams, an example of which is shown in Figure 2 [57].

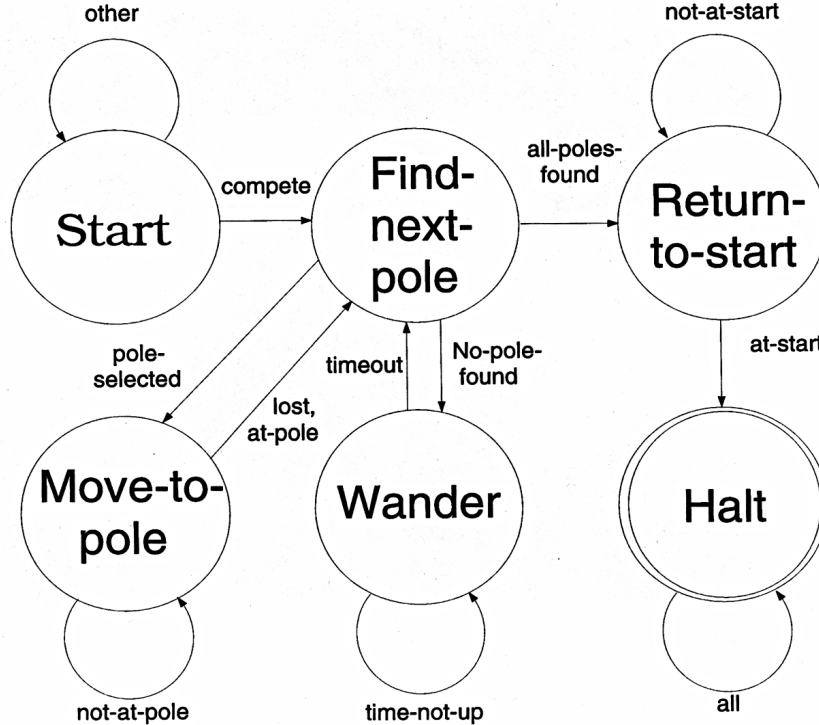


Figure 2: Sample FSA for competitive robot task [9]

2.4.2 POMDPs

The second key assumption associated with the use of Markov processes (and classic machine learning techniques in general) is the existence of a partial model (approximate or incomplete) of the environment. That is, in processing Markov decision processes we assume that the options between states and their associated transition probabilities are known. In an uncertain and hostile environment, however, this assumption can lead to undesirable results as the modeled and true environment may differ significantly [23].

POMDPs are a class of Markov model which account for the uncertainty seen in the data. Further, methods for robot coordination based on POMDPs can leverage the strong and extensive mathematical and algorithmic information about these methods for use in decision making. Although computationally quite expensive, recent research is providing techniques that are simpler and more efficient. This allows for optimal decisions to be found for a given uncertainty in both the model and perceived data being collected about the environment [81].

2.5 Neural Networks

Often used as the basis for action/behavior adaptation in robot coordination tasks, neural networks represent a popular form of the AI concept of reinforcement learning [56]. Modern neural networks make fairly extensive use of backpropagation algorithms as a means for allowing for both offline (i.e., trained by example in a “laboratory”) and online training (trained by experiences encountered in the real environment) [9]. These methods have also been applied to the robot coordination problem by either having individual neural networks on each robot and having them learn how to distribute and share tasks or by treating the task as one to be solved by a “global” neural network and then using each robot as a node in the system. In the latter case the network is trained in a controlled environment and the “gains” and/or performance variables of each robot are set by the central (off-line) controller [57].

2.6 Genetic Algorithms

Genetic algorithms (GAs) are often used as a method for performing stochastic searches as a means for approximating computationally hard (exponential order) problems. In a nutshell, GAs approximate the value of the global solution by encoding the problem as a “bitstring” and starting with a random (stochastic) population [65]. Then, based on some knowledge of the problem, the initial population is selected via a fitness metric (usually a costs minimization test). The fittest strings then are recombined to generate a second generation of the population. This is repeated until an acceptable solution is found or until the time allotted has expired. In order to add variation, GAs use crossover (a random shuffle – parts of the bitstring are spliced and mixed at a random point), mutation (a random walk where variation is added with some small probability), and recombination (the bitstring is disrupted) [58].

In addition to an approximation method for stochastic search and other similar hard problems, this method has been used as a method for robot coordination. In particular, this has been implemented as an alternative to the hierarchical control system (detailed in section 4). The coordination is achieved by applying the GA twice, once to determine an optimal action path/tour and then a second time to coordinate the robots (i.e., remove collisions, prevent dead-lock, etc.) [34].

2.7 Summary

By combining the aforementioned Artificial Intelligence techniques and methods, it is possible to yield a variety of traditional robot control systems and to extend these systems to the multi-robot domain. However, new architectures for multi-robot command and control often leverage these methods as tools and primitives that are combined and integrated as per the philosophy of the architecture to yield the methods described in the following seven sections.

3 Centralized Architectures

3.1 Overview: “Let some ‘omniscient’ server dictate”

Centralized architectures are generally the first method selected when trying to implement small team approaches to robotic problems. In this case, a central server (which may be one of the robots) has knowledge of the given requirements and collects all perceived data from the other robots in the field. This data is then processed using one of the aforementioned AI algorithms to find an “optimal” solution. This solution is then sent/served to the robots in the field for execution and the cycle repeats (see also Figure 3).

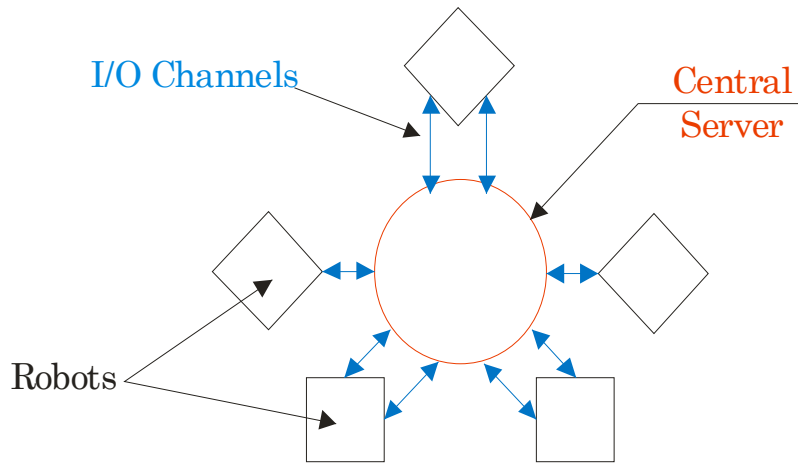


Figure 3: Centralized architecture model

3.2 Description

The centralized architecture is one of the most popular methods for robot coordination. As suggested by its name, this method uses a centralized system/robot to coordinate and communicate the actions of the ‘child’ robots. Often this central unit is implemented as a server which uses standard client/server [75] communication methods to update and communicate between each of the robots.

In a typical system the centralized module uses has a collection of symbolic AI algorithms that processes model variables and symbols, which are generated from the incoming sensor data, current state, previously collected data, and knowledge of the global environment into the position and actions map for the rest of the robots. This map along with the sensed model of the environment are then used as the basis by a second computing module which runs an optimization method in order to determine the best future action for the robots [30] (see also Figure 4).

More advanced versions of this coordination technique use algorithms that include forward prediction, probabilistic methods and/or machine learning techniques to best predict the rewards for future action of the robot given certain assumptions/constraints of the global environment [57]. Based upon these estimates, the technique usually selects actions so that the reward for the team is maximized for a particular time-horizon [58].

This implementation “ease” of centralized coordination often comes with a handful of complexities that are only negligible for small teams (i.e., a few robots) operating in ideal environments. The key problem with centralized coordination methods is that the optimal group computation involves exponentially complex algorithms that limit an autonomous system’s response rate. Secondly, these approaches are quite vulnerable in a hostile environment. For example, coordination could be lost (temporarily in a leadership switching configuration and at worst permanently) if the principal member of the team is destroyed. The coordinated control methods also require nearly ideal environments as the whole team suffer from noisy/invalid data (even when statistical techniques such as EM are applied) and that the data is basically “steady-state” (i.e., it is not changing significantly during the processing step). Finally, these approaches assume nearly ideal information flow where data can be transferred to/from the single key point and back rapidly, reliably, and securely [76].

The use of this method is normally acceptable only as a first-approximation to the coordination problem, but it does not fulfill the needs of a large-scale autonomous force of field robots in hazardous and hostile environments. It is fundamentally restricted in its scope because it is unable to leverage the parallel and team nature of the robotic workforce. Finally, its central vulnerability limits the robustness and reliability of this approach, especially without the addition of layers or methods of redundancy [76].

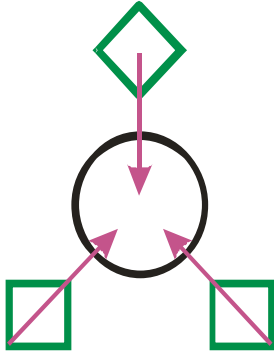
3.3 Comparative Advantages/Disadvantages of Centralized Control

Advantages:

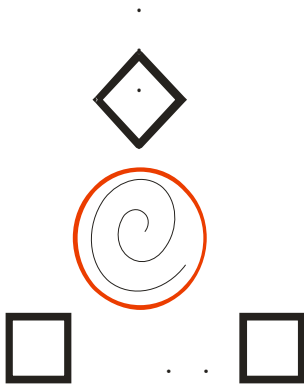
- **Implementation Simplicity** – The main advantage for centralized robotic coordination systems is their general level of simplicity in implementation.
- **Ease of Development** – Since the key decision maker in the system is the central system/robot, this method does not require an extensive level of development. Thus development and testing attention can be focused on this one agent instead of a distributed array of agents operating in particular fashion.
- **Global Memory** – There is a common global memory system, thus allowing for easier sharing and processing based on a common understanding of the data.
- **Optimality** – For small teams in constrained environments provably optimal methods can be applied in a reasonable manner.

Disadvantages:

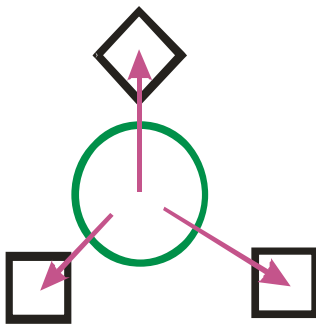
- **Experimental Computational Order** – This technique is order NP (nondeterministic polynomial), that is, the computational complexity grows exponentially as the order of the system increases and involves the algorithm guessing and verifying the solution.
- **Limited Scalability** – This technique simply does not scale. It needs a great deal of computational power to optimally generate a rapid response, especially in an environment where it is expected to serve appropriate commands to a robot that is in a wait state until it receives these commands [30].
- **Fails in Noisy Operational Environments** – The communication and sensor environment need to be nearly optimal for the central server to calculate and deliver optimal or nearly optimal solutions.
- **Critically Vulnerable** – If the central planning unit fails, leadership must be passed to a new leader (assuming one is available) or the entire team could be disabled [30].



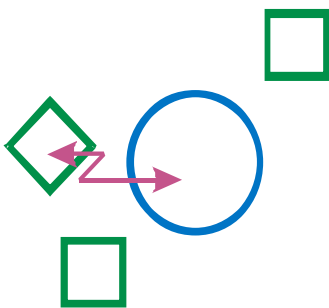
Step 1:
The robots send data perceived about the world (often containing noise and without much filtering) to the central server/leader. This can be done in daisy chain or simultaneously depending on the implementation.



Step 2:
The central controller processes the received data to determine the optimal policy. The optimal policy can be found via a variety of methods ranging from linear programming to neural networks. It should be noted that the world and/or robots may change their state during this processing.



Step 3:
The server then sends the computed solution back to the robots. Depending on the algorithm and the time-constraints this solution may or may not be optimal.



Step 4:
The robots then execute the instructions received from the central coordinator (conductor). It is possible that the robots may ask for clarification with the server.

Figure 4: Descriptive overview of the centralized architecture

4 Deliberative / Hierarchical Control Model

4.1 Overview: “Think through the situation fully at various levels, then respond”

In deliberative control the robot takes all of the available sensory information, and all of the internally stored knowledge it has, and it thinks (“reasons,” “cogitates,” etc.) about it in order to create a plan of action (see Figure 5). To do so, the robot must search through potentially all possible plans until it finds one that will do the job. This requires the robot to look ahead and be predictive, which is an order NP (i.e., Nondeterministic Polynomial and thus intractable for large cases) computationally hard problem. This can take an extremely long time, which is why if the robot must react quickly, it may not be practical for difficult environments. However, if there is time, this allows the robot to act quite strategically and optimally [53].

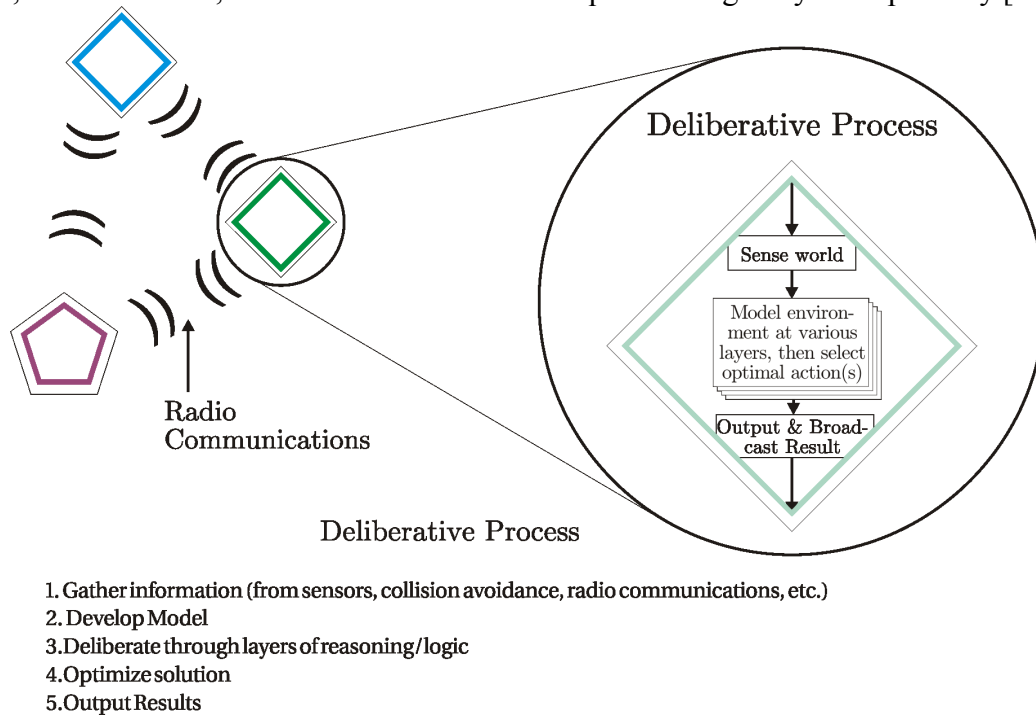


Figure 5: Deliberative / Hierarchical Control Model overview

4.2 Description

A commonly used method is the use of deliberative/hierarchical control. This form of control principally uses centralized architectures and is derived from deliberative reasoning methods for coordination and control. The deliberative methods, in turn, are derived from the symbolic intelligence/intelligent control community. As such, they also share properties/characteristics with classical symbol-based AI methods [9]. These include:

1. Having a clearly hierarchical structure that has been subdivided based on a given criteria (e.g., conditional independence to yield a Bayesian hierarchy).
2. Very structured communication orders.
3. Higher levels commanding (i.e., providing requirements and measuring their response) to lower levels of the hierarchy, which then govern lower (subordinate) levels.
4. Scope (time and space) decreases as a function of the level in the hierarchy being considered.

Figure 6 shows the NASREM (NASA/NIST standard reference model) architecture which is based in Albus methods. In this model a centralized system applies logic to perceived inputs and then directs action, but action and perception are decoupled. This method accomplishes this through a world model and some presumptions about the noise. Consequently, the system becomes very dependent on the model of the world (e.g., Kalman filtering of noise is a function of the noise model).

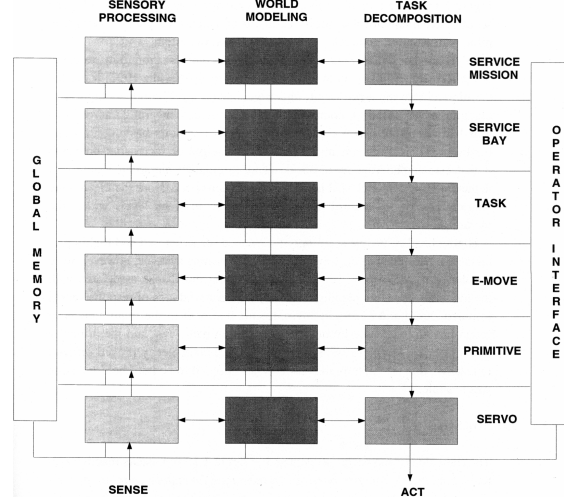


Figure 6: Sample Deliberative/Hierarchical architecture - the sensory, modeling, and task value-judgment steps are separate for each layer and communicate with other steps and levels through a global memory [9]

The NASREM architecture has also served as the reference and starting point for the 4-D/RCS (Four Dimensional/Real-time Control System) autonomous military control standard [4]. As with NASREM, control and coordination in 4-D/RCS is modeled as a set of hierarchical layers. However, RCS represents a stronger, more refined architectural embodiment of the concepts outlined Albus's model of intelligence [9], [4].

The 4-D/RCS is highly deliberative and hierarchical and is based on three axioms [3]:

1. The functional elements of an intelligent system are sensory processing (SP), world modeling (WM), value judgment (VJ), and behavior generation (BG)
2. The functional elements and knowledge database of an intelligent system can be distributed over a set of computational nodes. They can be represented within a node by a set of processes interconnected by a communication system that transfers information between them.
3. The complexity inherent in intelligent systems can be managed through hierarchical layering and the complexity of the real world can be managed through focusing attention.

In particular, this results in processing architecture composed of a hierarchy of general nodes, each having sensory processing, world modeling, value judgment, and behavior generation (see also Figure 7). This results in the processing of sensory signals (via modeling of the external world and internal state) followed by deliberative planning based on those signals to yield a collection of behaviors in response to the perceived environment. The scope of the processing is a function of the level of the node in the hierarchy: higher-level nodes have broader scope and longer time horizons, while lower level nodes have shorter time horizons and focus on details. This hierarchical organization yields an extensible and portable intelligence-processing methodology [3].

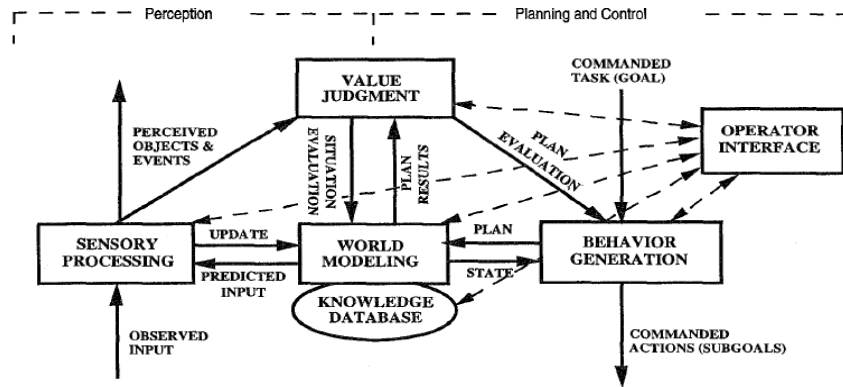


Figure 7: Sensor-Perception-Action Node in 4-D/RCS [3]

While this form of robot coordination and control dominated robotics in the mid-1980s, its use has been limited in part due to the heavy reliance on a high-quality *a priori* model of the environment. This is not to imply that hierarchical control is without application. Rather, it is only well suited for highly structured and/or predictable environments (e.g., spot welding, manufacturing, highway navigation, Demo III, etc.). However, in unstructured and uncertain environments this architecture can lead to unresponsive and rigid, mechanical manner. This is further complicated by engineering and implementation complications that make incremental competency and development difficult.

Hierarchical architectures have also been extended to the multi-robot domain [3]. In particular, coordination is achieved through the addition of higher team levels that process and coordinate activities and view the robot as a collection of node(s). For example, in the 4-D/RCS architecture the highest levels (i.e., levels 5-7) are responsible for multi-robot, multi-system coordination and tactics ranging in scope from the section, to platoon, to company levels [1].

4.3 Comparative Advantages/Disadvantages of Deliberative/Hierarchical Control

Advantages:

- **Highly Structured** – The architecture results in an ordered and structured arrangement for the coordination system. For example in Demo III, the hierarchical levels are separated on time and spatial range resolutions that are based on empirical psychology and increase a factor of 10 for each level [4].
- **Dynamic Replanning** – The multiple levels of deliberative planning ensure that plans can be recomputed so that the strategy and execution operation do not become obsolete [4].
- **Subsystem Maturation** – The levels and nodes of each level provide separation of task operations, which reduces cross-coupling concerns and results in a modular, easily maintainable implementation [3].
- **Logical Processing** – Logical processing is simple to add in the world modeling and value judgments sections of the standard node [3].
- **Manageability** – The use of specific, component nodes in a defined hierarchy results in a more manageable and serviceable architecture. Thus, errors can be isolated to specific areas.
- **Psychological Basis** – The architecture is based on human intelligence models, which are partially derived from empirical psychological studies (such as [55]) [4].

Disadvantages:

- **Rigid & Bureaucratic** – Sensor observations/instructions propagate via a fixed hierarchy. That is, for a sensor response to be communicated to a neighboring sensor/actuator it has to go up level(s), instead of being communicated directly. Further, certain decisions/planning issues are made by higher levels, even though the effect maybe local to a particular robot.
- **Exponential Computational Order** – The provably optimal technique often employed by the middle and higher nodes are order NP, that is, the computational complexity grows exponentially as the order of the system increases and involves the algorithm guessing and verifying the solution.
- **Unnecessary Processing** – The architecture defines a rigid structure between sensory processing, modeling, and actions. This can hamper system performance in “obvious” or “instinctive” applications. For example, if a sensor perceives a second robot ahead it need not model, process, and make judgment to simply avoid actions which will cause a collision.

5 Reactive Architectures

5.1 Overview: “Think maybe, use actions and directly respond to the stimulus”

Almost in stark contrast to the planning and analysis of the aforementioned deliberative architecture models, reactive control tightly (and almost wholly) couples perceptions to effector action [9].

This short-circuiting of the abstractions and symbolic processing procedures, allows the robot to respond very quickly to changing information or constraints in unstructured environments about which no *a priori* information is available. Thus, reactive control is essentially a “reflex” mechanism where stimulus-response pairs govern actions. While a powerful and flexible control method, the application of reactive architectures is limited to approaches where sensory inputs can be cross-referenced to future actions. Implementations of this technique usually do *not* store large quantities of information, have small memories, do not compute/store internal representations of the world/environment, and have no ability to learn over time [53].

5.2 Reactive Control Theory

Reactive architectures are based on reactive control theory [18]. This, in turn, is based in part on the motion-pathways observed in nature and on situational activity experiments [17]. As shown in Figure 8 these systems are purely at one extreme of the robotics spectrum and, in general, use a unique representation of the environment where virtually no information is stored and where planning and general cognition are minimized. Controllers/systems based on this theory (and the coordination patterns which arise from this model) have the following general characteristics [22]:

- Actions consist of a fixed set of reflex (i.e., stimulus-response) rules
- Inherently concurrent (parallel)
- Limited by their lack of internal state
- Very fast computation and motions
- Unable to plan ahead

- Unable to learn

Said another way, purely reactive systems are governed by the following assumptions [9]:

1. The environment lacks temporal consistency and stability they do look for consistency and stability, in the sense that the selection of the sensor/motor pairings are sufficient to operate in the assumed environment.
2. Sensor/perception data are adequate for the task set.
3. It is not efficient nor trivial to localize (to some arbitrary world coordinates).
4. Symbolic knowledge-bases and logical inferences based upon them are of marginal value.

The direct mapping between sensors and effectors along with minimal, if any, state information results in a simple, but canned, coordination patterns. The sum of these sensor-action pairs is a collection of rules that map specific situations to specific actions. For example, if a robot senses an obstacle to one of its sides it may avoid obstacles by steering (some fixed amount) in the opposite direction. This method works as long as the robot correctly perceives its environment and that a sufficiently large enough collection of reflex arcs has been established.

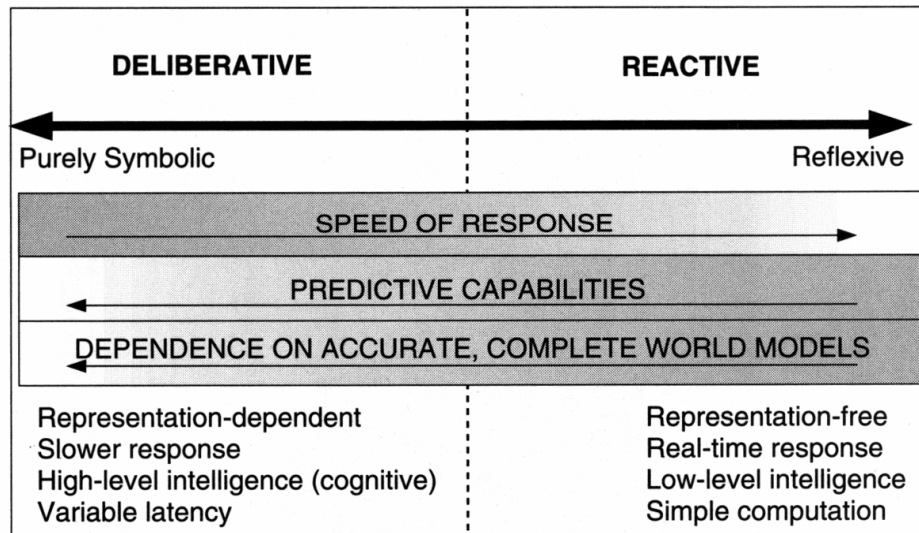


Figure 8: Robot control spectrum (from [9])

However, this results in somewhat obvious issues, such as the following (see also Figure 9) [50]:

- The world is divided into fixed (mutually exclusive) situations so that only when a catalogued situation is perceived can the resulting action be triggered. Thus, it is not readily possible to have mixed actions or complex response patterns.
- This places a significant developmental burden as it not only requires the development of an exhaustively large database to accommodate all possible robot action-sensory pairs (which grows combinatorially as the number of sensors and effectors present on the robot), but also an inconvenient split in the overall motions is desired.
- Reactive systems can support multi-tasking (i.e., parallelism); however, the underlying computing system needs to have a multi-threaded/parallel execution model. The ability to multi-task is critical in reactive systems: if a system cannot monitor its sensors in parallel, but must go from one to another in sequence, it may miss some event, or at least the onset of

must go from one to another in sequence, it may miss some event, or at least the onset of an event, thus failing to react in time.

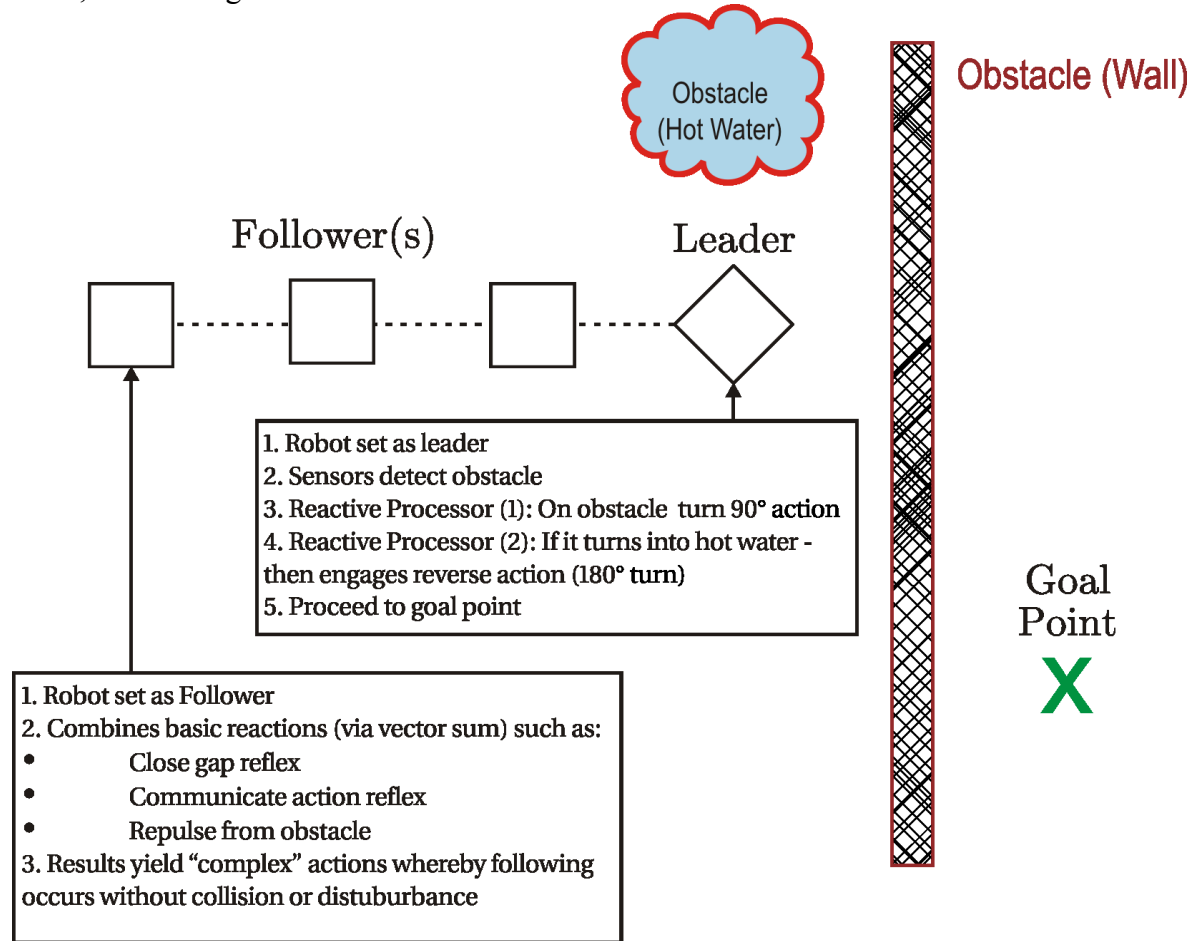


Figure 9: Reactive architecture overview

Reactive control methods have also been extended to the multi-robot domain and have been used with robot teams having up to twenty members [13]. Reactive multi-robot methods generally employ some form of reactive response to signals to interact with other robots. The level of interaction can vary from a simple model where the robots are ignorant of the coexistence of other robots and consider them obstacles (to be avoided) to complex models having informed and intelligent models of the robot. In these cases the controller can differentiate between other robots and obstacles and, if measured, can use proximity information to calculate a robot density, which is connected to further reactive actions. The principal result of these efforts is that complexity of a centralized planner is replaced by the complexity of inter-robot and inter-behavior dynamics [52].

Finally, to account for and optimally choose between multiple, competing actions the reactive control theory introduces the concept of arbitration [50]. This is quite similar to the method used in business and trade and operates by executing those actions that have the highest likelihood and cause for execution.

5.3 Comparative Advantages/Disadvantages of Reactive Architectures

Advantages:

- **Computationally Simpler** – There is no need for extensive processing as little planning and world model creation is used [18].
- **Reflex Arcs** – Actions are directly coupled to sensory inputs and processes. This can simplify programming and debugging as inputs can be constructed as simple, standard behaviors (e.g., if obstacle then move back one step).
- **Responsive in Dynamic Environments** – The “Stimulus-Response” model yields fast (i.e., millisecond) control interpretations and responses, especially since there are no complex processing steps [19].

Disadvantages:

- **Based on “basic” action-reaction pairs** – some coordination responses (e.g., build/break team formation, reconnaissance data construction) may need to apply complex logical/symbolic processes such as game theoretic bidding or EM, which are not easily included in the Subsumptive model [6].
- **Restrictive** – There is a tight coupling between sensory inputs and effector outputs
- **State memory limited** – there is no significant internal representation of the present. The little, short-term memory (if any) makes learning, pattern recognition, etc. difficult [53].

6 Behavioral Architectures

6.1 Overview: “Let your available actions dictate your thinking and options”

Behavior-based robotics is a large and actively explored area of robotics research. Originating from the Brook’s Subsumption Models, this architecture represents a design methodology for autonomous robot control that uses models of human (animal) perceptions and responses [17]. The resulting architecture consists of a collection of real and virtual behaviors that achieve/perform certain (well-defined) goals. By combining and controlling these behaviors, complex and “natural” robotic motions and actions can emerge [49].

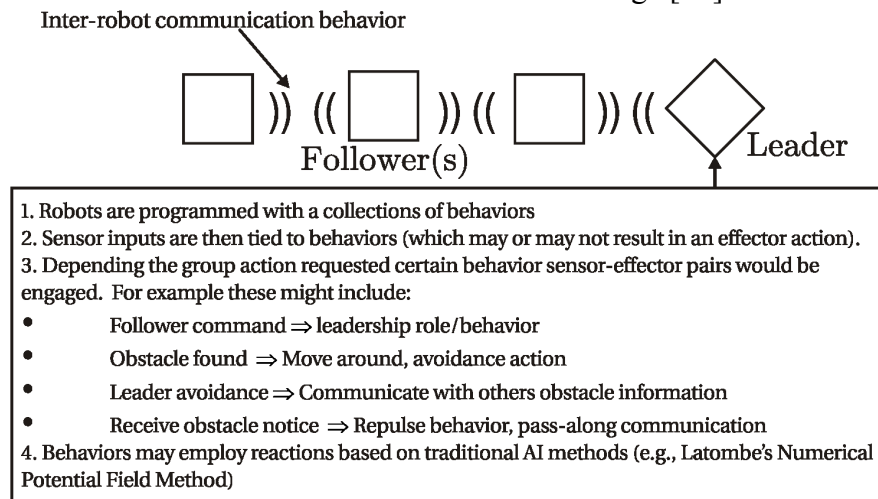


Figure 10: Behavioral Architecture overview

6.2 Subsumption Assumption

Developed by Arbib and Brooks in the mid-1980's this "paradigm shift" in AI reasoning is a method for exploring and constructing physically-grounded systems that tightly connect perception to action [6]. This allows for the development of optimized computational and programming tools to embed robots concretely in the world (see also **Figure 11**) [18].

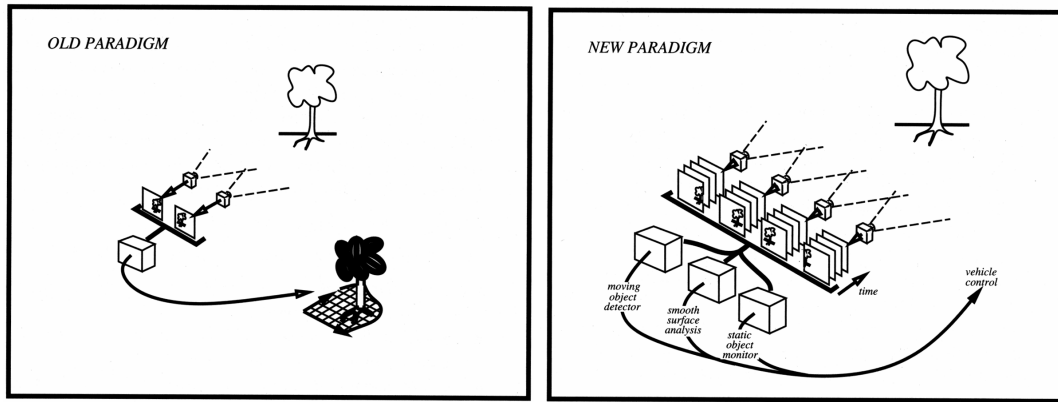


Figure 11: Subsumption's paradigm shift (from [9])

The subsumption architecture has the following principles and characteristics [6], [50]:

- Robot systems are built from the bottom up
- Components are task-achieving actions/behaviors; that is, they are objects with properties that can be shared and superseded as is common in object-oriented development.
- Components can (and to some extent should) be executed in parallel
- Newly added components and layers exploit the existing ones and each component operates in a manner so as not to disrupt the tight coupling between sensing and action.
- There is no need for internal models: "the world is its own best model"

The original subsumption architecture was implemented using a language based on finite state machines (FSMs). An FSM can be in one state at a time, can receive one or more inputs, and send one or more outputs. FSMs are connected by communication lines, which pass input and output messages between them. Coupling between layers need not be through a dedicated communication system, but rather through the world (e.g., a robot will mark or modify the environment in a unique manner with its "fingerprint") [18]. This allows for coordination in environments where traditional (i.e., radio or wire-based) communications are severed or unsecured. Subsumption uses a bottom up approach, so layers can keep being added depending on the tasks of the robot. How exactly layers are split up depends on the specifics of the robot, the environment, and the task. There is no strict recipe, but some solutions are better than others, and most are derived empirically [6].

This yields the key principle of the Subsumption Architecture: *"use the world as its own best model."* The idea is that if the world can provide the information directly (through sensing), it is best to get it that way, rather than to store it internally in a representation (which may be large, slow, expensive, and dated). This assumption, while similar to the one used in reactive control theory, does allow for cognition and modeling in addition to the fusion of actions to yield complex behaviors. For example, consider a two layer robot where: the lowest layer uses a vector sum from a se-

ries of proximity sensors to fire a halt reflex and the second layer is an “urge to explore” layer that is attracted towards moving objects tracked via differential changes seen in the proximity sensors. These simple, independent behaviors were implemented and combined on MIT’s Tom and Jerry robots [18]. The result is a C2 (command and control) system where each module/layer did not have (or need) explicit information about the other(s) yet was capable of producing complex behaviors, namely: exploring and following moving objects, but remaining far enough behind to avoid collisions with other robots or obstacles (which the system may or may not distinguish between) [18].

6.3 Description of behavior-based robots

Behavior-based robotic (BBR) coordination and control theory is an alternative method that focuses on controlling behaviors or actions of a robotic system. Originally derived in response to the stiff, mechanical, nature that generally characterizes robots built using highly centralized methods of control, behavior-based control is primarily focuses on the notion of sensing and planning in the environment as compared to the knowledge representation and planning focus which dominates symbolic and centralized control methods [9].

BBR controllers consist of a collection of behaviors that achieve and/or maintain goals. In particular, behaviors are implemented as control laws or as a processing element/procedure. The behavior itself can be viewed as a “black box” which can take inputs (including those from other behaviors) and process these perceptions to yield outputs to the robot’s effectors and/or to other behaviors. Therefore, BBR architectures allow for a rich structure and interactions between behaviors, which, in turn, allows for complex interactions and emergent behaviors [49].

Derived from biology and animal colony behaviors, behavior-based control directly couples perception to action via a series of logical (AI-based) inference patterns (see also Figure 12). Generally, Behavior-based AI defines a series of behaviors and actions and then allows the controller to perform coordination by changing its response (i.e., action) to perceived environmental signals [19].

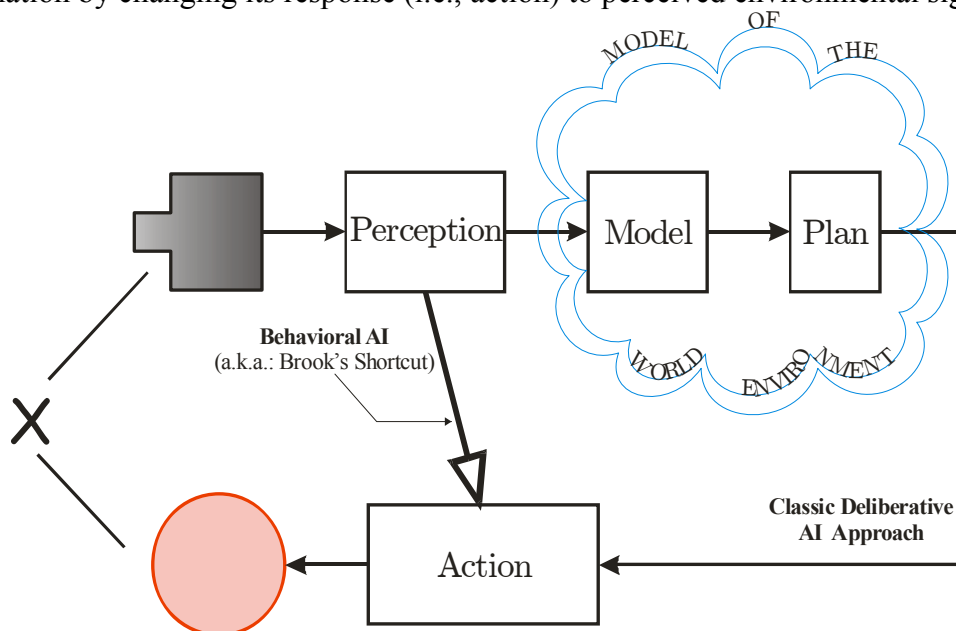


Figure 12: Behavior based vs. traditional architectures

Based upon the “new paradigm” of nouvelle AI, Behavior-based robot control has been and still is an actively researched area in AI and control theory [19]. However, it is also one of the most controversial areas in AI research as it quintessentially argues that the symbolic foundations of classical AI are inadequate and “fundamentally flawed.” As Brooks states in [18], “planning is just a way of avoiding figuring out what to do next” and thus is more concerned with the development of intelligence through “intelligent,” direct interconnections between sensors and actuators.

BBR systems, like reactive systems, also have different “parts” or layers, but unlike deliberative systems, they are not as isolated and varied from each other. In BBR the processes are encoded as behaviors, thus processes take inputs and send outputs to effectors or other behaviors with minimal latency. For example when a robot needs to plan ahead, it does so in a network of behaviors, which talk to each other and send information around, rather than a single planner, as with reactive systems. Behavior-based systems are an alternative to hybrid systems; these days, they are equally powerful and equally popular [49],[62].

The novelty of the Subsumption Architecture along with the way state information is represented and distributed is one of the factors leading the strength and recent popularity of BBR methods. Furthermore, the data structures often used in implementing BBR systems are generally conducive to distributed computing (i.e., information is not generally centralized or centrally manipulated) [9].

Even though built upon a framework conducive for distributed computing, the coordination of BBR systems is nontrivial. BBR essentially requires that the coordination activities be performed as “non-observable” behaviors, which update the behavior responses to given perceptions. Furthermore, repeated tasks such as error quantification and noise suppression are difficult to do without inadvertently adding a bias to the perception/behavior model [9]. Thus, erroneous/noisy data can potentially lead to significant response errors if not corrected through an additional layer of behavior interaction (e.g., a noise filter behavior) [12].

Behavior-based systems typically do not employ such a hierarchical division but are instead integrated through a homogeneous distributed representation. Like hybrid systems, they also provide both low-level control and high-level deliberation; the latter is performed by one or more distributed representations that compute over the other behaviors, often directly utilizing low-level behaviors and their outputs. The resulting system, built from the bottom-up, does not divide into differently represented and independent components, but instead constitutes an integrated computational behavior network. The power, elegance, and complexity of behavior-based systems all stem from the ways in which their constituent behaviors are defined and used [62].

Consequently, the organizational methodology of behavior-based systems differs from other control methods in its approach to modularity, the way in which the system is organized and subdivided into modules. The BBR philosophy mandates that the behaviors be relatively simple, incrementally added to the system, and be capable of concurrent execution. Subsets of behaviors are executed concurrently so that the system can exploit parallelism, both in the speed of computation and in the resulting dynamics that arise within the system itself (from the interaction among the behaviors) and with the environment (from the interaction of the behaviors with the external world). Behaviors can be designed at a variety of abstraction levels. In general they are higher than the robot’s atomic ac-

tions (i.e., typically above “go-forward-by-a-small-increment”, “turn-by-a-small-angle”), and they extend in time and space. Some implemented behaviors include: “go-home,” “find-object,” “get-recharged,” and “pro-tropism.” Because behaviors can be defined at different levels of abstraction and can represent various types of information, they are difficult to define precisely, but are also a rich medium for innovative interpretations [9],[49].

6.4 Motor Schemas

While BBR does not explicitly dictate the use of schema theory as the basis for designing behaviors for future motions, the use of Motor Schemas is not only simple to understand and implement, but also biologically motivated and thus can be compared against well-documented studies [17]. Motor schemas and their related methods are based on schema theory and have been used extensively by Arkin because of the following characteristics [9],[12]:

- Motor actions are explained as a combined function of the concurrent control of many different activities.
- States in the schema model can store both actions and reasons for these actions
- Schema theory provides a strong foundation for connecting perceptions to language via a series of inference steps.
- Schema theory provides an excellent model for the application of adaptive/reinforcement learning via schema tuning.

6.5 Multi-Robot Teaming

Robot team coordination in the BBR domain is achieved in a similar manner as coordination is in the reactive controller domain; that is, primitive (i.e., basic, reflex type motions) of varying levels of sophistication are used to coordinate the robots. In BBR robot teams, the concept of composite behaviors (i.e., high-level behaviors with memory and symbolic logic) allows the development of multi-robot systems which can apply logical inference and decision processes to variable sensor responses to give complex behaviors and more natural results (e.g., an *find-neighbor* composite behavior would recall previous robot positions and velocities and could predict in which direction its nearest neighbor would be while a primitive version of the behavior would simply ping and respond to sensor values) [8].

Good examples of the use of BBR in robot teams are research efforts in the formation and maintaining of robot group formations (patterns) [8], [60]. As shown in Figure 13, the purpose of these efforts is to have a small team of robots begin and maintain a given formation. In particular, this was performed using BBR via the following five schemas:

- *Detect-Formation-Position* – The formations were detected using three schemas (behaviors), namely: unit-referenced (each robot calculated the center and moved so that its position would maintain the formation center), leader-referenced (a leader robot sets the formation), and neighbor-referenced (formations are maintained relative to the nearest neighbor).
- *Move-to-goal* – Use formation information to move towards a position
- *Avoid-static obstacle* - Avoid obstacles while moving towards goal
- *Avoid-robot* – Recognize and avoid robots. Robot differentiation from obstacles allows for more complex behaviors as this information can be used to as part of the formation detection
- *Maintain-Formation Position* – Adjust motions to maintain formation, especially against noise and external disturbances.

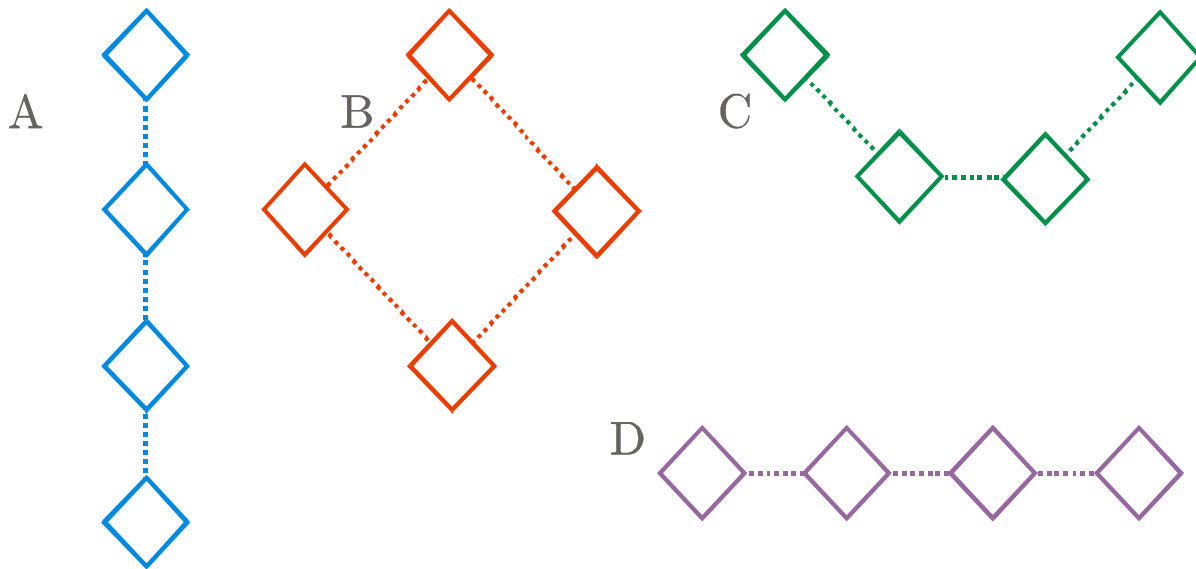


Figure 13: Robot formations (A: Row, B:diamond, C:wedge, D:column)

6.6 Comparative Advantages/Disadvantages of the Behavior Architecture

Advantages:

- **Subsumptive Assumption** – Physical grounding is one of the principal features of this architecture. That is, this architecture is based on the view that “the world is its own best model” and thus there are no, potentially confusing, layers of abstraction between the system and the world [18], [17].
- **Biological Inspiration** – By sharing a string basis to natural systems, it provides a framework for multi-robot control based on observations seen in other natural systems
- **Planning Possible** – Often just-in-time, it is implemented via network of plan generating behaviors which communicate/share necessary information as operands on which to apply said behavior(s) [53].
- **Behavior Representation** – Unlike reactive systems, this representation style allows for a distributed storage of state information, which allows for new behavior learning, etc. [49].

Disadvantages:

- **Nondeterministic Planning** – Planning/strategic motion only done if time permits [53].
- **Modular Approach** – BBR is organized into modules and that the behaviors be relatively simple objects that can be incrementally modified and need not execute in a serial manner. This insistence on a highly “object oriented” design is not compatible with some algorithms (e.g., computer vision filtering) that are not easily parallelizable [49].
- **Need for arbitration rules** – A means is needed to determine and quickly decide which behavior to execute at a given point, especially if there are competing behaviors/actions [49].

Observations:

- Not the same as reactive systems – the parts/layers are not separate, isolated objects from each other, but rather (within the behavior framework) can share information can reference one another [53].

7 Hybrid

7.1 Overview: “Combine thinking and action (independently) in parallel”

To some extent Hybrid methods are a compromise of Deliberative, Reactive, and Behavior based robotic architectures. The combined technique aims to combine the different representations, time-scales, and execution strategies to yield a best-of-breed solution that overcomes some of the strict assumptions associated with the aforementioned distributed methods [53].

Hybrid architectures generally have a rationalizing sub-system that performs higher-level functions (e.g., planning, pattern recognition, etc.) and a lower-level, reactive systems which performs basic tasks (e.g., avoiding obstacles, staying on path, etc.). This is capped by a third module, the arbitrator, which performs communication and prioritization between the higher and lower level functions. Thus, resulting in a “three-layer” hybrid control system [50].

7.2 Hybrid Model Description

Both deliberative and reactive control along come with significant trade-offs. Deliberative control often requires an approximate model of its operating environment and its methods do not respond well to situations with large variances and ambiguities. While reactive control addresses the need for robust performance in complex, dynamic, and unstructured environments/domains, purely reactive systems can not easily exploit planning, operational/environmental patterns, abstract models regarding its sensing/perception inputs, nor symbolic artificial intelligence.

A mixed strategy provides a more reasonable method for robot coordination for a general case where there are natural constraints during operation in a complex environment. In situations where the environment is restricted (e.g., the location is approximately known) or quasi-static (i.e., the environment does not change dramatically over short periods of time) hybrid control allows for the use of classical, symbolic AI processing and planning algorithms to yield approximately (or fully) optimal solutions to environmental stimuli). Further, this can yield a more robust solution as complicating factors (e.g., noisy sensor data, dead-reckoning error in effectors, etc.) can be analyzed and compensated online. Thus, hybrid control is seen by many researchers as being the vehicle necessary to deliver and realize the full potential of behavior-based robotic approaches [62].

The hybrid technique also complements and is easily integrated into behavioral architectures. For example, aspects of the hybrid architectures can be modeled as “black-boxes” that have the same interface as a general behavior. This “super-behavior” would then communicate and be directed by other behavior modules. An alternative approach is to build an mostly deliberative module that acts as a governor to the behaviors and their response with a similar arbitration module being available to resolve conflicts in effector actions that arise via dynamic system reconfiguration [49]. For example, a planning behavior based on A* or D* could dynamically change the mapping in an obstacle avoidance behavior based on the its knowledge of the optimal path.

However, the design rules and associated implementation details for “three-layer” hybrid architectures is also a compromise of both ends of the aforementioned robotic response spectrum. In particular, details such as the nature of the boundary between deliberation and behavior and optimal arbitration rules are often arbitrary to a particular system as general rules for these problems are still an active area of research. Section 6.6 of [9] lists four variations for constructing the hybrid model for the concurrent mobile planning and execution robot motion task [9].

The Hybrid systems approach is strongly supported by biological evidence and general models of the cognitive processes observed in nature. Thus, not only does this architecture benefit from advances in the cognitive and psychological sciences, but it also contributes as a vehicle for testing and refining theory [22]. Finally, this property (i.e., high-level though driving low-level reflex arcs) can be exploited in biomimetic robot control and/or in robot workforce interaction situations that require “natural,” fluid, and flexible control systems [27].

In the multi-robot domain, the hybrid architecture is combines and utilizes the multi-robot techniques detailed in previous sections. In particular, it approaches the multi-robot coordination problem by first factoring the positions and actions of the others member in its assignment/processing of behaviors to be executed.

7.3 AuRA

Arkin’s Autonomous Robot Architecture (AuRA) was one of the first hybrid controllers to successfully integrate reactive (schema-based) and deliberative control. This allows a workforce of robots to calculate and incorporate information from a conventional planner which can supervise and provide macroscopic “vision” to a behavior-based robotic system [9].

As shown in Figure 14, AuRA’s structure consists of two principal components: a classic system based on traditional AI and a reactive system that directly connects actuation and sensing. The highest level of AuRA is the Mission Planner which uses traditional hierarchical planning system (as often found in the intelligent control /literature) to establish high level goals and the constraints within which it must operate. The Spatial Reasoner uses cartographic knowledge obtained from a previously constructed map to devise a sequence of path lengths which the robot must execute in order to complete its mission. Originally computed via an A* algorithm, the spatial reasoner module uses a finite state sequencer. The Plan Sequencer then uses these path components to specify and initiate a collection of behaviors for the robot. At this point the behaviors are processed as part of a reactive execution model, thus representing the conclusion of the deliberative processing stage by the robot [12],[13].

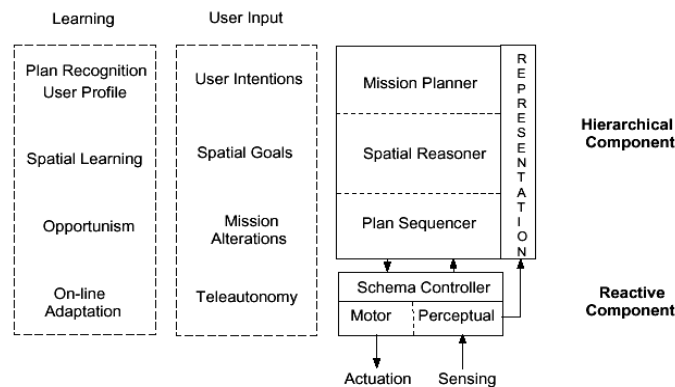


Figure 14: Macro level component structure for AuRA [12]

The schema manager is wholly responsible for controlling and monitoring the behavioral processes during run-time. In particular, each motor behavior/schema has a mapping of associated perceptual schema(s) which provide the necessary stimulus/stimuli required for this behavior. Thus resulting in the necessary action-oriented perception which is the basis for reactive navigation. As with tradi-

tional reactive control, the behaviors can operate asynchronously and transmit their results to an accumulator process which sums, normalizes, and prioritizes these inputs and transmits them to a low-level control system for execution (see Figure 15) [9].

Unless a failure or system error occurs, reactive control continues unabated without additional inputs from a deliberative control process. These failures or system errors are quite broad and encompass everything from changes to the “homeostasis” of the robot (e.g., battery status, temperature, motor operation) to lack of progress errors (e.g., time-out, lack of progress, etc.) often caused by errors in the initial path computed by the deliberative controller. In particular the deliberative process calls the modules in reverse order (i.e., plan sequencer, spatial reasoner, and mission planner) this allows the system to preserve as much information about the mission that the robot may have already obtained/computed [12].

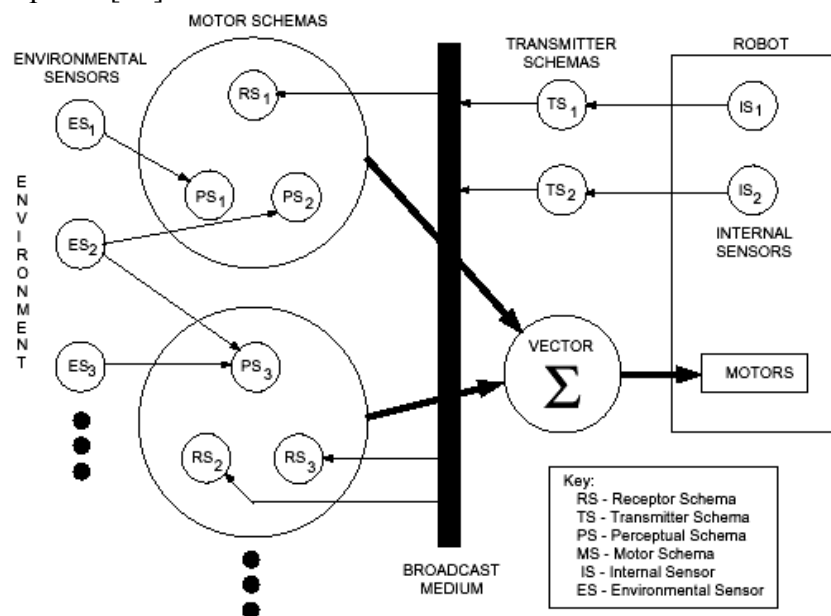


Figure 15: Reactive layer of the AuRA model [12]

The resulting AuRA architecture is modular, relatively flexible, and somewhat generalizable. It also provides a simple method to add and utilize specialized robots in a straightforward manner. The value of each of these aspects has been demonstrated in various venues, most notably the 1993 AAAI Mobile Robot Contest, in which a team of AuRU-driven robots won [12].

AuRA also demonstrates many of the classic strengths and weaknesses associated with implementing the hybrid architecture. As illustrated by Figure 14, it is a highly modular design. Thus, components can be modified and replaced simply, which allows for the development of specialized robots and, potentially, the ability for a robot to change its specialization in real time. A second feature of AuRA is the flexibility between paradigms as at an extreme it can approximate a purely reactive or deliberative system [9]. Finally, Balch and Arkin have experimentally verified and demonstrated AuRA’s use for workforce coordination and collaboration and that the controlled robots share features/properties associated with the base architecture [13].

7.4 Servo-Subsumption-Symbolic (SSS)

This hybrid architecture is named after its three principal layers: a servo layer, a subsumption layer, and a symbolic processing layer. The servo and symbolic layers interact in a fairly standard manner to provide behavioral modularity and flexibility to the underlying servomotor controllers via a manner similar to techniques used by subsumption methods. SSS's principal feature is its use of world model representations which are viewed not as a necessity but rather as a convenience. The symbolic layer controls the selection of behaviors (i.e., turns behaviors on or off) as well as modify the particular properties associated with the behaviors [27].

For example, on the TJ robot, the symbolic layer processed strategic decisions (e.g., where to go next), the subsumption layer was primarily concerned with tactical decisions (e.g., where to go now), and the servo layer was responsible for low-level control and operation (e.g., how to go/get to some given location). In particular, a course map is used by the symbolic layer which is refined using sensor data collected by the subsumption layer [27]).

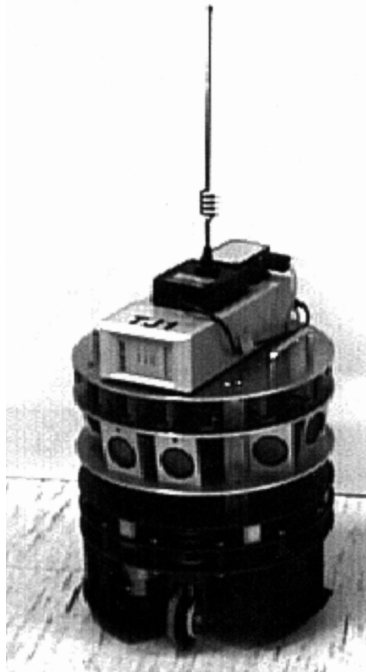


Figure 16: The TJ robot [27]

This approach is more generally specified in the Generic Robot Architecture. This architecture also uses three levels to connect symbolic planning to reactive controls, namely : planning (e.g., generate sequences of tasks to achieve high-level goals via a STRIPS algorithm), course system (e.g., translate the aforementioned plan into tasks), and functional level (Translate a set of functional modules and configurations for execution via a reactive processor/module) [59].

7.5 Multi-Valued Logic

Developed at SRI International this technique is based on the use of a multi-valued logic (MVL) representation for behaviors. In particular, the MVL logic allows for a variable planner-controller interface with strong context dependency (i.e., the planning versus reaction trade-off is varied by MVL in response to the environment). Pre-planned behavior models are selected as to be conducive with MVL and allow for “non-crisp” (i.e., non-binary or fluid) logic. The Flakey robot has demonstrated the effectiveness of these techniques in an non-uniform indoor office environment [67].

7.6 Comparative Advantages/Disadvantages of Hybrid Architectures

Advantages:

- **Biological Inspiration** – By sharing a string basis to natural systems, it provides a framework for multi-robot control based on observations seen in other natural systems
- **Combined Approach** – Allows for both reactive and deliberative methods to be used in a unified manner. By allowing for the integration of various components this method gives its designer a great deal of flexibility, especially in the multi-robot domain (where some robots could spend varying amounts of time on control)
- **Flexibility** – Hybrid architectures allow the designer several degrees of freedom for introducing adaptation and learning methods
- **Deliberative Processes** – This architecture can easily integrate deliberative processes and pattern recognition steps [61]
- **Hybridization** – Hybrid behaviors and control are a central tenant to this architecture

Disadvantages:

- **Modular Approach** – BBR is organized into modules and that the behaviors be relatively simple objects that can be incrementally modified and need not execute in a serial manner. This insistence on a highly “object oriented” design is not compatible with some algorithms (e.g., computer vision filtering) that are not easily parallelizable [49].
- **Implementation complexities** – This architecture provides several degrees of freedom and flexibility. However, the lack of rigid structure leads to implementation complexities such as having a mechanism to switch modes (deliberative and reactive) and a mechanism for arbitration between these processes.

8 Free Market Architecture

8.1 Overview: “Trade and act to maximize your profit”

The free market robot team architecture coordinates robot workforce activities based on a free market model. Compared to centralized methods, this architecture is able to perform the coordination problem in a computationally efficient and responsive manner. Further, it yields robust execution plans which result in a robot team which tweaks and exchanges tasks to efficiently complete complex tasks in dynamic environments [30].

8.2 Microeconomic Model

Derived from the generally accepted economic theories for free markets, this architecture is based on the assumption that the mission being executed (i.e., the robot’s environment) is one where market economies are superior to centralized planning [76].

In particular, this method uses free market economic theory. This theory states that market economies are an ideal means to organize large populations into a productive group in a manner that maximizes the output/return to the group [68]. Thus, by cooperation in a competitive environment the individual robots trade task components as each robot in the team works to maximize their individual profits [76].

The novel aspect of this approach is that it is based upon a microeconomic model instead of a biologically inspired system. The selection of a free market economy (and in particular a laissez-faire model) is apt because the assumptions of a free market generally hold for robot workforce executions in both structured and dynamic environments. That is, the member robots in the modeled economy are free to enter and exit from the market, generally have full information about the market, are approximately similar (i.e., the individual robots may have comparative advantages, but no one robot clearly dominates and influences the outcomes of all other robots in the population), and are trading a ubiquitous product/service (i.e., task competition) that has similar value throughout the local environment [86].

As the resulting free market system allows for the open exchanges of services (i.e., completion of sub-tasks) and rewards (i.e., profits from previous services rendered), it is unencumbered by the need for centralized planning, it overcomes the weaknesses and computational intractability associated with these techniques. The result is a system in which individual robots acting to maximize their individual return also result an efficient aggregate solution for the team of robots in a manner similar to how competitive firms operating freely result in a more productive society. Further, an individual-centric approach allows this method to assign direct correlations to individual actions thus having the individual reap or suffer as a directly due to certain actions. This focus also yields a simple, yet powerful, basis for more complex interactions including the transition between cooperation and completion, self-adaptation, and learning [69],[86].

8.3 Free Market System Employed

Clearly, the goal of this architecture is to maximize the return and minimize the costs associated with a given distributed task. To determine the revenue and costs for the systems a series of functions and maps are needed for the outcomes ($f_{outcome}$) and the costs ($f_{expense}$). This is then combined to yield revenue and profit equation for both the individual robots and entire multi-robot team (see Equation set 2) [30]. Further, these functions have to not only take into account parameters that are controlled by the robot and/or its configuration but also its relative position in the environment with respect to its ability to analyze and execute these tasks [68]. Assuming an honest economic system (i.e., self-organized without fraudulent behavior) these relations can be expressed mathematically as:

Revenue and Cost:

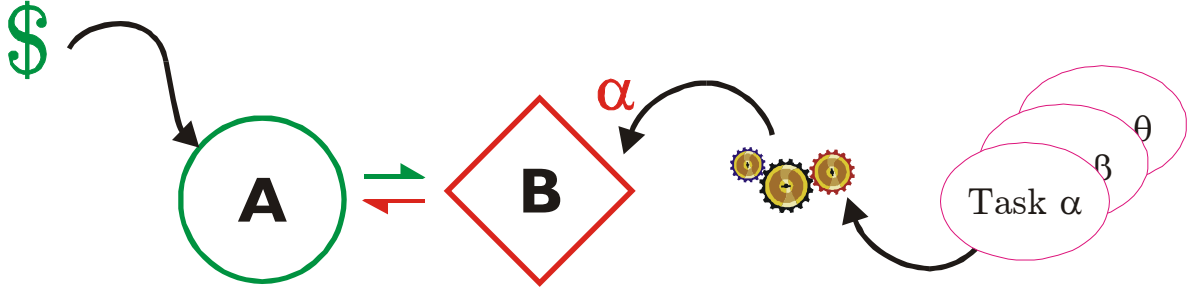
$$\begin{aligned} r_i &= f_{outcome}(Plan, Future Position) \\ c_i &= f_{expenses}(Plan Cost, Environment) \end{aligned} \quad (2.1)$$

Profit:

$$\begin{aligned} \pi_i &= r_i - c_i \\ \text{where: } c_i &\leq r_{i-1} \text{ (i.e., the cost can not exceed revenues to facilitate transaction)} \end{aligned} \quad (2.2)$$

$$\Pi_{team} = \sum_{i=1}^N \pi_i, \text{ where: } N = \text{Number of Robots}$$

The aforementioned costs and rewards are set in the price and bidding process of the free market model. This method is illustrated for two robots in Figure 17. In particular, bidding is used instead of averaging as this allows for the adjustment of price to include each individual robot's opportunity cost for a given instant in time. Thus, external events in the environment/world can be encapsulated though the opportunity cost function(s) [76].



1. *Start*: A wishes to purchase service α from B. A reaps a profit of x (i.e., $x = \text{Revenue}_A(\alpha) - \text{Cost}_A(\alpha)$) and B has a cost y for the service (i.e., $y = \text{Cost}_B(\alpha)$)
2. *Negotiate*: Depending on the competing opportunities and values of tasks that have been performed and need to be performed standard bid trading is performed to yield a final cost for α (computed using classical game/auction theory from AI).
3. *Trade*: Task is executed.

Figure 17: Illustration of Free Market Architecture bidding/negotiation

The free market architecture also entails a mechanism that allows robots to switch roles from being cooperative (i.e., [heterogeneous] robots with complementary roles interact to exchange services) or competitive (i.e., the robots are in a situation where an individual's profit is adversely affected by the presence of another) in a manner that maximizes their individual return [76]. The flexibility of game/auction theory, the market-model, and the robot negotiations allow individual robots to automatically calculate the best posture for a given task regardless of the homogeneity of either the task or the team itself [30],[76].

Finally the market-model of the free market architecture provides a mechanism for learning. Individual robots have the capacity to learn which bidding strategies and actions are most successful and profitable. The aggregate collection of bidding strategies allows the team "to learn" actions/responses to a changing environments [76]. This also follows microeconomic theory which explains that free-markets (such as the basis for this architecture) with free entry and exit show a remarkable ability to adapt and respond successfully to changing conditions [86].

8.4 Economy Based Robotic Coordination

This theory was implemented by Stentz and Dias as a multi-robot solution to a robot foraging or multi-city, multi-agent traveling salesman problem (TSP) and has been outlined in the latter sections of [30] and [76]. Figures 11 and 12 show the results of this architecture for quickly reducing the path cost and determining the appropriate actions/paths.

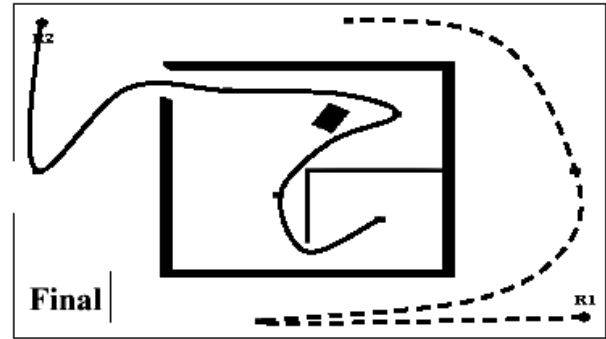
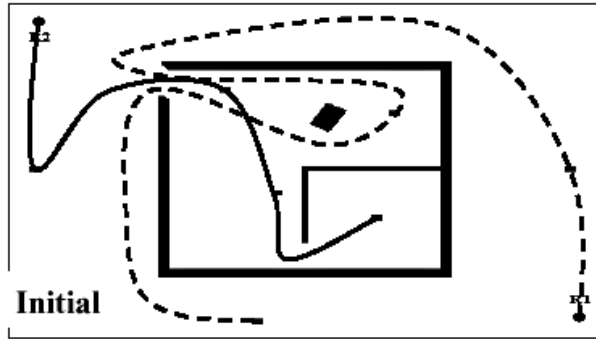


Figure 18: Initial assignments and final tours for 2 robots and 8 cities (14.7% decrease in team cost) [30]

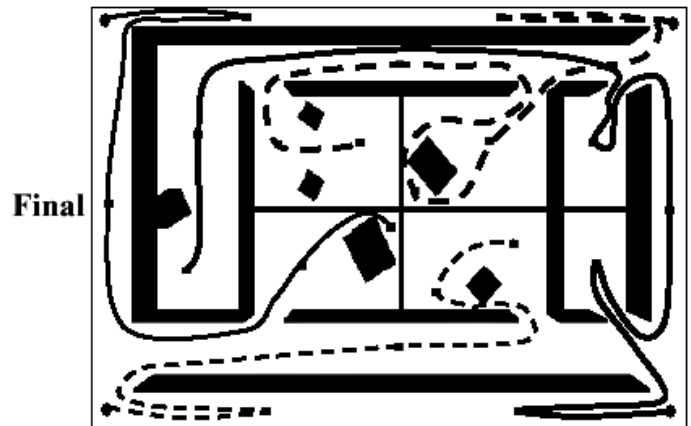
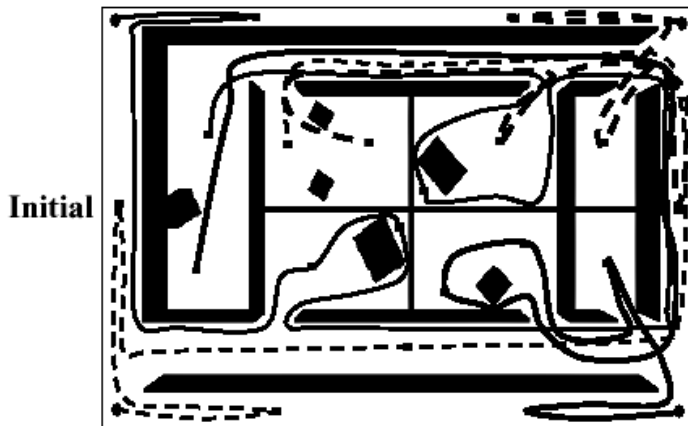


Figure 19: Free Market Architecture final tours for four robots and 20 cities [30]

8.5 Comparative Advantages/Disadvantages of Free-Market Architectures

Advantages:

- **Economic Model** – By using a free market exchange model the system is able to monitor costs and adjust its actions to maximize the gains for both an individual robot and the team.
- **Dynamic** – The system through its uses of contractual bidding and negotiation is flexible and responsive to changes in the environment.
- **Efficient** – The approach is computation and communication efficient as these resources are used for neither an exhaustive search nor constant inter-team updates/communications.

Disadvantages:

- **Negotiation Protocols** – The design needs to be robust without increasing communication requirements significantly.
- **Model Bias** – The nature and values associated with both the revenue and cost model cannot be too conservative nor too generous, as this will bias the operation of the system and lead to a non-optimal result.

9 Immunity Based

9.1 Overview – “Specific, dynamic assemblages with directed-stochastic actions”

There is a rich collection of scientific literature on immunology and the operations of the immune system. Immunity based architectures (IBA) are founded on the principle that suggests using this rich understanding of a robust multi-agent system operating in a complex and dynamic environment to form a basis for multi-robot coordination [29], [47].

At a primary level IBAs model how acquired (i.e., humoral and cell-mediated) immunity is achieved as a result of the multi-agent control/cooperation of the immune system’s four basic categories of components: monocytes, granulocytes, lymphocytes (B-cells, T-cells), and immunoglobulins (antibodies) [74],[36]. Using this as a basis, the architecture results in methods for robot coordination that are particularly apt for very large robot populations (e.g., thousands, if not, millions of robots). It is hypothesized that populations of the scale seen in the immune system, which are on the orders of billions (of active cells) and trillions (of molecules/factors), can eventually be implemented.

Many of the present computer science applications have used the immune systems as a basis for anomaly detection [29]. Robotic systems have generally used this as an algorithm for behavior development or as an arbitration model for behavior based approaches,[42],[43].

New architectures based on this method are based on the idea that small, minimally capable robots need to think and act massively in parallel in a similar manner as biology complexes antigen to immune cell in order to match and exceed the intelligence and planning afforded by “large” deliberative processes running on large robots [74].

9.2 Immune System ‘In a nutshell’

On the surface, the human immune system has a clear and basic role: the monitoring and preservation of the identity of the body. The operations of this diffuse system (it is a scant more than 1-2% of a person’s body weight) are individually simple, but combine to construct a rich and complex web of interaction and coordination that, while not optimal, display exceptional levels of robustness and flexibility, especially with regards to unknown situations and conditions [45]. The immune system works on two levels: a general response mechanism that is not directed at a specific disease organism/pathogen (i.e., innate immunity) and a specific, anti-body mediated response level that encompasses many of the pattern recognition and situational memory aspects that are a core aspect of the human immune system (i.e., acquired immunity). Figure 20 compares these responses and illustrates the body’s trade-off between response time and effectiveness [37].

Innate immunity is the natural and omnipresent resistance to a variety of pathogens. Its purpose is to act as the first-order, general defense mechanism. This mechanism (illustrated in Figure 21) primarily operates by permitting self/non-self discrimination and by activating certain general kill mechanisms, namely: tissue macrophages, digestive enzymes and acids, resistance by the skin to invasion from organisms, chemicals in the circulation that attach to and help destroy foreign antigens (e.g., basic polypeptides, complement complex proteins), and natural killer lymphocytes [71].

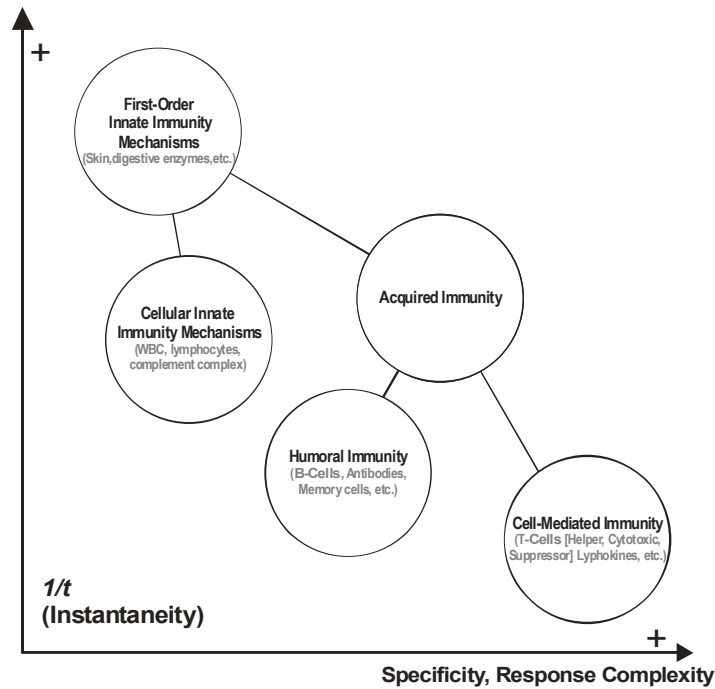


Figure 20: Immune system response structure
(response becomes more specific and advanced with time) [74]

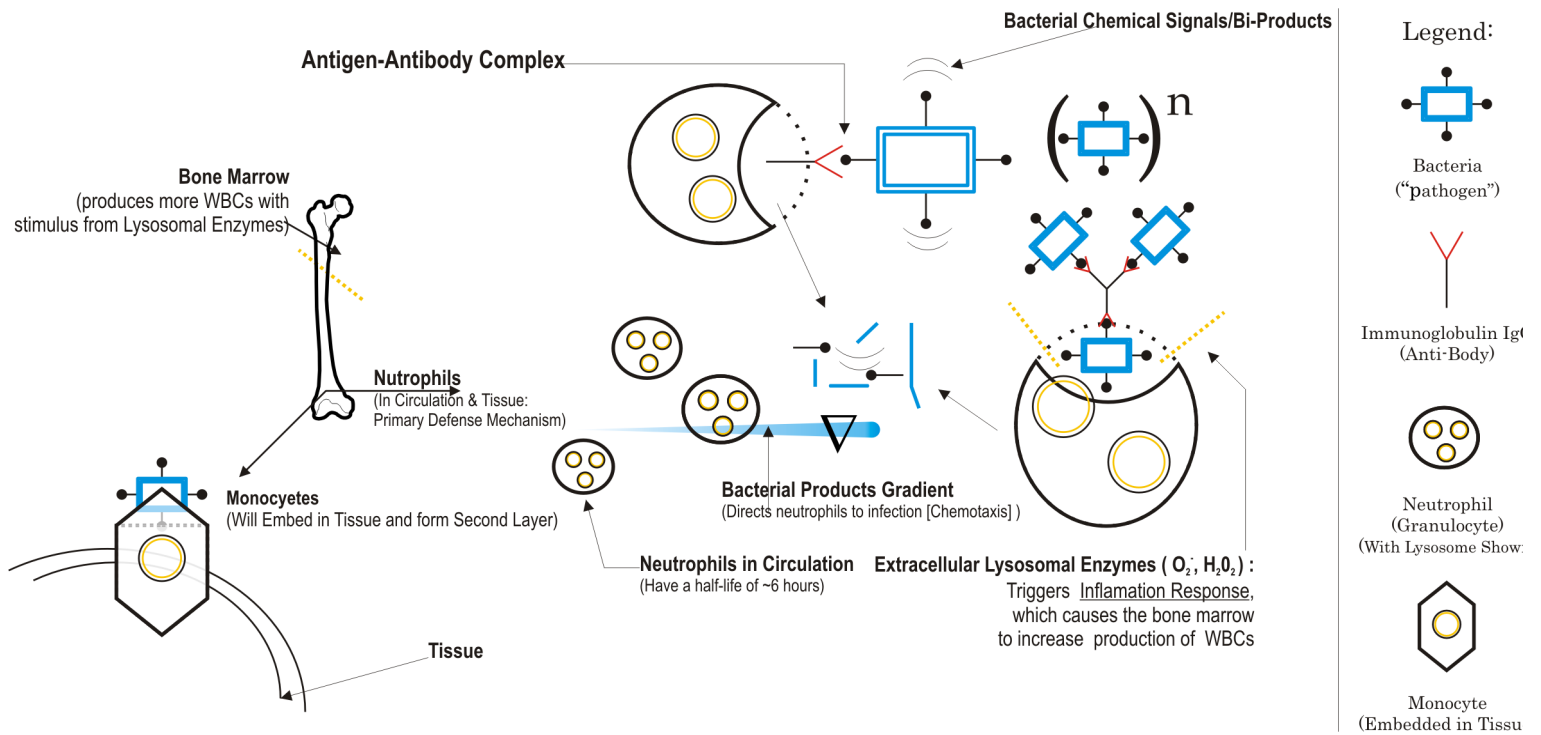


Figure 21: Model of human innate immunity [74]

In sharp contrast to the innate system, acquired immunity is principally about specific responses to specific threats. The specific higher-level response provides the life-long critical immunity (e.g., a person with normal immunity can survive up to 100,000 times the dose/exposure of a pathogen that would be lethal without having acquired immunity) [74]. There are two types of acquired immunity: humoral (i.e., B-cells and antibody control/regulation) and cell-mediated (i.e., T-cells and B-cell assistance/orchestration). Both are initiated by antigens and are signaled by antibodies (i.e., Y-shaped molecules which specifically match certain proteins based upon their encoded specificity; there are some 10 million present in the immune system) [37].

9.3 Artificial Immune System Model and Immunity Architecture

The artificial immunity model exhibits a host of useful characteristics, which define its response and are essential to operation. Salient features/characteristics of the immune system for multi-robot applications include [74]:

- a) **Massively Scalable** – In order to provide the extreme levels of protection that are essential for survival, its operation literally depends on the coordination of trillions of cells and molecules that together act to continually monitor the bioenvironment and, depending upon the perceived threat to the body, activates the necessary control systems to defeat this threat.
- b) **Innate Pattern Recognition** – All types/levels of the immune system use a chemical markers as a signature in order to determine the antigen and its nature. Further, the pattern recognition is used for learned “self-tolerance,” which arrests autoimmune cells during development thus preventing damage to oneself.
- c) **Innate Learning** – A feature resulting from the acquired immune system pattern recognition layers, the immune system exhibits remarkable and diverse learning and memory characteristics. This aspect alone has been the focus of many machine learning research projects.
- d) **Threat Kill Ladder** – The immune system has a partially redundant, yet specific and exact sequence for typing and resisting almost all types of antigens.
- e) **Breath of Specificity** – As documented extensively in the medical literature, the immune system has an elegant response mechanism that can specifically respond to a host of antigens.

9.4 IDARA and other Architecture Implementations

The Immunology-derived Distributed Autonomous Robotics Architecture (IDARA) is a robot coordination architecture based on the immune system that makes extensive advantage of the response structure (see also Figure 20) to vary coordination based on the specific level of information known. The novel aspect of this architecture is that it not only uses both the innate and acquired immunity models, but that it uses these models to perform both the specific (i.e., deliberative) and immediate (i.e., reactionary) processing and coordination tasks [74].

The IDARA architecture (as illustrated in Figure 22), yields an adaptive coordination method for very large robot colonies (i.e., populations in the thousands). IDARA is also robust to fair levels of noise (see also Figure 23). This architecture builds upon immunology models and other related to mimic the immune system’s control structure. Thus, it is capable of responding dynamically and efficiently to continually varying inputs. Further, an agent’s design is not constrained by traditional instability and recovery criteria, since the failure of an individual (disposable) agent is not detrimental to the entire system and may actually be beneficial to the overall action. Via this structure the IDARA architecture combines the power of classic deliberative, thorough planning architectures with the relative simplicity and rapid response of reactionary architectures in a unified framework.

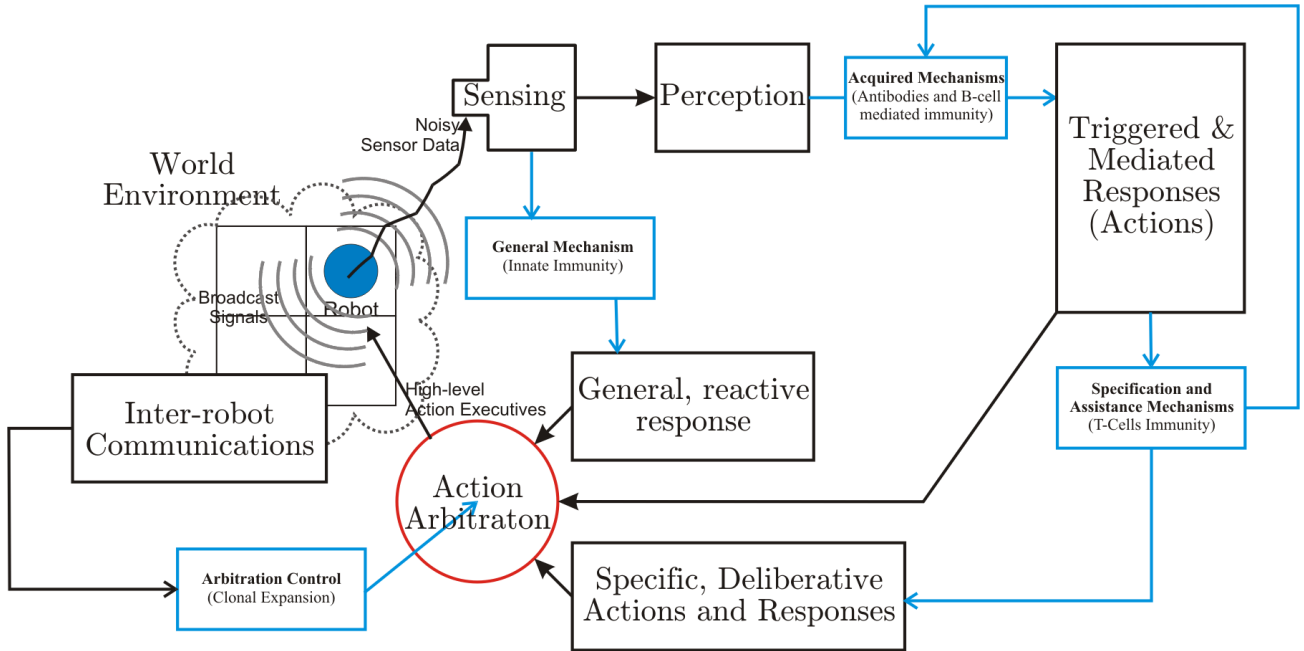


Figure 22: IDARA Software Architecture: Immunology analogous are shown in gray and, for comparison, the typical execution paths of reactive and deliberative/planner-based architectures are shown as dashed lines

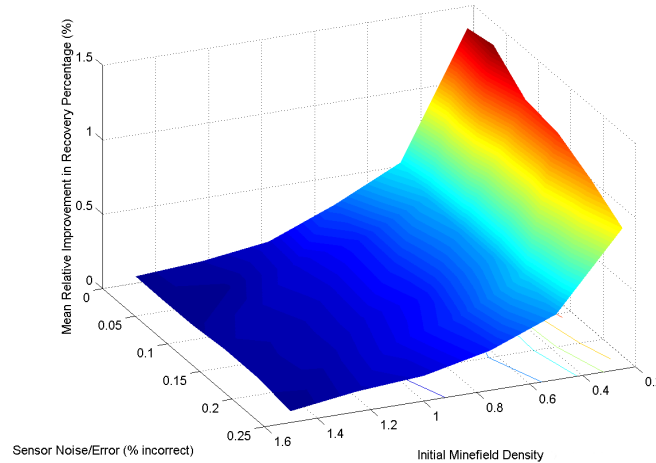


Figure 23: IDARA in a Noisy Environments: IDARA is efficient even with significant levels of sensor noise

However, IDARA has a significant weakness, namely: that interaction is initially based on Brownian motion until a threat is found and has been characterized by the architecture. By varying the center (i.e., the mean) and the distribution from which the random motions are selected, it is possible to direct and guide this stochastic behavior in a manner that some level of random exploration is maintained. Once a series of the threats has been characterized the method is the capable of generating a local gradient and using gradient decent, Levenberg-Marquardt, or a similar optimization method to follow the signals from initial interactions by general agents. This, however, predicates that there is an initial interaction between the two effectors. Thus, the architecture has an inherent “critical mass” [74]. This is illustrated in Figure 24, where the relative level of energy associated with the sparse case (of a mine field in this example) with the immunological approach almost doubles.

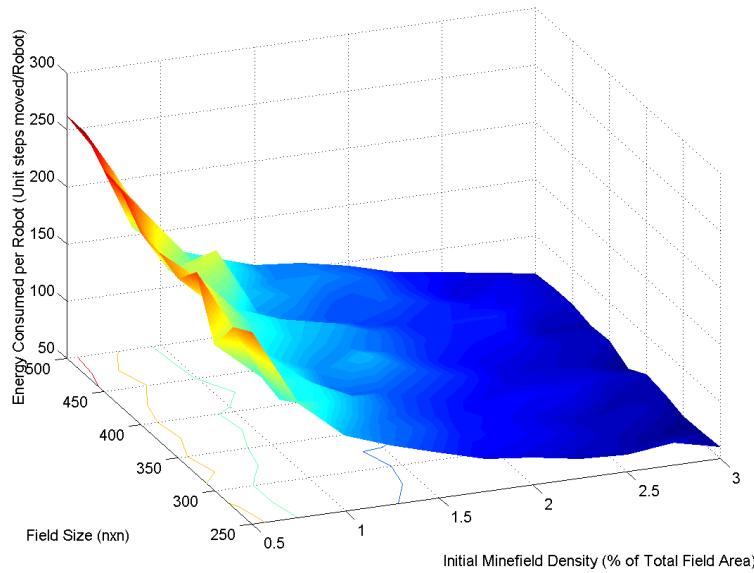


Figure 24: IDARA Energy Consumed: Sparse regions (i.e., large areas with few objects) lead to inefficiencies [74].

Earlier methods (e.g., [42],[43],[47],[54],[71]) generally used the immune system as an algorithm in a deliberative node/level or as the foundation for behaviors in a reactive model. That is, the controllers directly mapped certain cell functions to certain behaviors (e.g., “B-cell” actions would respond when a sensor input matched the “complex” for that behavior). By comparison, IDARA uses the immune system principally as a pattern recognition tool to catalyze the selection of behaviors in addition to the traditional behavioral mapping approach [74].

9.5 Comparative Advantages/Disadvantages of Immunity Based Architectures

Advantages:

- **Combined Approach** – Allows for both reactive and deliberative methods to be used in a unified manner. By allowing for the integration of various components this method gives its designer a great deal of flexibility, especially in the multi-robot domain (where some robots could spend varying amounts of time on control)
- **Massively Scalable** – In order to provide the extreme levels of protection that are essential for survival, the immune system provides exceptional mechanism for scalability. Likewise this architecture can easily scale its operations as its mechanisms are polynomial in general, and in some cases approximately linear.
- **Layers of Innate Pattern Recognition** – The architecture provides two forms of pattern recognition: a general signature model that characterizes threats and a second layer that provides “self-tolerance.”
- **Flexibility** – IDARA is a general enough coordination architecture such that it can be applied to a variety of domains (e.g., UXO sweep, exploration & mapping, and search & rescue).
- **Specificity** – This architecture has an incredibly elegant response mechanism that can very specifically respond to a host of treats.

Disadvantages:

- **Sparse “Deadlock”** – Interaction is initially based on Brownian motion until a threat is found and then local gradient is generated and processes. However, in sparse conditions the gradient cannot form and thus the search is essentially random. Thus, this architecture has an inherent “critical mass” [74].
- **Non-global Optimization (Local Minima)** – The system uses gradient fields and numerical optimization techniques. However, there is no guarantee that these will produce a globally optimal result, only that the result will be locally optimal within the time and interaction scope under consideration.

10 Technology Summary for Future Systems/Applications

10.1 Summary of Robot Workforce Issues

The social framework of an architecture is of equal if not greater importance than the control of an individual robot. This is especially true when considering workforces consisting of small microbots or Micro Electromechanical Systems (MEMS) devices. As with bees in a swarm or bacteria of an infection, the power and intelligence desired comes solely from the aggregate numbers of the population and their group behaviors. The organization and architecture, in this case, limit the size and effectiveness of population and, thus, the technology.

The dynamics of robot workforces are to some extent philosophical questions that apply to human endeavors (e.g., when to go it alone versus have teammates). The use of robot architectures in teams, especially large ones, presents a series of tradeoffs and compromises. Some of the features and concerns gained from the application of teams are as follows [9],[76]:

- **Improved system performance** – Especially in environments where the task can be decomposed into units for parallel execution, a group of robots can exploit this to yield results in less time just as a supercomputer solves problems. However, some tasks are inherently serial and thus render the team force concept essentially useless except as a relief force.
- **Task cooperation/empowerment** – Some tasks simply require teamwork and/or cooperation from teams of specialized agents. Thus a team would be able to perform tasks that are intractable for any one robot (e.g., climbing over a wall by stacking one on top of another).
- **Distributed computing/operation** – Distributed networks are generally more reliable because there is not a central point of failure. With the use of wireless networks for communication, it is also possible to route signals in to complex patterns (e.g., in an environment where A can see only B, B can see both A&C, and C can only see B, a team would be able to share communication information by routing it through B).
- **Shared sensing** – A team could share information gathered and/or provide information that is in a “blind spot” of another robot (e.g., in the same scenario as above, A could warn C of an imminent danger or concern and visa-versa).
- **Fault Tolerance** – Redundancy in the system provides reduced individual complexity and more robust performance as the redundancy can be manipulated/dynamically created (e.g., command both A and C to view the same point) to improve system reliability.

However, teams operation is not without difficulty. These include [9]:

- **Interference** – Simply: “Too many cooks spoil the broth.” Robots have to be cognoscente of the effect of their activities on other robots and should not needlessly interfere.
- **Communication and Coordination Issues** – Communication is generally necessary to enable coordination. At best this can be a distraction from the architecture and at worst it can result in failure of the system because signals are either unable to be broadcast and/or some robots may not be able to heed the instructions being sent to them.
- **Uncertainty/Error propagation** – Coordination mandates that the architecture (to some extent) monitor the activities of other robots. Furthermore, erroneous sensors can propagate errors in the models being used by the other robots (i.e., misinformation spread). Thus, robots need to also monitor/filter the sensor environment of their own and their peers sensors.
- **Complexity** – Adds an additional layer of complexity.

10.2 Comparisons

Table 1 shows a comparison of various characteristics/performance parameters for robot workforce coordination and control [9],[12],[13],[30],[47],[57],[74]. Table 2 lists some popular architectural implementations and classifies them into one of the seven principal architectures [9]. Finally, Figure 25 uses a gradient mixture spectrum plot to generally illustrate the prime application spaces of the robot coordination discussed.

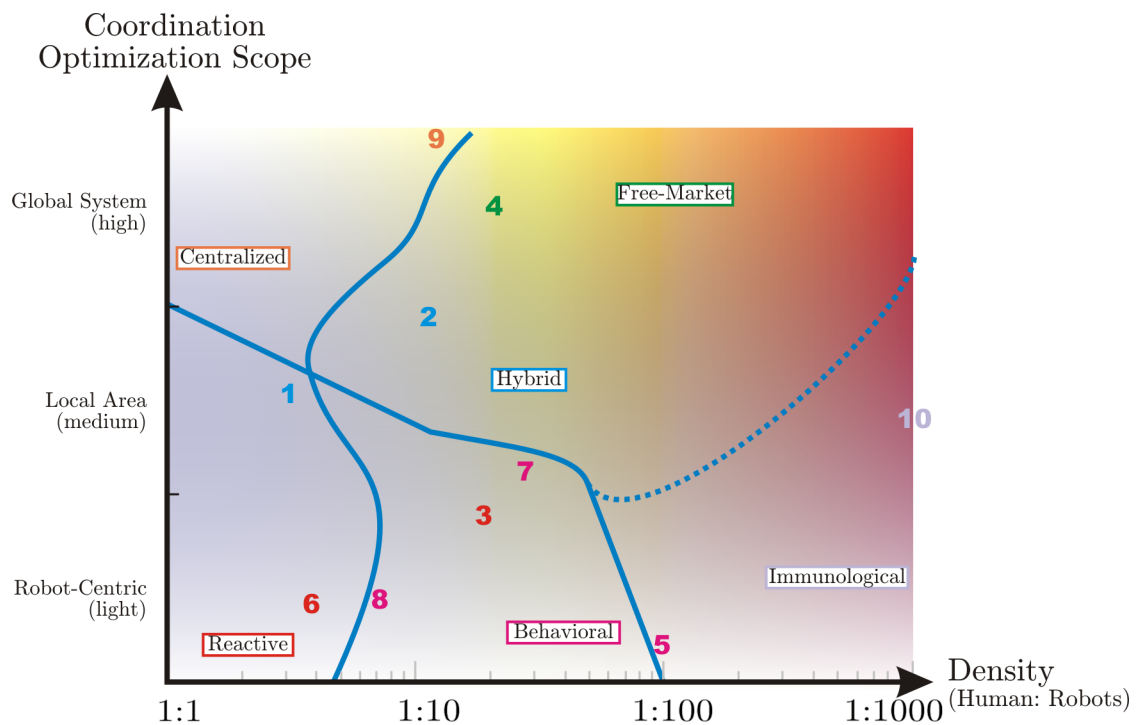
Technology Comparison Matrix:

Table 1: Property Comparison Principal Architecture Classes							
Characteristic	Coordination Mechanisms						
	Centralized	Deliberative	Reactive	Behavioral	Hybrid	Free Market	Immunology
<i>A priori</i> information needed	Not necessary, but can be utilized	Yes	No	Not Generally	Some – Especially the Spatial Reasoner Module	Some – Cost functions need to be defined	Not necessary, but can be utilized
Adaptability (i.e., operates outside training set)	Medium	Low (but high in structured environ.)	Medium	Medium	High	Medium	High
AI Reasoning/ Planning	Yes	Yes	No	Some	Yes	Yes	Some
Central Vulnerability	Yes	Maybe	No	No	No	No	No
Communications Needed/Avg. Load	Yes Heavy	No Medium	Yes Heavy	Yes Heavy	No Medium	Yes Medium	Yes Light
Computational Complexity/Order	NP/Exponential	NP/Exponential	Robots-Behaviors	Varies approx. Robots-Behaviors	Mixed, but tractable	Varies on bidding methods	N ² :Antigens
Distributed	No	Maybe	Yes	Yes	Maybe	Yes	Yes
Fault-Tolerant	No	No	Yes	Yes	Maybe	Yes	Yes
General/Instant Response	No	No	Yes	Yes	Usually	Some	Yes
Key Strength	Easy to implement	Symbolic AI logic	Robust in dynamic environments	Direct sensor-effector mapping	Combines deliberative & reactive	Free market economy model	Scalability
Key Weakness	Intractable order	Slow, rigid	Can not “think”	Limited to available behaviors	Complex structure	Non-open environments	Brownian interactions
Learning Capable	Yes	Some	No	Yes	Yes	Some	Yes
Massively Scalable	No	No	Somewhat	No	No	Yes	Yes
Optimal Solution Guaranteed	Sometimes	In enough time	No	No	Maybe	No	No
Robustness	No	No	Yes	Yes	Yes	Yes	Yes
Schema theory based	No	No	Yes	Yes	Yes	No	Some
Specificity	Varies	Varies	No	Varies w/ initial behavior set	Some	Some	Yes
System-wide Approach	No	No	Yes	Yes	Some	Yes	Yes

Architecture Matrix:

Table 2: Architecture Classifications/Comparisons

Architecture	<i>Properties</i>					
	Core Architecture	Background	Principal Design	Coordination Method	Programming	PI/Reference
Action-selection	Behavioral	Dynamic competition system	Experimental	Arbitration via action selection	Competence modules	P. Maes (MIT)
ALLIANCE	Hybrid	Multirobot, motivational behavior	Experimental	Motivation regulated subsump.	Behavioral language	Parker (ORNL)
Animate agent	Behavioral	RAP situated activity system	Situated activity	Sequencing	RAP Language	Firby (U Chicago)
Ant Colony Control	Traditional	Ant Colony Optimization methods	Ethologically /experimental	Pheromone signal	Ant Colony optimization	Smith (CMU) / [26]
Atlantis	Hybrid	Reactive and subsumption methods	Experimental	Sequencing	RAP Language	Gat (JPL)
AuRA	Hybrid	Reactive and subsumption methods	Ethologically /experimental	Deliberative driven subsumption	Parameterized behavioral libraries	Arkin (GIT) / [12]
Circuit	Reactive	Earlier reactive methods	Situated activity	Hierarchical mediation	Rex & Gapps	Kaelbling et al. (SRI)
BUGS	Behavioral	Earlier reactive methods	Guided motion(s)	Placed beacons, internal behaviors	Custom	Naval PGS / [39]
Colony	Behavioral	Subsumption architecture	Ethologically /experimental	Priority-based arbitration	Subsumption	Connell (IBM)
DAMN	Behavioral	Fine subsumption architecture	Experimental	Multiple winner-take-all arbiters	Custom	Rosenblatt (CMU)
GOFER	Hierarchal (Petri-Nets)	Indoor Automation	Experimental	Action planning and allocation	Best-first-of-received "proposals"	Latombe (Stanford) / [21]
GRASP	Hybrid	Multirobot coordination	Experimental	Sequencing	GRASP GUI modular system	Kumar (U.Penn) / [5]
IDARA	Immunity	SAIS and Behavioral methods	Biomimetic and Simulation	Best-first, greedy with adaptation	Modular units	Singh & Thayer (CMU) / [74]
Large-Scale Assembly	Hierarchal (Layered units)	Heterogeneous layers with explicit coordination	Experimental	Distributed coordination via task trees	Task Description Language	Simmons (CMU) / [73]
Microeconomics	Free-Market	Multiagent negotiation and bidding	Economic contract bidding/exchange	Profit maximizing	Layered execution	Dias & Stentz (CMU) / [30]
Motor Schemas	Reactive	Reactive engine for AuRA	Ethologically	Summation and normalization	Parameterized behavioral libraries	Arkin (GIT) / [13]
Occupancy Grid Maps	Centralized	Exploration of uncharted environments	Simulation	Grid maps used to set plans and prevent overlap	Value iteration	Thrun (CMU) / [20]
Planner-reactor	Hybrid	Situational methods	Anytime planning	Planner-reactor	Custom	Lyon & Hendricks
Ranger – Scout	Centralized	Multirobot surveillance	Experimental	Client-Server	Modular with OTS system integration	Rybski, et al. (UMn) / [66]
Ropes	Centralized	Flexible manipulation for robots	Manipulation Theory	Strings between robots	Custom	Böhringer (Dartmouth) / [16]
SSS	Hybrid	Subsumption architecture	3-stage hybrid	Model world representations	Subsumption	Connell (IBM) / [27]
Subsumption	Reactive	Earlier reactive methods	Experimental	Competitive (arbitration-inhibition & suppression)	AFSMs, Behavioral language	Brooks (MIT) / [19]
Territorial Task Division	Behavioral	Tropism-Based Cognitive Arch.	Experimental	Deliberative driven subsumption	Behavioral libraries	Materic (USC) / [70]
UGV Demo II	Hybrid	DAMN	Ethologically /experimental	Deliberative driven subsumption	Parameterized behavioral libraries	Arkin (GIT) / [12]
UGV Demo III (4-D/RCS)	Hierarchical	UGV II, VaMoS 4D	Reasoning layers	Higher layers	Parameterized behavioral libraries	Albus / [3]



Legend:

- 1 Arkin - AuRA
- 2 Balch - MultiAgent Control
- 3 Brooks - Subsumption-based reactive control Approach
- 4 Dias and Stentz - Microeconomic approach
- 5 Hodgins and Brogan - Dynamic group behavior(s) modeling*
- 6 Lumelsky - "The Cocktail Party" Architecture
- 7 Materic - Biologically inspired reinforcement learning ("Nerd Herd")
- 8 Parker - L-ALLIANCE
- 9 Rybski, et al. - Ranger/Scout
- 10 Thayer and Singh - IDARA*

* Simulation results indicated

Number



Coordination



Figure 25: Spectrum comparing the prime application areas for various architectures [12],[18],[49],[61],[66],[74]

11 Military / Defense Implementations

11.1 Overview

The use of collaborative robotic technology has many significant military and defense applications. Research into and implementations of the use of robot workforces and their coordination has been done for at least the following military scenarios: Field reconnaissance [76], Demining [12], and teams of unmanned aerial vehicles/unmanned ground vehicles (i.e., UAV/UGV) [9].

Collaborative robotics is an enabling technology for FCS necessary to realize many of the higher-level objectives (i.e., reduced crew size and logistics footprint, more precise synchronization in maneuver and fires coordination, and reduced command and control burdens for commanders and operators of robotic forces). Traditional systems often require a single unit to be a generalist, capable of performing all tasks as well as possible [30]. By comparison, FCS collaborative robotics must assume a diverse collection of specifically equipped and enabled units that can specifically respond with much more decisiveness than any general purpose platforms. Through collaborative robotics, multiple units will be able to accomplish a task more efficiently and with more precision because collaborative robotics is the underlying technology that [40]:

- Enables teams of autonomous robots to actively collect and maintain timely and accurate situational awareness
- Facilitates control of multiple robotic units by a single operator
- Allows for dynamic deployment of robotic resources depending the environment
- Provides layers for standardized communication
- Offers a standard extrasensory mechanism; that is, for information exchange (and potentially force projection) through the sensors and effectors of other (external robots)

As mentioned in the aforementioned descriptions of robot coordination architectures, several military application domains are catalyzed, if not made possible, via the use of these architectures. As future military systems advance in automation and complexity these methods have risen to the forefront of robotics research and design. Development and engineering efforts in this area have focused on several fronts, namely: robot teaming software (including development tools, optimization standards, etc.), robot team hardware, and dynamic *ad-hoc* networking.

11.2 Robot Team Software Developments and Technologies

Software architecture development has been a central focus of the military robotics research. There has been significant interest in architectures and methods that define a central framework for robot operation and communication [both external and internal (i.e., between stages or nodes of the system)]. In particular, two significant development and demonstration efforts include the JAUGS distributed computing program (outlined in [87]) and the 4-D/RCS unmanned vehicle-sensing and operation system/architecture [1],[84].

11.2.1 Joint Architecture for Unmanned Ground Systems (JAUGS)

JAUGS is a reference architecture development effort being led and coordinated by the Department of Defense (DoD) Joint Robotics Program (JRP). As such, JAUGS is quite broad especially due to the spectrum of autonomy and operational scenarios within the defense oriented unmanned ground

systems community [35]. The JAUGS architecture itself is not a design or implementation scheme; rather, it is a framework for the design of architectures with the following design requirements [83]:

- Vehicle/platform independence
- Mission independence/isolation
- Computer hardware independence
- Technology independence

In essence, JAUGS is essentially a broad architecture specification primarily concerned with implementation and control issues necessary to ensure interoperability and exchangeability of unmanned ground systems and robots. That is, a team of JAUGS robots would not only have standardized communications (e.g., TCP/IP) but a standardized *modus operandi* (i.e., basis of operation). This along with the common interface allow for improved interpretability and portability as not only can features be upgraded simply, but also new options can be added [87]. Further, the JAUGS architecture is centered on the JAUGS Domain model (see also Figure 26) [83]. This model outlines the requirements for the user (“in the language of the user”), defines system topology (i.e., operational systems, nodes, components), specifies interface issues, provides requirements on methods for message passing, and standards to support component integration [87].

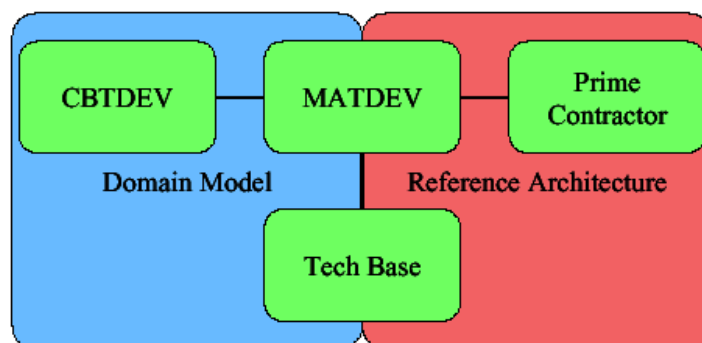


Figure 26: JAUGS General Structure [83]

The use of JAUGS will be mandatory in all robot systems built or developed for the JRP. The JAUGS “domain model,” in its current form, focuses on the remote driving of large unmanned ground vehicles and not small human-level or micro-robot technologies [35]. Further, the JAUGS model also does not allow for separation or distinction based upon internal and external system operations, which is an important aspect in multi-robot coordination. However the specification is still being designed and revised (by the JAUGS Working Group for Standard Operating Procedures) and currently only constrains solutions that are within the “area of compliance” (i.e., those specifications of a robot system that are outlined by the JAUGS architecture) [83].

The JAUGS model has both its strengths and weaknesses. Some of the strengths of the JAUGS architecture are: an architecture open to all of the robotics community, reduction of development time/life cycle costs, a focus on “new” requirements/engineering system, and a framework for technology insertion [87]. However, some of the weakness of the JAUGS system (in its current form) are: that it is too highly linked to the deliberative architecture/model, it does not have enough flexibility to properly handle architectures without world models (e.g., reactive), and it appears applica-

ble only to large ground systems and not the numerous multi-robot domain examples under consideration in this paper [35],[84].

While it is not clear what the final JAUGS architecture will specify and which multi-robot military systems it will impact, it is likely that robots and architectures such as the Demo III system will be impacted by JAUGS. As detailed in section 11.3.1, this ground vehicle system is a prime example of the benefits gained by a standard architecture and platform as this will allow interoperability and coordination between multiple types (and makes) of vehicle spun off from this research project.

11.2.2 4-D/RCS

A second significant software architecture development is the 4-D/RCS reference model architecture for the design and development of intelligent vehicle systems and software. In particular, 4-D/RCS combines the National Institute of Standards and Technology (NIST) real-time control system with the German VaMoRs 4-D approach to dynamic computer vision [1],[31]. Designed in part as a theoretical basis for future standards, 4-D/RCS is an open system architecture that will facilitate the integration of a wide variety of subsystems [3].

The 4-D/RCS architecture is described in detail in section 4.2 (page 11). In summary, this method uses a layer of nodes to provide increasingly more sophisticated levels of intelligence for operation. Within 4-D/RCS, levels of collaboration and multi-robot cooperation can be integrated via the addition of another superior node layer that coordinates the activities of the nodes immediately under its command [4]. That is, the coordination nodes would operate with a broad, supervisory scope and not a high-level of micromanagement. Finally this architecture has been a rich development history starting with its foundations in the 1980s with NASREM, it has been used successfully in the German VaMoRs autonomous vehicle project and, most recently, in the Demo III project (see also section 11.3.1).

11.2.3 Sandia's Modular Architecture for Robotics and Teleoperation (SMART)

Unlike the 4-D/RCS and JAUGS robotic control architectures, is a software architecture for building telerobotic systems from commercial hardware that achieves maximum utility from the hardware and minimum complexity for the end-user. The SMART architecture consists of three modules (the Editor, Supervisor, and Real-time Engine) to enable rapid application development (RAD), eases design compliance to standards (e.g., JAUGS), and guarantees stability of operation via the use of classical control stability criteria as defined in the component models that are developed as part of SMART. Its system approach allows for checks and optimization of the entire system and a unified environment to assemble and compile control and coordination codes to yield the desired behavior [15].

While not making direct use of robot coordination architectures, the SMART architecture is of importance to robot coordination because it demonstrates RAD and system-wide development tools. This would not only allow an operator to easily orchestrate robots or collections of robots in a workforce, but also to “tweak” parameters of semi-autonomous robots for certain types of performance. It is envisioned that future versions of a multi-robot SMART could integrate the aforementioned robot coordination architectures, which would allow for intelligent systems logic and planning in the control of “classic” telerobotic systems [15], [3].

11.3 Robot Team Hardware Developments and Technologies

In addition to software system, robot control architectures have been implemented and advanced in several hardware robotic systems. Some prominent examples, which are described below, illustrate the influence robot architectures can have on the operation and utility of robotic systems. The Mobile Detection Assessment Response System (MDARS) and its coordination and control architecture, Multiple Robot Host Architecture (MRHA), are a good example of a flexible robot system. The MDARS robot is a military research effort to field interior and exterior autonomous platforms for security and inventory assessment at military warehouse sites [35].

The indoor and outdoor robots have been specialized for particular operation scenarios and the security tasks that are to be completed. For example, the indoor robots have been in research/development for over ten years under the auspices of improving the security guard force effectiveness. Using preprogrammed (*a priori*) patrols and an enhanced obstacle avoidance system, the robots were able to not only perform security tasks (i.e., detect intruders and emergency events [i.e., smoke, fire, flood, etc.]), but also track tagged inventory with nothing more than supervisory operation from professional security guards. The external version of the MDARS uses platforms from GDRS (General Dynamics Robotics Systems) that have been equipped with autonomous navigation, collision avoidance, and intruder detection capabilities [41],[33].

The control of both types of MDARS is performed via the MRHA method. The MRHA is a distributed multiprocessing system that provides coordinated control of multiple autonomous vehicles from a single host console. The increased functionality of the base platform required the addition of sensor equipment and accompanying distributed processing. In particular it uses centralized methods for coordinating behaviors and this is limited to small teams (i.e., up to twenty sentry robots). The MRHA controller performs navigation using a multi-tiered approach consisting of five steps, namely: sensor fusion and arbitration, optical re-referencing, uncertainty modeling, states, and concurrent processes. MRHA has been successfully demonstrated and is capable of supporting a mix of interior and exterior platforms to offer site commanders optimal capability and flexibility in an automated security solution [33],[35].



Figure 27: MDARS (External) and Demo III UGV



Figure 28: MDARS user environment

11.3.1 Demo III XUV (Experimental Unmanned Vehicle)

The Demo III project is an ongoing DoD robotics research effort that is examining technologies being developed and evaluated for reducing hazards faced by military forces while enabling them to move more rapidly and nimbly deploy themselves to distant locations with reduced logistical complications. Being led by the Army Research Laboratory, Demo III is the cornerstone of the JRP UGV Technology Enhancement and Exploitation program [72].

The Demo III research effort is highly coupled to this architecture to the 4-D/RCS architecture for control and coordination as Demo III is the experimental vehicle for this architecture [4]. Being based on this hierarchical architecture, Demo III sensor and hardware development were tweaked and designed with standardized implementation and operator ease of use (see also Figure 28) as prime concerns. As such, this effort is seen as critical for reliable, deployable autonomous vehicle operation in the near future. Further, the use of layers of autonomy present in the 4-D/RCS architecture allows for various forms of technology integration and transfer to other military systems; however, it does come at the expense of having an architecture that is highly deliberative and thus not having the speed and simplicity of reactive, behavioral, or hybrid control models [72].

The 4-D/RCS architecture classifies intelligent activity, regardless of complexity, as one of four elements, namely: observation, registration (against a knowledge database) and model perception, value judgment, and behavior generation. This integration of complex response from simple perceptions is largely possible due to the layered nature of the architecture which allows autonomous operation to be constructed in a modular, device-independent manner on top of well-defined sensor-perception-action nodes [4].

Primarily meant as a scout mission vehicle, Demo III uses an upsized MDARS (external operation) vehicle platform with a diesel hydrostatic power train. It has had two previous demonstrations (October, 1999 and October, 2000) with a third one scheduled for October 2001 expected to show off-road, semi-autonomous operation at speeds up to 32 km/hr [4].

Demo III is also able to use the higher multi-robot coordination layers of the 4-D/RCS architecture. In particular, DEMO III will be able to (on operator command) form and maintain a group formation as it operates autonomously [72],[82].

11.3.2 BUGS (Basic UXO [unexploded ordnance] Gathering System)

Another type of military robot system, BUGS uses the robot workforce model to yield significant results from essentially insignificant robots and provides an excellent vehicle for testing more modern control and coordination strategies (e.g., behavioral, free-market) [35].

Developed by the Naval Explosive Ordnance Technical division, BUGS uses dozens of cheap small robots to gather surface-litter UXO and sub-munition parts. These robots then as a collective team coordinate their activities to pile the ordnance in collection areas for later processing and destruction [39]. The high-level supervision is performed by a team of skilled explosives ordnance (EOD) technicians who perform additional perception tasks such as pre-surveying the area (including general marking of obstacles), confirming UXO operations, and rescuing a robot from either a system anomaly or similar error [35]. Through the use of more involved hybrid robot coordination methods, BUGS is able to use autonomous and semi-autonomous operational modes to reduce the danger and drudgery associated with the recovery of dangerous materials such as unexploded ordnances.

BUGS also exploits multi-robot coordination efforts in its UXO operating scenario. While more advanced scenarios are under possible, the expected procedure is one that involves an EOD technician, a Remote-Controlled Reconnaissance Monitor (RECORM) advanced UXO sensing and characterizing robot, and a series of BUGS robots for “pick up and carry away” (PUCA) operations (see also Figure 29). It is expected that the EOD operator (through tele-operations via the RECORM) will place radio beacons at the most probable UXO locations. The BUGS series of robots, which are equipped with effective (but low-cost [inaccurate]) sensors, then proceed to move in a point-to-point manner so as to approach the closest beacon. Upon finding UXO the robots then progress to a marked UXO stockpile. Obstacle avoidance is achieved through simple veer left and veer right maneuvers. The coordination methods in BUGS do not apply multi-member coordination nor to they perform any significant deliberative processing steps (i.e., they are dependent on the good placement of beacons for successful operation) [39].

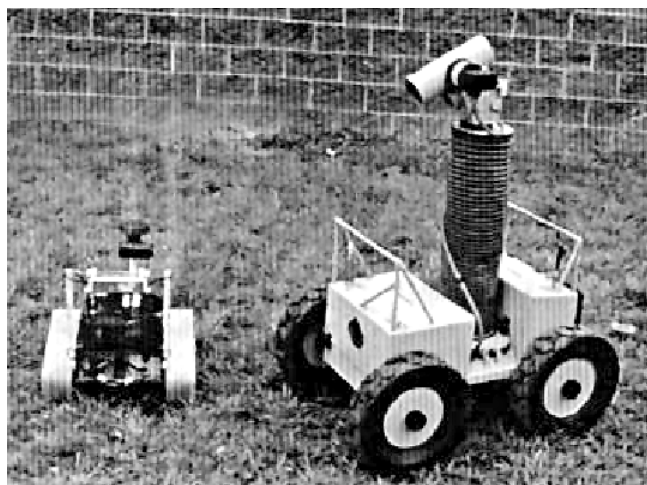


Figure 29: BUGS and RECORM robots in the field [85]

11.3.3 Rangers and Scouts

The Ranger and Scouts (R&S) project at the University of Minnesota uses a hierarchal robotics team that can execute complex missions in both military and civil environments. Through the use of a centralized architecture the R&S project has produced and demonstrated a 10:1 (robot: human) easily controllable system that can conserve resources (e.g., battery power) and accomplish a variety of tasks [66].

The key to this system is its two-tier approach of lead robots (rangers) and small agents (scouts). This allows the system to gain the general utility of a traditional large (fast) mobile robot platform along with the covertness and mobility of small, miniature robots [88].

The miniature robots of the R&S project are the scout robots (see also Figure 30). These robots are easily deployable and are meant to act as the sensory leads for of this system. Although fairly small, they are able to traverse obstacles and operate in uncertain environments in a controlled manner. Mechanically it has two forms of location, namely: a set of roller wheels and a spring “foot” which it can use for jumping. The scouts perform sensing and communications using OTS electronics. Furthermore, the scouts are specialized by their sensing capabilities which include passive infrared (IR), CMOS camera, condenser microphone, vibration sensing, and gas sensing [66].

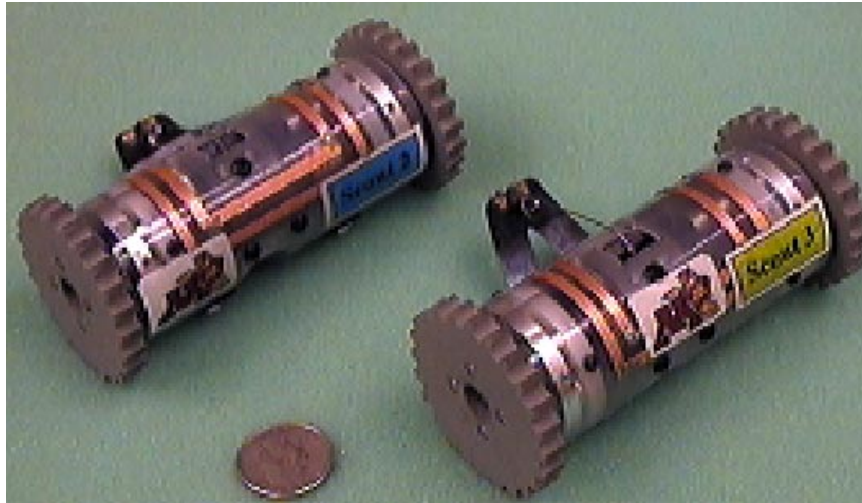


Figure 30: Scout robots [66]

The deployment of the scouts and larger robotic activities is accomplished via the rangers (see also Figure 31). In addition to acting as a utility and launching platform for the scout robots, these rangers communicate and coordinate their activities with other rangers. The hardware of the rangers is based on the ATRV-Jr™ platform to which a custom camera and launching mechanism has been added. Finally, the ranger serves as a communications and coordination hub for scouts under its command [66].



Figure 31: Ranger robot (and launcher) [66]

The coordination and control of this hardware is generally performed through centralized methods, which control (via activation or deactivation) behaviors being executed by the scouts and rangers. Basically, the scouts send sensory information back to the ranger. The ranger (using coordination method on-board along with operator instructions for the scouts received via tele-operation) then communicates the appropriate behavior actions/parameters.

This form of coordination simplifies the communication of tasks and compensates for the relatively poor video quality that comes from the low point of view (the scouts are usually on the ground) and communications noise (the images are processed off-site by the rangers). Some of the behaviors included with the scouts include: goal/target location, motion detection, goal seeking motions, and collision avoidance.

The coordination of the rangers is performed through the combination of traditional centralized methods and tele-operation. In addition the rangers use a simpler version of behavioral control to assist with motions/navigation in the environment (e.g., door/obstacle identification, etc.). The results of this project, especially the hardware platform, are impressive. The coordination of the robots is not as automatic nor sophisticated as is possible via the application of the algorithms and architectures discussed in previous sections. Finally, it is interesting to note that one of the greatest weaknesses found with the system (and its operation) was power consumption, which can be addressed (in addition to the use of more efficient electronics) via the integration of more involved coordination methods [66].

11.4 DARPA TMR

DARPA's Tactical Mobile Robots (TMR) project is interested in the development and application of man-portable, adaptable robotic platforms for dismounted operations. With regards to overall force and equipment, the use of small robots is envisioned to fill the gap between autonomous and semi-autonomous large-scale maneuverability forces and the need for wide-range sensor coverage and first-response actions [48].

As this is a developing area no one predominant operational architecture has surfaced. The TMR project fielded several robots on September 2000 in a reconnaissance and rescue mission. However, of the 47 robots fielded, many of them include advanced sensor fusion and behavioral/hybrid software technologies [82].

As the TMR program does not specify a standard architecture or method (unlike the JAUGS structure of the JRP) there is much more variance in the designs and applications being researched and developed [87]. It is expected that Phase II of the TMR project will see the refinement of these technology and a major field technology demonstration. In addition to operation in dense forest/vegetation (a difficult challenge in and of itself) the robots are expected to operate in urban and semi-urban environments; that is, they are capable of climbing stairs, estimate position (with and without GPS), and discriminate objects through walls. Finally, one of the many areas that TMR technology can be applied is the much-discussed Military Operation in Urban Terrain (MOUT) scenario. While consisting of numerous flavors, it generally involves the discovery and neutralization of potential threats in unknown complex urban environments. For example, this might include finding which locations within a commercial building an hostile force may have captured and acting in a manner to prevent the force from leveraging these captured positions to launch offensive actions [48]. Several of the TMR inspired robot hardware and software architectures can be applied to this situation. For example, both MDARS and Rangers and Scouts along with other multi-robot architectures have been tested for use in this application .

11.5 Summary

Robots, especially in high-risk situations, will act as “force multipliers” and allow warfighters to project force more directly with less personal less risk. By integrating novel coordination and control methods along increasing levels of automation, military robots will reduce the burden and risk for future warfighters. This will permit the soldiers to provide high-level supervision for generally autonomous multi-robot systems customized for their respective tasks [60].

Using collaborative robotics as the basis for all autonomous teaming operations in FCS, the autonomous and semi-autonomous robot systems of the future will collaborate effectively with both humans and other robots. Further, many new military technologies and systems (e.g., manned platforms, Micro UAV, MOUT, Lemmings tracked vehicles, etc.) will not only extend the scope and reach for coordinated robotics, but will be able to leverage collaborative technologies to maximize their potential and scope.

By enabling collaboration (i.e., communication, synchronization, specification, and task-optimization), the “cells” in FCS are able to support and switch between a variety of platforms or operational scenarios in a robust and seamless manner. Some of the teaming applications enabled by FCS collaborative robotics are: forward reconnaissance, security, terrain mapping, long-duration surveillance, logistics and support), path discovery, construction of local common operating picture (L-COP), autonomous maneuverability, forces maneuverability, etc.

As alluded to earlier, the force of the future will rely on robotic technologies [82]. These military applications (and FCS in particular) will reap tremendous benefit from the use of collaborative robotics initiatives. These include, but are not limited to:

- Mission execution without a single point of failure. Ultimately enabling a force with multiple platforms fighting as a network centric entity.
- Highly specific deployment and response that is more effective and proactive than traditional systems and is a key part of the new warfare of FCS.
- Deductive levels of force multiplication
- Super data collection and integration - by fusing the data streams from individual sensor array it is possible to generate final, combined results with higher quality (i.e., noise and variation) than may be present in any one data stream.

12 Potential Impact and Bearing of Collaborative Robotics

12.1 Overview

As illustrated in Figure 32, Collaborative robotics is an integral part of the Future Combat Systems (FCS) strategy and vision. The “objective force” will require collaboration and intelligent Command and Control (C2) so that heterogeneous forces of varying levels of autonomy and specificity will be able to act together in a cohesive manner. In a nutshell, collaboration is a framework for robotic teamwork; and teamwork will be necessary for the optimal performance and impact demanded of FCS and its forces.

It is envisioned that various forms of collaborative robotics will be deployed at all levels of operations ranging from the high-level (potentially multi-national) force collaboration to unit C2. At each level of operations, collaborative robotics methods will further FCS technologies and result in new levels of force effectiveness by not only leveraging the unique strengths of particular units, but also providing a common framework for exchanging critical information so that actions are precise, focused, and synchronized [40].

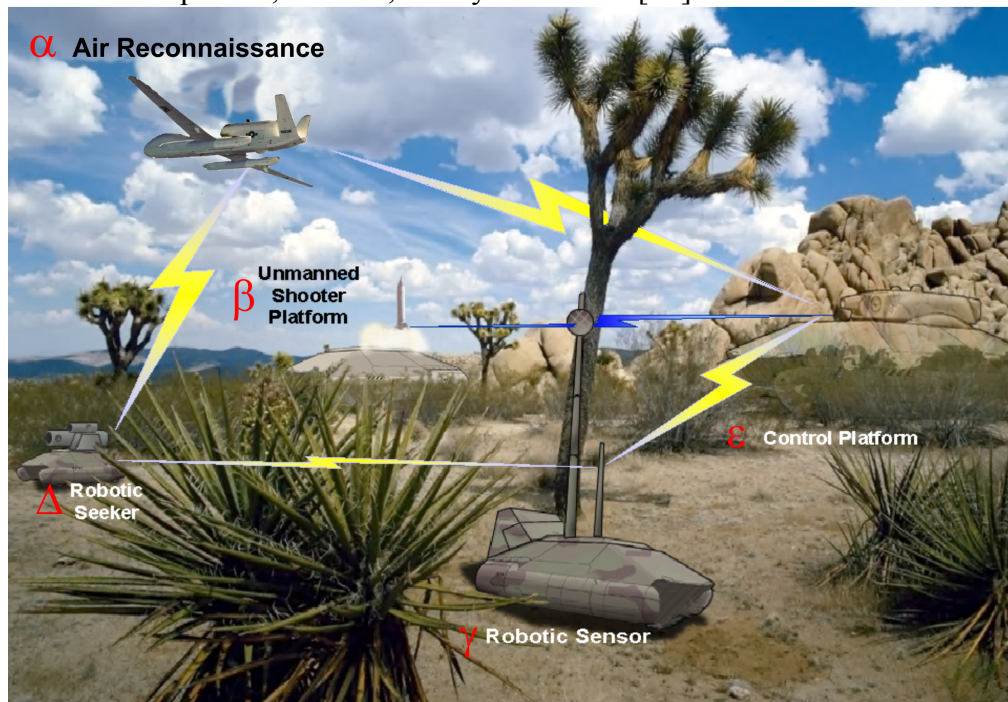


Figure 32: Illustration of a hypothetical FCS “unit cell” (based on [24])

12.2 Coordination Level Description and Illustration

Future forces will consist of a spectrum of specialized, heterogeneous members. The use and application of robotic and multi-team coordination at each FCS echelon (see Figure 33) will result in unique forms of operation and coordination that are impossible in platform-centric approaches.

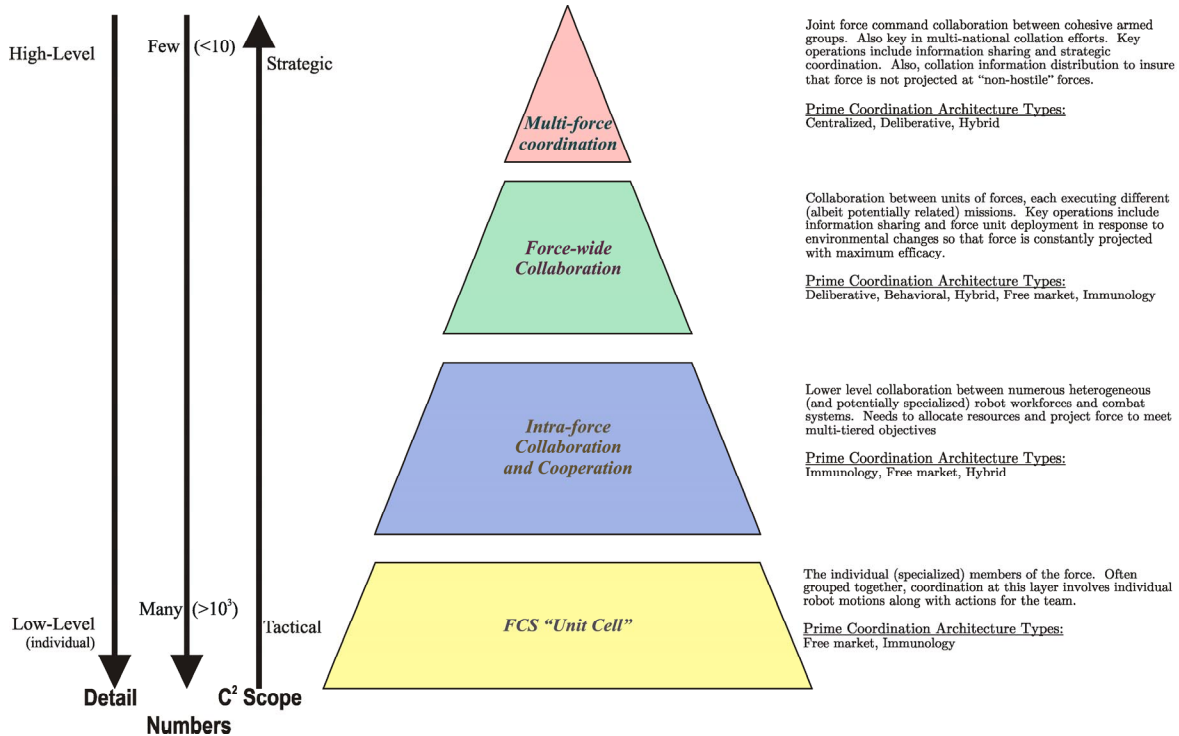


Figure 33: Envisioned FCS Coordination Echelons
(each level of the hierarchy has unique command and control aspects)

12.2.1 Intra-force FCS

Principally seen at the battalion level of organization and below, this is the primary point in FCS where heterogeneous, cooperative and collaborative control becomes necessary. Collaborative architectures at this level must be configured for the mission in the same manner that the FCS early entry forces will be deployed. Further, it is expected that the engineering of the specifics of the architectures at this level will vary amongst FCS modalities - air, land, sea, space – but that they have a common framework and/or communications layer, as this will be critical for higher levels of collaboration [84].

The collaborative efforts at this step will include task decomposition, sharing and exchange of individual tasks, synchronization activities, and collaborative plans. For example, a hybrid architecture maybe employed on each of the individual robot war fighters. In this case each robot would act in a relative manner until deliberative/hierarchical control is engaged to realign the sensor-action reward. For example, in the situation illustrated in Figure 32, a sensor robot would work quietly while avoiding obstacles and local disturbances until it sensed a hostile threat or was engaged by another robot (i.e., sensor space).

12.2.2 Force-wide Collaboration

The second level of collaboration is between force systems as part of a cohesive, concentrated, and highly lethal force projection. At this level the collaborative methods are far more hierarchical and employ greater levels of planning and strategic deliberation. It simply does not need to worry about “minutia” such as obstacle avoidance, interference, and action deliberation.

For example a free-market or immunity-based architecture may be employed at this level of collaboration. This would not only allow for the coordination of large workforces to yield very-large scale command and control. In the case illustrated by Figure 32, this level of collaboration would more or less guide the group and may involve the reconnaissance aircraft instructing the ground-based sensor and kill FCS firepower entities to concentrate efforts on a particular enemy force or cell [76].

Further the collaborative efforts at this level would utilize current and developing secure communication technology (i.e., spread-spectrum communication, cryptographic methods, etc.) to ensure secure, robust performance. The architectures would be error tolerant thus preventing an error from cascading through the network and disabling it. Finally, the coordination architectures will employ mission, not self, optimization/maximization. This will allow for self-sacrificial behaviors and collection and transmission of information following a disaster [74].

12.2.3 Multi-force coordination

Multi-force coordination is one of the highest forms of coordination and encapsulates the concepts detailed in intra-force and force-wide collaboration. The purpose of this form of collaboration is to aid as an assessment for tactical decisions being made by strategic command and command centers.

This form of control recognizes that future combat will involve joint force Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR). As such it would provide a means for various forces to coordinate their activities without the direct C2 of a joint force command. For example, via this form of coordination, Air Force or National Security Agency (NSA) reconnaissance data would automatically be sent to relevant Army ground FCS systems and Navy surface and maritime systems. This makes the total force more nimble and mobile as replication can be minimized to the extent needed for redundancy and parallel operations. The key, however, for this form of coordinated effort is a strong/standardized communications and coordination layer along with robust data analysis tools. That is, the problem is still collecting information from a deluge of vague messages none of which are uniquely specific enough [10]. Through the use of coordination, it (along with data-mining) may be possible to coordinate the data retrieval process and to share threat assessments so that hidden patterns emerge soon enough that a minor preemptive strike or a “shot across the bow” may be all that is necessary to eliminate the threat [35].

This level of coordination is also important in multi-national/collation efforts as the collaborative system needs to be aware and cognizant of action being made by Joint or Coalition forces. For example, to use local radars/guidance systems that may actually be part of a coalition partner’s army to ensure the highest-level of accuracy and pinpoint lethal force. Furthermore, this form of coordination will be needed generate self/non-self recognition algorithms employed by autonomous collaborative architectures at the intra-force and (to some extent) force-wide collaborative levels.

12.3 DARPA / Army Impact

Underlying collaborative robotics technologies are methods for not only optimal team activity/participation, but also information/sensor fusion from a distributed network of agents. As such, many of the collaborative robotics technologies under consideration have impact, relevance, and/or utility to many of the current DARPA/Army and military research thrusts under investigation. Appendix 2 outlines and illustrates DARPA projects that can either benefit from collaborative methods and/or whose technology could significantly impact collaborative robotics [28].

12.4 Sample Collaborative Applications

It is envisioned that collaborative robotics architectures will enable a wide array of heterogeneous multi-robot/multi-system cooperative interactions. Some of the envisioned applications of this technology are described in the following subsections; however, as with any primary infrastructure such as collaborative methods, it is quite likely that some of the most novel applications have yet to be discovered/developed.

In a nutshell, automated coordination architectures provide a means by which a task can be distributed amongst a team of robots which share pertinent task information and work as a team to yield results not obtainable by simple division of labor. To some extent the robot teams, which may consist of thousands of individuals, are using their collective capabilities to gather, sift, and share insights gleaned from the volumes of incoming data which is collected by various, sometimes disparate systems.

12.4.1 Urban/Intra-Building Reconnaissance

Urban warfare is one of the most challenging environments faced by future war fighters [48]. Unlike “open” field combat, the environment is often poorly surveyed, fully of uncertainties, and intermixed with friend, foe, and bystanders. A long-standing research problem, many projects have investigated and developed methods for autonomous, robotic navigation and map reconstruction. These include, for example, sonar map and robust Monte-Carlo reconstruction methods [81], [80].

In this case collaborative robotics technology would be more interested in integrating the data collected from the various surveyor/prospector robots/technologies in a manner that maximizes the information quality from each individual unit. That is, it is envisioned that each method/algorithm for reconstruction is particular apt at certain reconnaissance tasks and, furthermore, has some statistical properties (e.g., distribution, mean, variance, etc.) that best describe this operation. The collaborative architecture would then be responsible for coordinating and determining which methods are employed by certain robots, especially in environments where robot heterogeneity leads to certain robots being particularly adept at particular forms of reconnaissance, mapping, or reconstruction.

Multi-robot technology provides significantly more degrees of freedom as the system could explore multiple paths simultaneously. The IDARA architecture has been used to show how a stochastic search can still be generally directed to yield a robust mapping system that is robust to erroneous guide Nance and sensor noise. This is illustrated in some of its results on mapping interior domains, which are shown in Figure 34.

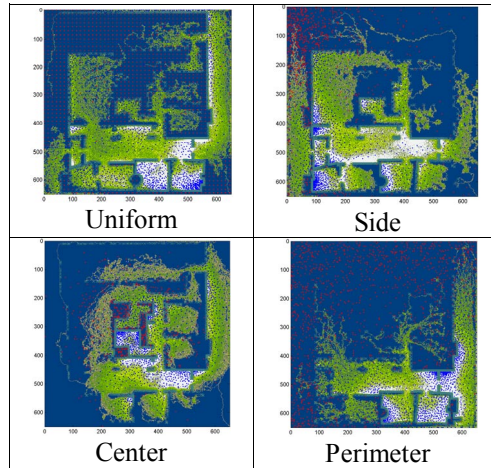


Figure 34: Comparative Visitation Maps

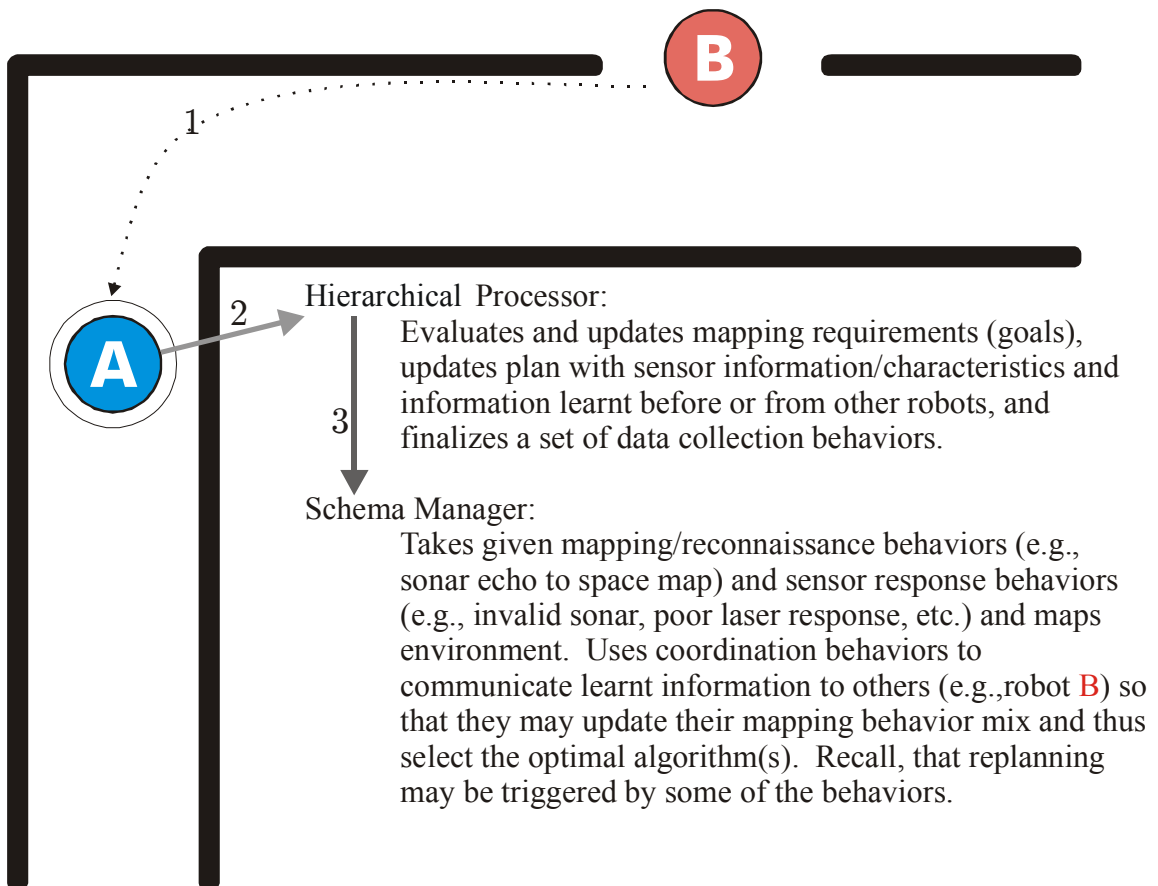


Figure 35: Hybrid architecture and mapping

For example, consider a system governed by a hybrid collaborative architecture such as AuRA [12]. In particular, the hybrid controller on each robot would be responsible for setting the actions/behaviors on a particular robot and the coordination actions between robots. As illustrated in Figure 35, the planner on each robot would use the information it has about the environment, its sensor array(s), the mapping/reconnaissance task, and knowledge it has received from other robots as it selects sensor-behavior pairs [7]. After this step, the hierarchal planner then would pass control to the reactive or subsumption controller on the robot. This stage of the controller would govern

the operations and data collection. Included in the sensor/behavior pairs would be actions to send collaborative information (e.g., sonar status, relative laser reflectance, etc.) to other robots. This would update neighboring robots (as needed) with information that has been gathered thus providing feedback and coordination for higher-level parameters such as which algorithm(s) would be most effective and which sensor parameters/gains are necessary for optimal operation [12].

An alternative approach would be the use of methods developed and demonstrated by Stenz and Dias using their free-market robot architecture [30]. In this case, a particular robot having a specific set of sensors would exchange/bid on task sub-components in an effort to maximize its “profit.” The result of this architecture would be that each robot would run the algorithm(s) its sensor arrays are particularly well suited for and then would share (through negotiation and contracts) the task so that in general it is working in a manner that maximizes its comparative advantage(s) at the task. For example a robot with a sonar array would bid for tasks in which the sonar information would have a high probability of resulting in valid map/model recreation [76].

Collaborative robotic technology also allows for advanced and improved sensor fusion and integration. That is, it allows a robot to collect, filter, and integrate information collected by sensors that are external to the robot in a bandwidth-efficient manner. In particular, collaborative methods could enable the use of often-neglected data streams that maybe present, especially in traditional densely constructed urban environments (e.g., closed-circuit camera arrays, architectural drawings and CAD models, card-key system, etc.). Further, it would employ statistical techniques to insure that inconspicuous or errors in any one form of the information stream do not detrimentally impact the results of the system [76]. The following example details how intelligent control would integrate “external” sensors to yield a more complete map:

1. Upon being directed to initiate a survey, the robotic team begins by assessing its internal capabilities (i.e., sensors mobility characteristics of each robot)
2. A general response is orchestrated based upon any prior knowledge available (e.g., architectural drawing, class of building, type of situation, enemy threat/armament). This response is then updated via data that is collected as the robots traverse the environment.
3. Assuming traditional communications are possible, the system would communicate and coordinate with building systems for access to areas of the building and to gain additional forms of perception from other robots and sensors in a manner that results in a positive return for the collaborative efforts expended. Furthermore, the system would use probabilistic data (e.g., data distribution, mean, variance, etc.) to not only be resilient to sensors/model errors, but to characterize and potentially correct for erroneous data in the system.
4. This would result in a more accurate map of the environment. Further, the coordination system makes the robot more effective and capable of obtaining the data needed to support the mission in progress.

In summary, collaborative robotics technologies will enable a much more effective urban reconstruction and will allow for complex levels of sensor integration and robot specialization all in a manner that makes the system resilient against cascading error propagation and data outliers. By allowing for more efficient data collection and information capture/discovery, robotic coordination enables a new level of control where perception models can be potentially checked and corrected in real-time.

12.4.2 Distributed Intermittent Jamming

Distributed Intermittent Jamming (DIJ) is a special form of a distributed communications system arrays. This tactical system uses a large number of radio-frequency (RF) emitting cells to sub-task and spread the processes associated with the broadcast of jamming signals amongst a wide array of points that intermittently operate in both the time and frequency domains. Enabled by the rapid evaluation and advances of MEMS and robot collaboration technology, this system is able to jam communications in both the time and frequency domains and thus provide an agile, high-impact system with high survivability regardless of the physical rate of mobility for the system [10],[46].

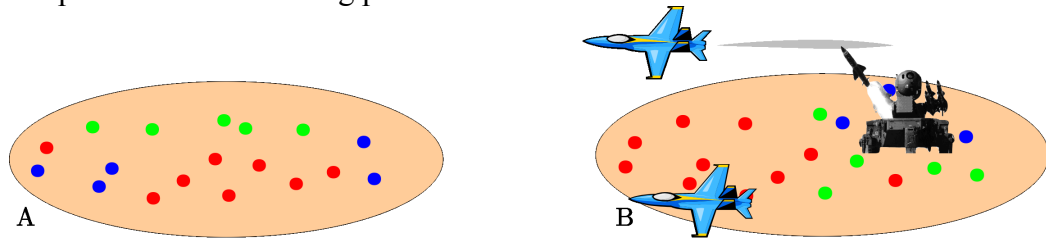
The point of this jamming technology is not to destroy, but rather, to annoy, which is tactically more valuable than a complete shutdown. By crippling (but not outright destroying) the infrastructure, significant communications are prevented; however, the hostile force is now not only confused, but also distracted, especially if attention and resources are diverted to continue using the now compromised network. Unlike traditional jamming methods which often make use of a centralized source (e.g., a fighter with radar jamming circuitry), DIJ relies on a ground based array of points that coordinate their intermittent actions to thwart hostile communications and/or guidance systems without placing highly-sensitive aircraft systems at risk. Finally, because a DIJ system is basically a “set-and-forget” technology, it has a much lighter logistical footprint as it does not need a support staff, hanger, and other necessities associated with active air-combat [46].

Collaborative robotics technologies are essential to realizing this class of autonomous weapon system which delivers the necessary high-impact “objective force” in a light, survivable, and autonomous manner. The parallel design facilitated by these methods allows for the delivery of a combined force (whose capability is not dependent on any one member) which yields distributed force that can achieve near perfect reliability even though individual points are unreliable, vulnerable, and easily captured. The distributed architecture has high survivability from all but extreme counter-measures as there is no central point for mounting a reaction nor is there a singular signal that systems could be designed around [35].

While a variety of collaborative methods could be applied, such a large scale distributed system would benefit from the large scale control and operations afforded by the immunity-based architecture. The architecture would provide a means by which the jamming robots could coordinate their activities and effectively block (as necessary) communications and/or hostile weapon guidance systems. This architecture also supports the intermittent strategy as points could be reassigned to critical frequencies and/or time-patterns as needed to counteract perceived threats. An example of how an immunity-based architecture could achieve the coordination necessary for the DIJ system is illustrated in Figure 36 and described as follows:

- An array would be distributed by airborne delivery via traditional methods, an UAV (unmanned air vehicle) or a munitions system.
- This points in the array would respond to stimuli in environment and use the methods of immunity-based architectures to optimally reallocate their methods of response. In particular, hostile systems, such as the tank, could be considered “antigens” and friendly forces, such as the fighter aircraft, could be “helper T-cells.”

- The response is then shifted towards specific forms of jamming as needed.
- The coordination methods would update depending on the current risk assessment.
- As points are lost, the cause for loss would be estimated and this information used to refine the response of the remaining points.



Legend: • Wide band jamming • Time intermittent jamming • Frequency intermittent jamming

A. Initially the distributed jamming robots are dispersed in an area and take some general state

B. This system monitors its environment adjusts to the system it is trying to disable so that measures with the highest probability of success are undertaken. That is, the type and modality of the jamming change with the treat (e.g., infra-red [IR] for Stinger missiles and radar-blocking for Rapier platforms).

C. Using collaborative robotics technologies the robots switch their strategy to best respond to the environment/situation. In this example, the jamming system responds so that resources are conserved yet tracking and navigation of enemy surface to air armaments are overwhelmed and rendered useless against the (friendly) fighter aircraft.

Figure 36: Distributed Intermittent Jamming

12.4.3 Collaborative Guidance for Munitions

One of the guiding visions of FCS is that future forces employ all information available to deliver tactically mobile, survivable forces that deliver overwhelming firepower for decisive victory.

Traditional systems are generally constrained to information that is gathered or obtained via internal perception and communications systems. While highly effective, these systems are often forced to trade compactness for sophistication and generally are incapable of perceiving and utilizing data obtained from neighboring or other systems. Further, they are often characterized by a heavy logistical/support footprint [46].

By allowing for the integration and use of information obtained from other battlefield munitions and systems, robotic coordination technology and methods can provide significant improvements FCS armaments that will make them more accurate, lethal, and mobile – in other words an “objective force.” In addition the provide warfighters a more lethal and effective system as coordination assures that the best positioned units or members will be engaged.

Robot coordination makes several types and forms of coordinated munitions possible. Some of the principal forms include engagement coordination, tactical coordination, and multi-system battlefield coordination. Using policy (ROE) and teamwork optimization, these forms of coordination go beyond simple sensor fusion to yield the highly intelligent results characteristic of FCS systems. These results are detailed as follows:

- *Engagement Coordination* – The engagement tasks are shared by the robot team and information from the various range of sensors and robots is integrated between the robots to assess and characterize a potentially hostile threat. This includes sharing partial information in

order to fully assess a threat and using combined data streams to perform friend/foe determinations with very high accuracy in approximately real-time. For example, if the reconnaissance aircraft (shown as α in Figure 32) were to initially observe some threat, the plane would coordinate with the other robots (such as the seeker [Δ]) to characterize the concerned object (i.e., do other robots register it as a threatening or hostile member). After this the workforce determines what or how to share and prioritize workforce resources to engage the hostile force. More advanced coordination would integrate strategy (i.e., an estimate of the hostile forces response) in the engagement plan.

- *Tactical Coordination* – If guided to continue, the robot workforce then shares sensor and perception data to fully characterize the threat and the munitions need to deliver a swift, lethal, and stealthy impact to minimize (if not totally destroying) any oncoming threat. For example, a small, smart munitions may be followed by a high-energy weapon that uses the guidance and navigation of the smart weapon as a beacon for its own path. A second example, is the use of general munitions to weaken and redirect an enemy so that it is more vulnerable to a second, in this case devastating, weapon.
- *Multi-System Battlefield Coordination* – The coordination architecture is used by various robots on the field to tune the specifics of their actions and responses. For example, this could range from a sensor robot (e.g., γ) acting as cover for other armaments (e.g., β and ϵ) to a system where the specific skills of each robot are exploited as the workforce shares a mission and its requirements.

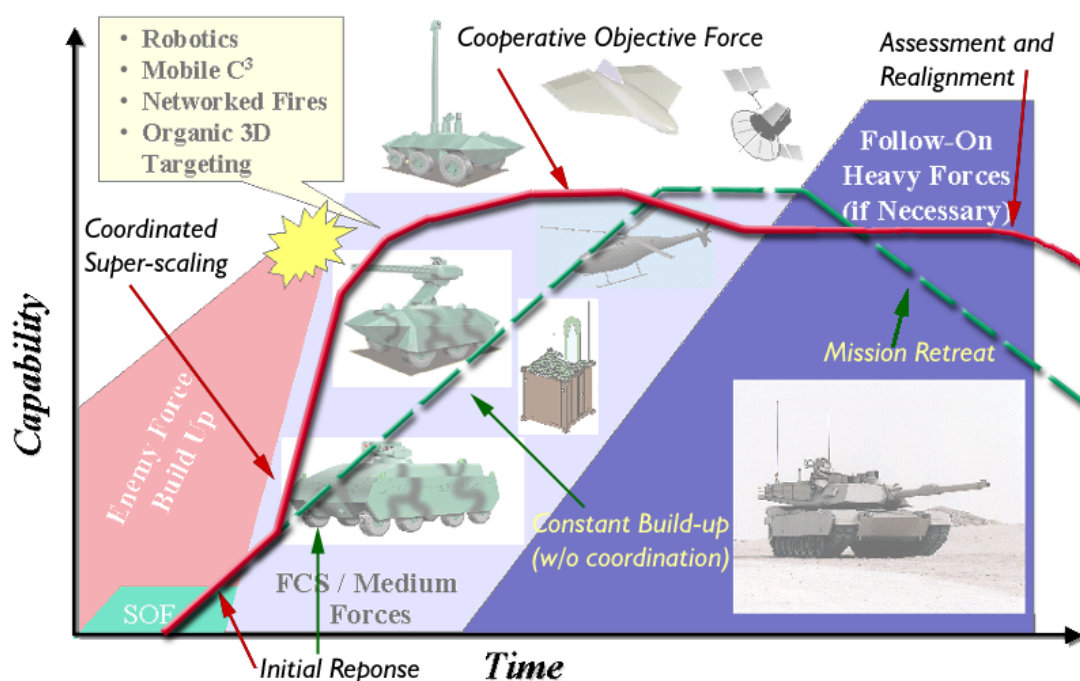
12.5 Summary

Traditional systems often require a single unit to be a generalist, capable of performing all tasks as well as possible. By comparison, FCS collaborative robotics assumes a diverse collection of specifically equipped and enabled units that can specifically respond with much more force and command than any general unit. Through collaborative robotics agents will be able to accomplish tasks more effectually and with more precision. Some of the principal fundamental capabilities afforded by this technology to FCS unit cells include following:

- Enables teams of autonomous robots to actively collect and maintain timely and accurate situational awareness
- Facilitates control of multiple robotic units by a single operator who will issue broad “high-level” commands (e.g., secure this area) instead of the micromanagement typical by present-days systems.
- Execution of inherently team oriented tasks. Some tasks (e.g., moving a heavy object) require coordinated motion of at least two team members
- Simplified strategic command and direction. As actions are coordinated a commander can issue higher-level, more “natural” commands (e.g., “platoon survey and secure this area”).
- More efficient consumption of resources (communications, armaments, fuel, etc.). By maximizing the group’s outcomes for a set level of resources, the efficiency increases. This, in turn, results in a more mobile and agile force (i.e., improved logistics) and one that is less vulnerable (i.e., less errors lower risk of enemy retaliation or detection).

By having collaborative robotics as the basis for all autonomous teaming operations in FCS, the autonomous and semi-autonomous robot systems of the future will collaborate with both humans

and other robots. By enabling collaboration (i.e., communication, synchronization, specification, and task-optimization), the autonomous teams in FCS are able to support and switch between a variety of platforms or operational scenarios in a robust and seamless manner. Some of the teaming applications enabled by FCS collaborative robotics, namely: unmanned platforms, Micro UAV, MOUT, collaborative operations (i.e., forward recon., security terrain mapping, long-duration surveillance, logistics and support), path discovery, construction of L-COP, etc.



13 Conclusions

- Robot workforces will be key in shifting the emphasis in the principals of war towards leaner, more capable forces: the principal enabler in synchronized coordination.
- There are a range of collaborative technologies, but no one “catch-all” method. Thus, FCS systems will need to employ the various architectures at their optimal points of application.

- Collaborative methods will not only empower the warfighter with new capabilities, but will allow for the completion of the mission with reduced risk.
- By enabling new forms of reconnaissance and guidance, these technologies allow a smaller, more mobile unit to deliver a more objective and lethal force.

The collaborative methods described and detailed are more than a mere technological component to the FCS system; rather, they enable many of the complex interactions and inter-echelon operations that will characterize and realize the FCS vision. The reason for collaborative method is simple – they allow autonomous and semi-autonomous robots of various levels of specificity to interact and operate as a cohesive and decisive force. Furthermore, collaborative robotics will allow the increasingly joint (and perhaps multi-national) forces to leverage their individual strengths to the utmost thus furthering the impact of future tactical systems in their deployment around the world. In a nutshell, collaborative robotics is an integral part of the FCS vision and will just as integral as the “smart-munitions” which it controls.

Robotic coordination technologies are essential for maximizing the utility and power of these systems. As this survey has shown, there is no “one-size fits all” answer to the robotic coordination problem, rather a host of solutions each particularly apt for a certain type/aspect of the coordination problem. We recommend that coordination methods used by multi-robot systems adapt and vary with the scope (task level) of their point application, in the manner shown in Table 3

Table 3: Architectures and Prime Application Areas	
Principal Activity (i.e., Level of application)	Optimal Architecture Algorithm
Cell execution	Free-markets
Distribution	Immunology
Strategic	Deliberative
Survivability	Reactive
Tactical	Hybrid

NEXUS:

With the continued pace of research and the thrust that is underway This system will be were there is a clear comparative advantage. It is proposed that this will be mature by 2010 and that the scope of the existing set of programs may need to be modified to be FCS relevant. Collaborative robotics in academia and industry will build upon DARPA, Army, and JRP the current focus for effective architectures.

BOTTOM LINE:

Collaborative robotics extends FCS’s precision and survivability to levels of operation and efficiency hitherto not obtainable. With continued research and application of novel coordination architectures, the FCS force will have the robustness and performance necessary to decisively and resolutely maintain control over any form of combat ranging from dense, urban combat to open, wilderness/terrain.

14 Acknowledgments

This research was made possible through support from the Defense Advanced Research Projects Agency (DARPA) TTO – Future Combat System (FCS). The authors greatly acknowledge the generosity and leadership of this organization. The authors further acknowledge the assistance and co-operation of the Future Combat Systems Team Gladiator (TRW, Lockheed Martin, Battelle Memorial Institute, CSC). The authors also thank Dr. Wesley Thayer (M.D.) for his guidance and expertise with respect to molecular immunology aspects of the Immunology Based Architecture.

This report was produced as part of the Design Concepts Phase of the FCS research by Team Gladiator under contract number MDA972-00-90-004 and was originally prepared the DARPA Contracts Management Directorate in July, 2001.

15 References

- [1] Albus, J. S., “4-D/RCS Model Architecture for Unmanned Ground Vehicles”, *Proceeding of SPIE Conference on Unmanned Ground Vehicle Technology {Orlando, Florida, USA}*, April 1999, pp. 11-20.
- [2] Albus, J., Lacaze, A., Meystel, A., Theory and Experimental Analysis of Cognitive Processes in Early Learning, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* (1995), Volume 5, pp. 4404-4409.
- [3] Albus, J.S., “4-D/RCS Reference Model Architecture for Demo III”, *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, September 1998, pp. 634-639.
- [4] Albus, J.S., “4-D/RCS Reference Model Architecture for Unmanned Ground Vehicles”, *Proceedings of the IEEE International Conference on Robotics and Automation* (2000), No. 4, pp. 3260-3265.
- [5] Alur, R., Das, A., et al., “A Framework and Architecture for Multirobot Coordination”, *Seventh International Symposium on Experimental Robotics*, 2000.
- [6] Arbib, M., “Schema Theory”, *Encyclopedia of Artificial Intelligence*, Second Edition, S. Shapiro Editor, New York: Wiley, 1992.
- [7] Arkin, R. and Becky, G. (Eds.), *Robot Colonies*, Boston: Kluwer Academic Publishers, 1997.
- [8] Arkin, R.C. and Balch, T., “Behavior-based Formation Control for Multiagent Robot Teams”, *IEEE Transactions on Robotics and Automation*, December, 1998.
- [9] Arkin, R.C., *Behavior-Based Robotics*, The MIT Press, Second Printing, 1999.
- [10] Aviation and Missile Research, Development, and Engineering Center (AMRDEC), *AMRDEC Briefing to Industry and Future Combat Systems*, July 2000.
- [11] Baeten, J. C. (Ed.), *Applications of Process Algebra*, Cambridge: Cambridge University Press, 1990.
- [12] Balch, T. and Arkin, R.C., “AuRA: Principles and Practice in Review,” *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 9, No. 2/3, pp.175-188, 1997.
- [13] Balch, T. and Arkin, R.C., “Communication in Reactive Multiagent Robot Teams”, *Autonomous Robots*, Vol. 1, No. 1., 1994, pp. 27-52.
- [14] Balch, T., Khan, Z., and Veloso, M., “Automatically Tracking and Analyzing the Behavior of Live Insect Colonies”, *Agents 2001 Conference {Montreal, Quebec, Canada}*, June 2001.
- [15] Bennett, P. C. (of Sandia National Laboratories), “SMART Website”, Online: http://www.sandia.gov/isrc/Capabilities/Integration_Technologies/SMART/smart.html
- [16] Böhringer, K., Brown, R., Donald, B., Jennings, J., and Rus, D., “Distributed Robotic Manipulation: Experiments in Minimalism”, *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 1995.
- [17] Brooks, R. A., “A Robust Layered Control System for a Mobile Robot”, *IEEE Journal of Robotics and Automation*, RA-2, April 1986, pp. 14-23.
- [18] Brooks, R. A., “Elephants Don't Play Chess”, *Robotics and Autonomous Systems*, Vol. 6, pp. 3-15, 1990.
- [19] Brooks, R. A., “New Approaches to Robotics,” *Science*, Vol. 253, September 1991, pp. 1227-1232.
- [20] Burgard, W., Simmons, R., Thrun, S. et al., “Collaborative multi-robot exploration”, *Proceedings of the IEEE International Conference on Robotics and Automation* (2000), pp.476-481.
- [21] Caloud, P., Choi, W., Latombe, J-C., Le Pape, C., and Yim, M., “Indoor Automation with Many Mobile Robots”, *IEEE/RSJ International Conference on Intelligent Robots and Systems* (1990), pp.67- 72.
- [22] Cameron, J.M.; MacKenzie, D.C.; Ward, K.R.; Arkin, R.C.; Book, W.J., “Reactive Control For Mobile Manipulation”, *Proceedings of the IEEE International Conference on Robotics and Automation* (1993), vol. 3, pp. 228-235.
- [23] Cassandra, A. R., “POMDPs for Dummies: POMDPs and Their Algorithms, Sans Formula!” Online: <http://www.cs.brown.edu/research/ai/pomdp/tutorial/>.

- [24] Chiodo, P. E., "Army Quality Engineering Directorate Presentation: Quality, Reliability, Safety", FCS/DARPA Gladiator June 12-13 Briefings, June 2000.
- [25] Christensen, H. and Pirjanian, P., "Theoretical Methods for Planning and Control in Mobile Robotics," *Proceedings of the 1997 First International Conference on Knowledge-Based Intelligent Electronic Systems*, May 1997. pp. 81-86.
- [26] Cicirello, V., and Smith, S., "Ant Colony Control for Autonomous Decentralized Shop Floor Routing", *Proceedings of the 5th International Symposium on Autonomous Decentralized Systems*, 2001, pp. 383-390.
- [27] Connel, J., "SSS: A Hybrid Architecture Applies to Robot Navigation", *Proceedings of the IEEE International Conference on Robotics and Automation (1992)*, pp. 2719-24.
- [28] DARPA, "DARPA Technical Offices Website" Online: <http://www.datpa.mil/body/darpoff.html>
- [29] Dasgupta, D. and Atttoh-Okine, N. "Immunity-based Systems: A Survey", *IEEE International Conference on Systems, Man, and Cybernetics*, Vol.1, 1997, pp. 369-374.
- [30] Dias, M. B. and Stentz, A., "A Free Market Architecture for Distributed Control of a Multirobot System", *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pp. 115-122, July 2000.
- [31] Dickmams, E., et al., "The Seeing Passenger Car 'VaMoRs-P'", *International Symposium on Intelligent Vehicles {Paris, France}*, October 1994.
- [32] Dyer, J. "ARDEC Modeling and Simulation", FCS/DARPA Gladiator June 12-13 Briefings, June 2000.
- [33] Everett, H.R., "Multiple Resource Host Architecture for the Mobile Detection Assessment and Response System," Technical Document 3026, Space and Naval Warfare Systems Center, San Diego, CA, August 1998. (Online distribution via: <http://www.spawar.navy.mil/sti/publications/pubs/td/3026/td3026.pdf>)
- [34] Fukuda, T., Shibata, T., et al., "Selfish and Coordinative Planning for Multiple Mobile Robots by Genetic Algorithm", *Proceedings of the 31st IEEE Conference on Decision and Control*, Vol. 3, pp. 2686-2691.
- [35] Gage, D., "Current Thrusts in Ground Robotics: Programs, Systems, Technologies, Issues", *Fourth International Symposium on Technology and the Mine Problem*, March 2000. (Online distribution via: <http://www.spawar.navy.mil/robots/pubs/demine00.pdf>)
- [36] Ganong, W., *Circulating Body Fluids*, Fourteenth Edition, Appleton Lange Publishers, 1989.
- [37] Guyton, A. and Hall, J., *Textbook of Medical Physiology*, Ninth Edition, W.B. Saunders, 1996.
- [38] Harrison, L., *Anti-Aircraft Systems*, Online: http://home.clear.net.nz/pages/juni0r/anti_aircraft.html.
- [39] Healey, A., McMillan, S., et al., "BUGS: Basic UXO Gathering System", *Autonomous Vehicles in Mine Countermeasures Symposium {Monterey, CA}*, April 1995.
- [40] IIT Research Institute (Alexandria, VA), *Future Combat Systems Concept Paper: Team Gladiator*, ver. 1.4, DARPA: Contracts Management Directorate, November 27, 2000.
- [41] Inderieden, R.S., et al., "Overview of the Mobile Detection Assessment and Response System", DND/CSA Robotics and KBS Workshop, St. Hubert, Quebec, October 1995.
- [42] Ishiguro, A., Kondo, T., Shirai, Y., and Uchikawa, Y., "Immunoid: an Architecture For Behavior Arbitration Based On The Immune Networks", *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3, 1996, pp. 1730 - 1738.
- [43] Ishiguro, A., Watanabe, R., and Uchikawa, Y. "An Immunological Approach to Dynamic Behavior Control for Autonomous Mobile Robots." *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, 1995, pp. 495 - 500.
- [44] Jennings, J., Kirkwood-Watts, C., "Distributed Mobile Robotics by the Method of Dynamic Teams", *4th International Symposium on Distributed Autonomous Robotic Systems*, 1998.
- [45] Jerne, N. K., "The Immune System", *Scientific American*, Vol 259, July 1973, pp. 52 - 60.
- [46] Johns Hopkins University Applied Physics Laboratory (APL), "APL FCS Mobile Code", *APL Briefings*, September 22, 2000.
- [47] Lee, D., Jun, H., et al. "Artificial Immune System for Realization of Cooperative Strategies and Group Behavior in Collective Autonomous Mobile Robots", *Proceedings of the Fourth International Symposium on Artificial Life and Robotics (AROB) {Oita, Japan}*, January, 1999.
- [48] Lopez, R., "DARPA Develops Tactical Robots for Dirty and Dangerous Missions", *Unmanned Systems*, January/February 2001, pp. 12-13.
- [49] Mataric, M. J., "Behavior Based Robotics", *MIT Encyclopedia of Cognitive Sciences*, Ed: R. A. Wilson and F. C. Kiels, MIT Press, April 1999, pp. 74-77.
- [50] Mataric, M. J., Class Notes to *CSCI 445: Introduction to Robotics*, Online: <http://www-scf.usc.edu/~csci445/>.
- [51] Mataric, M., "Coordination and Learning in Multirobot Systems", *IEEE Intelligent Systems*, March/April 1998, pp. 6-8.
- [52] Mataric, M., "Minimizing Complexity in Controlling a Mobile Robot Population", *Proceedings of the IEEE International Conference on Robotics and Automation (1992)*, pp. 830-835.
- [53] Mataric, M., "The Basics of Robot Control", Online: <http://www-robotics.usc.edu/~maja/robot-control.html>.
- [54] Meshref, H. and Van Landingham, H., "Artificial Immune Systems: Application To Autonomous Agents", *IEEE International Conference on Systems, Man, and Cybernetics 2000*, Vol. 1, 2000, pp. 61-66.
- [55] Miller, G. A., "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information", *The Psychological Review*, vol. 63, 1956, pp. 71-97.
- [56] Mitchell, T. M. and Thrun, S., "Explanation-Based Neural Network Learning for Robot Control", *Advances in Neural Information Processing Systems V*, Hanson, Cowan, and Giles (eds.), Morgan-Kaufmann Press, 1993, pp. 287-294.

- [57] Mitchell, T., *Machine Learning*, WCB McGraw-Hill, 1997.
- [58] Moore, A., "Lecture Notes on Advanced AI concepts: Decisions, Diagnosis, Cooperation, Learning and Planning in uncertain worlds," Carnegie Mellon University Computer Science Course 16-731, Spring 2001.
- [59] Noreilis, F. and Chatila, R., "Plan Execution Monitoring and Control Architecture for Mobile Robots", *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 2, April 1995, pp. 255-66.
- [60] Parker, L and Balch, T, "Guest Editorial", *Autonomous Robots Special Issue on Heterogeneous Multi-Robot Systems*, July 2000.
- [61] Parker, L. E., *L-ALLIANCE: A Mechanism for Adaptive Action Selection in Heterogeneous Multi-Robot Teams*, ORNL/TM-13000, October 1995.
- [62] Parker, L., "Task-oriented multi-robot learning in behavior-based systems", *IEEE/RSJ International Conference on Intelligent Robots and Systems (1990)*, vol. 3, pp. 1478-1487.
- [63] Quinlan, J. R., "Decision Trees and Multi-values Attributes," Machine Intelligence II, Hayes, Michie, and Richards (Eds.), Oxford, England: Oxford University Press, 1998, pp. 305-318.
- [64] Rabiner, L., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, Vol. 77, No. 2, February 1989, pp. .
- [65] Russel, S. J. and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice-Hall, 1995.
- [66] Rybski, P.E., Papanikolopoulos, N.P., Stoeter, S.A., et al., "Enlisting Rangers and Scouts for Reconnaissance and Surveillance", *IEEE Robotics & Automation Magazine*, Volume: 7 Issue: 4 , Dec. 2000, pp. 14 -24.
- [67] Saffiotti, A. Konolige, K., and Ruspini, E., "A Multi-Values Logic Approach to Integrated Planning and Control", *Artificial Intelligence*, Vol. 76, No. 1-2, July 1995, pp. 481-526.
- [68] Sandholm, T. and Lesser, V., "Equilibrium Analysis of the Possibilities of Unenforced Exchange in Multiagent Systems," *14th International Joint Conference on Artificial Intelligence (IJCAI-95) {Montreal, Canada}*, 1995, pp. 694-701.
- [69] Sandholm, T., "Automated Negotiation", *Communications of the ACM: Special issue on Agents in E-commerce*, 42(3), 1999, pp. 84-85.
- [70] Schneider-Fontan, M., Mataric, M.J., "Territorial Multi-Robot Task Division", *IEEE Transactions on Robotics and Automation*, 14:5, October 1998, pp. 815-822.
- [71] Segel, L.A. and Bar-Or, R. L., "Immunology Viewed as the Study of an Autonomous Decentralized System." *Artificial Immune Systems and Their Application*, Ed: D. Dasgupta, 1999, pp. 65 – 88.
- [72] Shoemaker, C.M. and Bornstein, J.A., "The Demo III UGV Program: A Testbed for Autonomous Navigation Research", *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, September 1998, pp. 644-651.
- [73] Simmons, R., Singh, S., Hershberger, D., Ramos, J., and Smith, T., "First Results in the Coordination of Heterogeneous Robots for Large-Scale Assembly", *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2000.
- [74] Singh, S. P. N. and Thayer, S. M., "Immunology Directed Methods for Distributed Robotics: A Novel, Immunity-Based Architecture for Robust Control and Coordination", Submitted to: *Proceedings for SPIE – Mobile Robotics Conference 2001*.
- [75] Stallings, W., *Data and Computer Communications*, Fifth Edition, Upper Saddle River, NJ: Prentice Hall, 1997, Chapter 7.
- [76] Stentz, A. T. and Dias, M. B., "A Free Market Architecture For Coordinating Multiple Robots", CMU Technical Report: CMU-RI-TR-99-42, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1999.
- [77] Stentz, A., "Optimal and Efficient Path Planning for Partially-Known Environments", *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1994.
- [78] Stentz, T. and Dias, M. B., "Dynamic Mission Planning for Multiple Mobile Robots", *Proceedings of the IEEE International Conference on Robotics and Automation (1996)*, No. 3, pp. 2396-2401.
- [79] Team Gladiator, "Design Concepts Phase", Future Combat Systems (FCS) Concept Paper, Nov. 27, 2000.
- [80] Thayer, S., Digney, B., et al., "Distributed Robotic Mapping of Extreme Environments", *Proceedings of SPIE: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*, Vol. 4195, November, 2000.
- [81] Thurn, S., Fox, D., Burgard, W., and Dellart, F., "Robust Monte Carlo Localization for Mobile Robots," *Artificial Intelligence*, Summer, 2001 (preprint dated February 28, 2001 used in advance of publication).
- [82] Toscano, M., "UGVs: Postured for Military Operations In the 21st Century", *Unmanned Systems*, Volume 18, No. 4, July/August 2000, pp. 16-19.
- [83] Unmanned Ground Vehicles/Systems Joint Project Office (Redstone Arsenal, Alabama), *Joint Architecture for Unmanned Ground Systems (JAUGS) Technical Architecture*, Vol. 2 (Reference Architecture Specification), September 25, 2000. (Online distribution via: <http://www.jointrobotics.com/Jaug/index.html>)
- [84] US Department of Defense Joint Robotics Program (JRP), *JRP Master Plan: 2001*, US Department of Defense, 2001. (Online distribution via: <http://www.jointrobotics.com/Jaug/index.html>)
- [85] US Department of Defense Joint Robotics Program (JRP), *JRP Master Plan: 1999*, US Department of Defense, 1999. (Online distribution via: <http://agdec2.technion.ac.il/telerobotics/DoD/Centers/robotics/>)
- [86] Varian, H. R., *Microeconomic Analysis*, Third Edition, W. W. Norton & Company, 1992.
- [87] Wade, R. L., *The Joint Architecture for Unmanned Ground System: Tutorial*, DoD: Missile Research Development and Engineering Center, July 1996. (Online distribution via: <http://www.jointrobotics.com/Jaug/index.html>)
- [88] Weisbin, C., Blitch, J., Lavery, E., et al., "Miniature Robots for Space and Military Missions", *IEEE Robotics and Automation*, 6:3, September 1999, pp. 9-18.

16 Appendix 1: Current DARPA Projects Impacted by Collaborative Robotics

DARPA Office ¹	Project Name	Additional Explanation
ATO	Airborne Communications Node (ACN)	Autonomous communications architecture
ATO	Buoyant Cable Array Antenna (BCAA)	Adaptively combine signals from multiple, interconnected antenna elements. Robotic coordination technology (RCT) methods would be used to minimize vulnerability and/or performance losses associated through open communications and other forms of thermal and sonic probing.
ATO	Center of Excellence for Research in Oceanographic Sciences (CEROS)	
ATO	Command and Control (C2)	
ATO	FCS Command and Control	
ATO	FCS Communications	
ATO	Project Genoa	RCT would aid pre-crisis assessment, characterization, and mitigation through collective reasoning and sharing of discovered core pieces of information.
ATO	Undersea Littoral Warfare: Netted Search, Acquisition and Targeting (NetSAT)	Real-time target localization and tracking for torpedo guidance in presence of countermeasures via a distributed network of sensors passing target location and command guidance. RCT would allow for smarter-munitions that could not only steer/pilot/maneuver itself, but provide and use feedback to aid other munitions (e.g. provide real-time assessment and reconnaissance for latter attacks).
ATO	Wolfpack	
ISO	Advanced Logistics Project (ALP)	Coordinated clusters focusing on logistical efforts
ISO	Control of Agent-Based Systems (CoABS)	This project not only could provides control methods for ABS, but the architectures developed for aiding in the implementation of C2 robotic coordination architectures
ISO	DARPA Agent Mark Up Language (DAML)	
ISO	Joint Force Air Component Commander (JFACC)	Agile and stable control of distributing military applications in uncertain and rapidly changing military environments.
ISO	Ultra*Log	enhance security, robustness, and scalability of large-scale, distributed agent-based logistics systems.
SPO	Advanced Tactical Targeting Technology (AT3)	RCT gives flexible, robust, passive geolocation targeting technology for tactical platforms
SPO	Affordable Moving Surface Target Engagement (AMSTE)	With RCT multiple units will target multiple threats from long range and will precisely engage these threats via stand-off weapons.
SPO	Airborne Video Surveillance (AVS)	UAVs for obtaining large volumes of surveillance information. RTC methods allow AVS technology to be integrated into a full force tactical

¹ DARPA Offices/Directorates Surveyed:

- Advanced Technology Office (ATO) – RD&D for superior cost-effective systems in maritime, communications, special operations, and command and control to support military operations throughout the spectrum of conflict.
- Information Systems Office (ISO) – RD&D into information systems technology yielding flexible methods that give more pertinent information faster.
- Information Technology Office focuses (ITO) – Studies networking, computing, and software technologies vital for long-term military superiority
- Microsystems Technology Office (MTO) – Heterogeneous microchip-scale integration of electronics, photonics, and microelectromechanical systems (MEMS). MTO technologies will provide the necessary fabrication methods and products (robots) for very large-scale coordinated swarm applications.
- Special Projects Office (SPO) – Development of systems solutions, along with the required enabling technologies, to counter current and emerging national threats
- Tactical Technology Office (TTO) – engages in high-risk, high-payoff advanced "system" and "subsystem" approaches to military research, emphasizing the to the development of aeronautic, space, and land systems.

Note: survey compiled June 2001 from public DARPA documents.

		response.
SPO	Immune Building Program	RTC can significantly aid in making buildings less vulnerable and attractive grounds for attack
SPO	Real-Time Battle Damage Assessment (RTBDA)	
SPO	Reconfigurable Aperture Program (RECAP)	
ITO: Networking & Distributed Systems	Active Networks	Networks which turn on a time via smart packets which contain their own handling instructions.
ITO: Networking & Distributed Systems	Dynamic Coalitions	
ITO: Networking & Distributed Systems	Quorum	Deliver assured dynamic response to mission-critical applications on a pool of shared resources. RCT would allow for self-vulnerability for the benefit of a common goals
ITO: Networking & Distributed Systems	Ubiquitous Computing	
ITO: Embedded & Autonomous Systems	Autonomous negotiating teams	Highly distributed, decentralized, autonomous teams with negotiation of tasks, roles, and allocate to provide solutions that are good enough, soon enough
ITO: Embedded & Autonomous Systems	The Joint Air Component Commander (JFACC)	1. Mobile autonomous software – software technology to enable the safe, reliable and cooperative operation of autonomous, free-ranging systems. 2. Software for distributed robotics – large scale results from many small case robots 3. Polymorphous computing architecture – multi-mission, in-flight rechargeable embedded IT system 4. Power-aware computing and communication 5. Software enabled control
ITO: Intelligent Software	Rapid Knowledge Formation	
TTO:	Aeronautic Systems	Hummingbird Warrior (Supersonic) and MAL Interceptor (MALI) - RCT provides the necessary early warning and cooperative integration features
TTO:	Embedded Processors & Control Systems	
TTO:	Land Systems	• Combat Hybrid Power System (CHPS) • Counter-artillery Force Protection (CFP) • Future Combat Systems (FCS) • FCS Concept Development • FCS Net Fires • FCS PerceptOR • FCS Tech Development • FCS UGCV: Unmanned Ground Combat Vehicle • Recon., Surveillance & Targeting Vehicle Technology (RST-V)
TTO	Space Systems	