14052

Paper Number

# Collaborative Robotics Design Considerations

**Parag Batavia**
Applied Perception

**Robert Watts**
TACOM

## ABSTRACT

As research advances individual robot capabilities, a logical progression is the use of multiple robots to complete a task more effectively. Mission performance can be improved by the ability to allocate robots with diverse capabilities to perform different parts of a complex task. To paraphrase [[10], there are many advantages to enabling robotic collaborative technologies: probability of task completion, completion speed, task precision, optimized functional allocation, collaborative localization, robustness and diversity of solution. In previous work, the authors conducted independent literature surveys of multi-robot research, using two different approaches. Both found the underlying research to be categorized within some common sub-disciplines. This paper consolidates findings from the two surveys, presents insights gained from the two different survey methods, and recommends approaches for future research into multi-robot systems.

## INTRODUCTION

Collaborative robotics is becoming a highly relevant area of research and development in the field of mobile robotics. As individual robot capabilities increase, it is only natural to think of how to use multiple robots to complete a task more effectively. To paraphrase [10], there are many advantages to multi-robot systems. These include task completion, completion speed, ability to make optimal use of robots with special capabilities, ability to collectively improve localization estimates, overall system robustness, improvements in task execution precision, and the ability to work in a solution space with greater diversity than that offered by a single robot.

While research in this field has spanned nearly 20 years (going back to work on autonomous software agents), very little of the existing work and research has been demonstrated on large teams of real robots in real-world situations, beyond prototypes and demonstrations.

Based partly on these motivations, there has been a great deal of work in the past 15 to 20 years on collaborative technologies. While the early work was very theoretical and "AI" oriented, more recent work, as we will explore in this paper, generally takes a practical approach. This means that the concepts are often tested on real platforms, in outdoor conditions. However, as we will see, despite all the research being conducted into this topic, there are still open areas, particularly relating to what is necessary to build *deployable* teams of unmanned ground and air vehicles.

This paper presents the results of two different analyses of the state of the art in collaborative robotics. The first survey was conducted using automatic data mining techniques to extract trends and areas of focus in US and Japanese multi-robot research efforts. This automated analysis was able to consider a very large number of publication abstracts, but does not consider the content or quality of the publications. The second analysis was conducted in a more traditional manner, by manually reviewing a large number of recent publications in multi-robot research. This effort, while able to consider the quality and semantics of the reviewed work, was conducted using a smaller sample than the automated survey.

The paper is outlined as follows. First, we present an overview of both surveys. Following that, we present additional detail on different topics in multi-robot research. The discussion on each topic area also includes recommendations for future research. Finally, we present concluding thoughts and insights gained from comparing the results of both surveys.

## DATA MINING SURVEY

Research forecasts, technology assessments and state-of-the-art overviews of a given research area often categorize the sub-disciplines or enabling technologies of the larger field to more easily discuss progress and accomplishments. For example, the research overview, "Guest Editorial, Advances in Multirobot Systems [2]," breaks down multi-robot research into seven topic areas, referred herein as IEEE multi-robot research groups:

1. biological inspirations
2. communication
3. architectures, task allocation and control
4. localization, mapping and exploration
5. object transport and manipulation
6. motion coordination; and
7. reconfigurable robots.

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*

| | | |
|---|---|---|
| 1. REPORT DATE<br>**06 MAY 2004** | 2. REPORT TYPE<br>**Journal Article** | 3. DATES COVERED<br>**06-05-2004 to 06-05-2004** |

4. TITLE AND SUBTITLE
**Collaborative Robotics Design Considerations**

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
**Robert Watts; Parag Batavia**

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
**U.S. Army TARDEC ,6501 E.11 Mile Rd,Warren,MI,48397-5000**

8. PERFORMING ORGANIZATION REPORT NUMBER
**#14082**

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
**U.S. Army TARDEC, 6501 E.11 Mile Rd, Warren, MI, 48397-5000**

10. SPONSOR/MONITOR'S ACRONYM(S)
**TARDEC**

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
**#14082**

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES
**Submitted to NDIA Intelligent Vehicles Conference 2004**

14. ABSTRACT

**As research advances individual robot capabilities, a logical progression is the use of multiple robots to complete a task more effectively. Mission performance can be improved by the ability to allocate robots with diverse capabilities to perform different parts of a complex task. To paraphrase [[10], there are many advantages to enabling robotic collaborative technologies: probability of task completion, completion speed, task precision, optimized functional allocation collaborative localization, robustness and diversity of solution. In previous work, the authors conducted independent literature surveys of multi-robot research using two different approaches. Both found the under1ying research to be categorized within some common sub-disciplines. This paper consolidates findings from the two surveys, presents insights gained from the two different survey methods, and recommends approaches for future research into multirobot syStems.**

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | |
| **unclassified** | **unclassified** | **unclassified** | **Public Release** | **13** | |

This IEEE editorial, multi-robot review, directly references 71 technical papers, and the authors are recognized experts in the field. So, these seven research categories can be considered well founded.

Watts et al. (2004) used text-mining software tools to profile multi-robot research. This bibliometric analysis combined two literature search sets -- 198 abstracts from the El Compendex database and 254 abstracts from the INSPEC database. The combined file contained 354 unique abstracts, published during the 1998-2003 period, related to collaborative robotics, also defined as "multi-robots." The IEEE multi-robot research groups, above, were manually created within the 354 abstracts file using terms/phrases taken from the text of the topic area discussion in the IEEE multi-robot overview paper. Thereby, 310 of the 354 multi-robot abstracts were categorized into one or more of the seven groups. To assess whether any additional research topic areas existed in the analyzed file, a new dataset from the abstracts that were not contained in any of the seven groups was created. Analysis of the abstract noun phrases and descriptors of the ungrouped abstracts revealed two possible additional topic areas -- *robot learning* and *human-robot interface*. Because of specific research interests, the two new topic areas were included in the multi-robot research analysis. The nine resulting topic areas captured 324 of the 354 multi-robot abstracts. The 30 non-grouped abstracts were put into a tenth group entitled "IEEE OTHER."

The Survey [3] reviewed research from 94 technical papers and categorized the underlying research into four similar sub-disciplines: Architectures and Communication, Task Allocation, Localization and Mapping and Human / Robot Interaction. This survey grouped architecture with communication rather than with task allocation, as done under the IEEE survey. Again, because of focused research interests, an eleventh group was created using search terms, *task, auction* and *goal*, to capture the related *task allocation* sub-discipline research activity within the 354 abstracts file.

Table 1, shows the Japanese and USA affiliations' abstracts categorized among the 11 topical groupings and total membership numbers for each research category. Such information categorizations can reveal organizational research emphasis areas. Note that more Japanese research fell into the *motion coordination* category, 52 abstracts, than for USA sources, 37 abstracts; perhaps implying a national research focus on multi-robot motion coordination. Knowledge of research emphasis areas can assist managers in both program planning and collaboration decisions; recognizing that a high-level indicator of a willingness to collaborate is the extent of disclosure of on-going research (i.e., publishing technical reports). Note for the USA sources, the leading source of *motion coordination* categorized research is JPL. For multi-robot motion coordination technologies, one might want to review the research findings of an identified field leader, JPL.

Software tools, as used in the Watts et al. multi-robot survey, can also perform relational analysis of the grouped literature and depict the relationships graphically [27, 28]. Figures 1 and 2 depict the combined multi-robot groups' relatedness maps for Japanese and USA sources, respectively, using descriptors from each countries abstracts to determine and display group relatedness. The size of each node relates to the number of abstracts in the depicted grouping; *Architecture Allocation Control* is the largest node for USA sources multi-robot research (Figure 1) and *Motion Coordination* the largest for Japanese sourced research (Figure 2). The links between nodes depict the relatedness of the research categories (i.e., based on common usage of abstract descriptors). In the upper left corner of each map, Figures 1 and 2, is a summary box showing the number of relatedness links for each of 4 relatedness levels. In comparing Figures 1 and 2, one observes that Japanese research in *Transport Manipulation Grasping* has more emphasis and greater interrelatedness to the core multi-robot research categories, *Architecture Allocation Control, Motion Coordination* and *Task-auction-goal* than USA sourced research. Interestingly, USA sourced *Human Interface* research relates most to the *Task-auction-goal* and *Robot Learning* research categories. Table 1 and Figures 1 and 2 have been included for the reader to reference during the remainder of the paper to note sources and relationships of the multi-robot research sub-disciplines. However, the remaining discussion will focus on five categories: architecture, communication, task allocation, localization and mapping and human / robot interaction, and the perceived design recommendations drawn from the above discussed literature reviews.

## ARCHITECTURES AND COMMUNICATION

While somewhat amorphous, "architecture" defines the methodology by which the robot control problem is decomposed in terms of control, planning, and information flow. The boundaries between what constitutes different architectures are often fuzzy. In general, there are only significant differences between performance and capabilities at the extremes – i.e., there are true differences in capabilities between purely deliberative and purely reactive systems. However, these tend to blur when discussing architectures that are semantically more similar.

While one approach is to structure the solution of a problem to fit the mandated architecture, we believe the applied architecture should fit the desired solution, and not the other way around. The best architectures provide flexibility in design and implementation, and are really more of a framework than a proscription regarding how mobile robotics is to be solved. This is especially important when dealing with multi-robot systems, as this field is immature (even when compared to single-robot

Table 1 – Multi-Robot Research (1998-2003) Affiliations Abstracts in IEEE Categories

|  | # Records | 49 | 46 | 37 | 30 | 24 | 16 | 11 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # Records | **USA Affiliations** | Architecture Allocation Con | Communication | Motion Coordination | Task-Auction-goal | Localization Mapping Explo | Biological | Human Interface | Transport Manipulation Gra | IEEE OTHER | Robot Learning | Reconfigurable |
| 18 | The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, United States | 10 | 9 | 1 | 6 | 5 | 1 |  | 1 | 2 |  |  |
| 10 | Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA | 4 | 4 | 10 | 4 | 4 | 1 | 2 | 3 |  |  | 2 |
| 6 | University of Minnesota, Minneapolis, MN, United States | 3 | 3 |  | 2 | 1 |  |  |  | 1 |  |  |
| 5 | Oak Ridge National Laboratory, Oak Ridge, TN, United States | 1 | 1 | 3 | 2 | 1 |  |  |  |  | 5 |  |
| 5 | Dartmouth College, Hanover, NH, United States | 1 |  |  |  |  |  |  |  |  |  | 5 |
| 5 | University of Southern California, Los Angeles, CA, USA | 1 | 3 | 2 |  | 1 | 2 | 1 |  |  |  |  |
| 4 | University of Southern California, Marina del Rey, CA, USA | 3 | 2 | 3 | 2 | 1 | 2 |  |  |  |  |  |
| 4 | Sandia National Laboratories, Albuquerque, NM, United States | 2 | 3 | 3 | 1 | 3 | 1 |  |  |  |  |  |
| 3 | Clemson University Clemson, SC, USA |  |  | 3 | 1 |  |  | 3 |  |  | 2 |  |
| 3 | University of Notre Dame, Notre Dame, IN, United States | 2 |  |  |  |  |  |  |  | 1 |  |  |
| 3 | Northwestern University, Evanston, IL, United States | 1 |  |  |  |  |  |  | 2 |  |  | 1 |
| 3 | US Army Research Lab., Adelphi, MD, USA | 1 | 3 |  |  |  | 2 | 2 |  |  |  |  |
| 2 | US Naval Acad., Annapolis, MD, USA |  | 2 | 2 |  |  |  |  |  |  |  |  |
| 2 | Michigan Univ., Ann Arbor, MI, USA | 2 |  |  |  |  |  |  | 2 |  |  |  |
| 2 | Lockheed Martin Federal Systems, Owego, NY, USA | 2 | 1 |  | 2 |  |  |  |  |  |  |  |
| 2 | Oakland University, Rochester, MI, USA | 2 |  |  |  |  |  | 1 |  |  |  |  |
| 2 | Ocean Power Technol. Inc., Pennington, NJ, USA |  |  |  |  |  |  |  |  | 2 |  |  |
| 2 | Michigan State University, East Lansing, MI, USA |  | 1 | 1 | 2 |  |  | 1 |  |  | 1 |  |

|  | # Records | 52 | 38 | 38 | 35 | 25 | 19 | 17 | 16 | 16 | 12 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # Records | **Japanese Affiliations** | Motion Coordination | Task-auction-goal | Architecture Allocation Con | Communication | Human Interface | Robot Learning | Transport Manipulation Gra | Localization Mapping Explo | Biological | IEEE OTHER | Reconfigurable |
| 15 | Osaka University, Osaka, Japan | 3 | 3 | 2 | 1 |  | 6 |  | 4 | 4 | 3 |  |
| 11 | Tokyo Inst. of Technology, Yokohama, Jpn | 6 |  | 2 | 5 | 2 |  |  |  |  |  |  |
| 9 | Tohoku University, Sendai, Japan | 2 | 4 | 3 | 1 | 2 |  | 4 |  | 4 | 1 |  |
| 8 | University of Tokyo, Japan | 3 | 2 | 4 | 3 |  | 1 | 1 | 2 | 2 |  | 1 |
| 5 | Doshisha University, Kyo-Tanabe-shi, Kyoto, Japan | 5 | 5 |  |  | 5 |  | 2 | 1 |  |  |  |
| 5 | Saitama University, Urawa, Japan |  |  |  |  | 4 |  |  |  |  | 1 |  |
| 4 | Keio University, Yokohama, Japan | 3 | 3 | 3 |  |  |  | 1 |  |  | 1 |  |
| 4 | Mie University, Mie, Jpn | 4 | 4 | 1 | 1 | 2 |  |  |  |  |  |  |
| 4 | Hokkaido University, Sapporo, Japan | 3 |  | 1 | 2 | 1 | 2 |  |  | 1 |  |  |
| 3 | Nagoya Univ., Chikusa, Nagoya, Jpn |  |  | 1 | 1 |  |  |  | 1 | 1 | 1 |  |
| 3 | Electrotechnical Lab., Ibaraki, Japan |  |  | 2 | 1 |  |  |  | 1 | 1 | 1 |  |
| 3 | Kobe University Kobe, Jpn |  | 3 | 2 | 1 |  | 3 | 3 |  |  |  |  |
| 2 | RIKEN, Inst. of Phys. & Chem. Res., Wako, Japan | 1 | 2 | 1 | 1 |  |  |  | 1 |  |  |  |
| 2 | Iwate Prefectural University Iwate, Jpn | 2 | 1 | 1 | 2 |  |  |  | 2 |  |  |  |
| 2 | Kyushu Univ., Kasuga-shi, Fukuoka, Japan |  |  | 1 |  |  |  |  |  |  | 1 |  |
| 2 | Kobe University, Hyogo, Japan |  | 1 |  |  |  | 2 | 1 |  |  |  |  |
| 2 | Oita University, Japan | 1 |  | 1 |  |  |  |  | 1 | 1 |  |  |
| 2 | Nara Inst of Science and Technology, Nara, Jpn | 2 | 2 |  | 2 | 2 |  |  |  |  |  |  |
| 2 | PRESTO, Japan Sci. & Technol. Corp., Tokyo, Japan | 1 | 1 |  |  |  |  |  |  |  | 1 |  |
| 2 | Toin University, Yokohama, Japan |  |  |  |  | 1 |  | 2 |  |  |  |  |
| 2 | Saga University, Saga-shi, Saga, Japan | 2 |  | 1 |  |  |  | 1 |  | 1 |  |  |

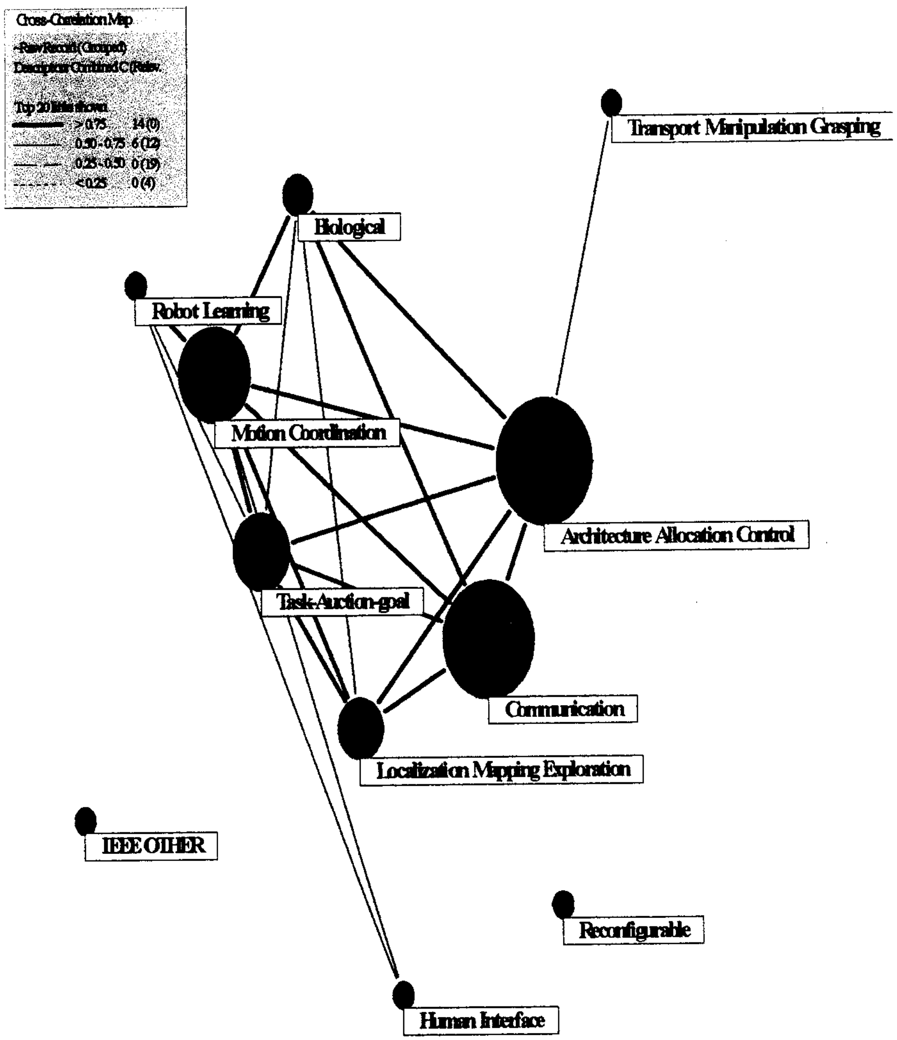

Figure 1 - Multi-Robot Research, 1998-2003, USA Sources, IEEE Categories Map

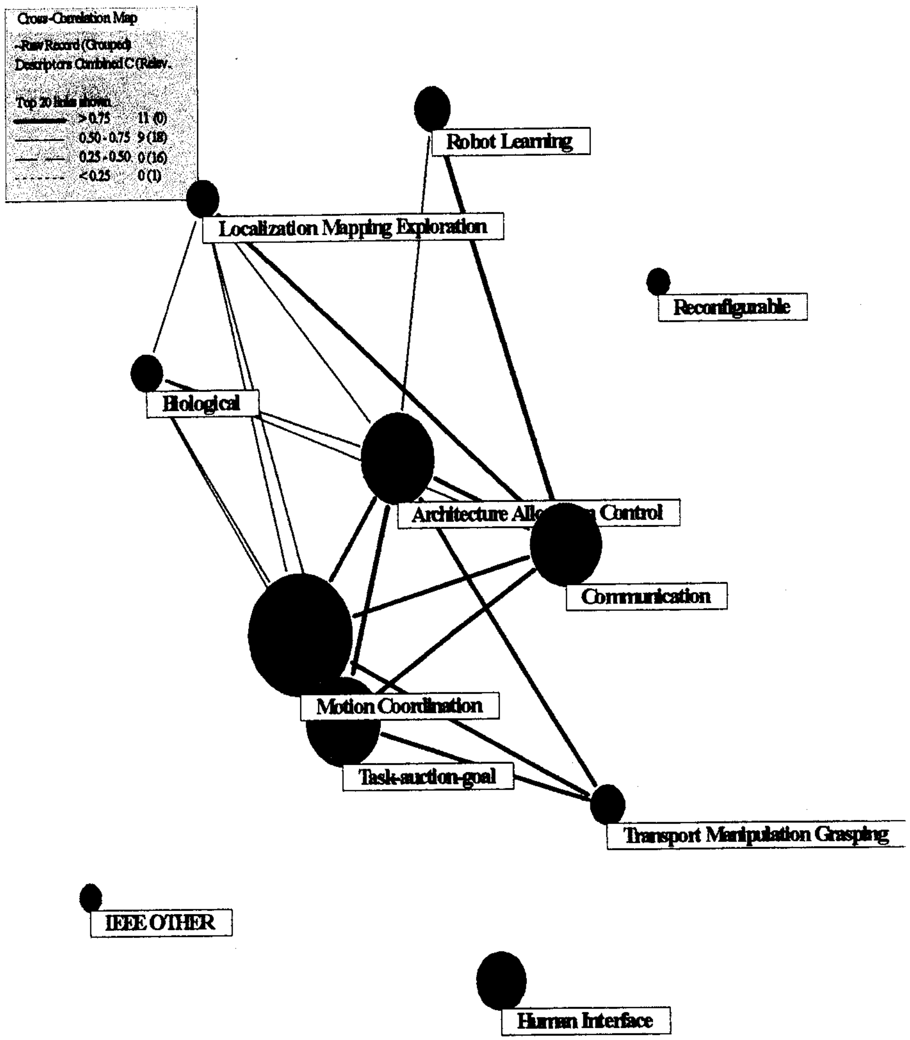

Figure 2 - Multi-Robot Research, 1998-2003, Japanese Sources, IEEE Categories Map

systems). Settling on an architecture at too early a phase can result in designing into a corner, resulting in added cost and development time. In the rest of this section, we present an overview of 8 different architecture types: centralized, hierarchical, reactive, subsumptive, behavior-based, swarm, hybrid and 4D/RCS.

**Centralized** architectures, perhaps the easiest to understand and implement, use a single computer or robot for coordination and collaboration between all team members. In the strictest case, all team members communicate with other team members solely via the master controller.

While purely centralized approaches rely on the central server to do nearly all sensor and cognitive processing, **hierarchical** approaches take a more deliberative view, allowing certain types of processing to occur at different "levels". One constant in pure hierarchical architectures is that action at all levels is deliberative – even at the lowest, servo level. As such, examples of pure hierarchical systems, especially applied to collaborative robotics, are rare, as it is often necessary to include some low-level reactive capabilities.

**Reactive** architectures treat each robot as an individual agent which has certain reactive behaviors to stimuli. The simplest reactive systems exhibit behaviors such as goal seeking or stimuli avoidance. At this level, developing complex coordinated behavior is challenging. As such, systems which require complicated behaviors use one of the other architectures which are described here.

**Subsumption** architecture, another form of reactive architecture first proposed by Brooks [6], emphasizes the lack of representational knowledge. Typically, subsumption architecture uses a set of Augmented Finite State Machines (AFSM), with timers that enable states to change after preprogrammed time periods. When applied to cooperative robotics, reactive architectures strive to enable emergent behaviors. Complex behaviors are supposed to emerge from the relatively simple reactive behaviors of individual robots. In the subsumption architecture, the concept of hormonal activation was introduced [7] as a way to mediate the behaviors of nearby robots, enabling a biologically inspired level of cooperation or competition.

**Behavior-based** architectures, popularized by Mataric [16], consider robot actions as a set of distributed behaviors, each activated or suppressed based on internal and external stimuli. Often confused with Subsumption architecture or purely reactive systems, Mataric lists key differences of behavior-based architectures:

1. Unlike reactive systems, behavior-based systems are not limited to pure lookup (i.e., lookup the response to a particular stimuli).
2. Behavior-based systems can use complex internal representations to model the state of the world.
3. Unlike subsumption-based systems, behavior-based systems are typically distributed, with no centralized arbiter to decide on behavior activation.

**Swarm** architectures involve large numbers of very low cost, low capability homogenous robots – from 10s to 1000s. They are researched in application areas like exploration, self-constructing sensor networks, de-mining, and hazardous area exploration. The main philosophy behind swarm approaches is that each individual robot is expendable, and that if enough robots are applied to the task (given some basic behaviors and rules), the task can be accomplished. Swarm concepts draw heavily from the reactive and behavior-based literature.

**Hybrid** architectures combine deliberative and reactive control strategies, taking advantage of the fact that reactive systems tend to provide the most robust low-level behavior, whereas deliberative systems provide the ability to make long term plans that make use of global knowledge stores and models of the world. This is currently a popular architecture, based on the survey of recent publications. Perhaps the most common example of hybrid architecture is the Three-Level-Architecture (3T) [5]. In this approach, robot and multi-robot control is decomposed into three main levels. The first two are typically deliberative, and consist of the planner and executive level. The lowest level is typically reactive, and maintains low-level vehicle behavior and safety. For multi-robot systems, the typical organization is for both the planner and executive to exist in one location (similar to centralized approaches), but for the behaviors and low level control to be resident on the robot. Shirkhodaie [22] has another approach, with an architecture with six levels. Each level operates using subsumption. This architecture is not classified as subsumption, though. Each layer is recognized in accordance with its specific functionality and task planning hierarchy abstraction. The architecture supports both deliberative and reactive task planning.

Another example of a hybrid approach is the NIST **4D/RCS** architecture [1]. While a hybrid architecture, 4D/RCS is highly slanted towards a top-down deliberative approach, even though reactive control is possible at all levels. The functionality, temporal scales, and spatial extent of the various levels in 4D/RCS are rigidly specified, and follow a pseudo-military hierarchy structure. 4D/RCS levels span timescales from milliseconds to hours. Level 1, the Servo level, is the lowest level and is concerned with safety and actuator loops. It typically operates at the 50ms timescale. This is the level that is most likely to benefit from a reactive approach. At the other extreme, level 8 deals in long-range mission planning, at the Battalion HQ level. The most significant implementation of 4D/RCS was the Demo III UGV program.

At every level, a plan is formulated, and actions to execute the plan are taken. A reactive capability can be specified at each level to allow that level to react quickly to changing plan conditions. Every level has access to a world model database, which contains the state and location of world objects, terrain, and the status of the vehicle and its progress in the various plans each level is executing. Regarding the suitability of 4D/RCS for multi-robot systems, the latest incarnation explicitly includes upper level coordination capabilities (levels 5-8) to address cooperation and collaboration at various levels. However, the actual methods for task allocation, coordination, and conflict resolution are not yet defined.

### DESIGN RECOMMENDATION

Architecture selection in multi-robot systems is a topic which has been studied a great deal, and there are significant opinions on all sides of the issue. Many of these architectures share concepts and have similar performance on various domains. Hybrid architectures tend to have a good balance between the deliberative and reactive, and offer wide latitude to the system developer in how to partition the task assignment and planning problems. They are the least restrictive of the architectures, while still allowing deliberative planning, and as such, offer the most potential flexibility. The applied architecture should fit the solution, and not the other way around. However, DoD policy requires that all new and upgraded command, control, communications, computer and intelligence (C4I) systems, which includes robotic technologies, must comply with the Joint Technical Architecture (JTA), http://www.disa.mil/main/jta.html. Thus, collaborative robots must use open-systems architectures as specified in the JTA.

Multi-robot teams should be able to recognize and adapt to the complete or partial failure of companion robots. Many architectures and task allocation methods surveyed make claims to being fault tolerant; typically, this is limited to some form of recognition of robot failure to complete its task, and then assigning another robot to the task. More efficient failure recovery mechanisms can be achieved with multi-robot systems. Robots do not always fail in ways that render them completely inoperable and useless. Local failures can be contained and corrective actions taken.

Consider a team of robots, each of which having sensors for targeting enemy vehicles, and weapons to disable them. If the robots engage in a coordinated attack on a set of targets, and one robot has a sensor failure, the architectures surveyed would likely remove that robot from action, leaving only one robot to handle the entire task. If, however, more comprehensive fault recognition and handling capabilities were available, the robot with the damaged sensor could make use of targeting and localization information from the remaining robots to aim its weapon. This type of fault recognition and handling capability would greatly increase the robustness of collaborative robotic teams.

Similarly, the ability to recognize *impending* failures of a team member could be used to smoothly adapt to the situation, which is vital in tightly constrained military operations. Multi-robot systems should apply existing embedded diagnostics and prognostics methods to detect impending failures. Detection of an imminent failure could be communicated to the task allocation module (either centralized or decentralized) before the failure occurs, allowing tasks to be assigned based on the reduced performance capabilities. Another, more ambitious method is to investigate methods of *external* diagnostics. In this concept, nearby robots communicate state and performance information with each other, while observing their neighbors' actions. If the communicated state and performance does not match the observed actions, failure can be inferred, and the failing robot can be queried for diagnostic information and/or taken offline. This is analogous to human team members noticing fatigue or poor performance in another team member.

## BANDWIDTH-OPTIMAL COMMUNICATIONS

Multi-robot teams must share and fuse data to optimize combined performance. The method for data sharing typically depends on the implemented architecture. Centralized architectures require communications directly to a centralized server. Communication in distributed architectures is typically between individual robot team members. While the topology of communication in multi-robot systems is well studied, there are areas regarding bandwidth utilization and data representation that are not These areas are explored below

### DESIGN RECOMMENDATIONS

Methods to automatically allow for data transmission, based on the receiver's needs and available bandwidth, should be developed. For example, neighboring robots, sharing sensory information, may require high-bandwidth localized communications. In other applications, a robot may only need a higher-level representation of another robot's sensor data, such as the location of a target, or a low-resolution terrain elevation map. Contextual analysis is required to optimally manage communications. This involves inferences regarding the content of the data, the sources of that data, and the available bandwidth, and, then, abstracting the data to a higher level if bandwidth constraints require it. For instance, consider that Robot A, equipped with a laser scanner, shares scan data with Robot B. If the available bandwidth is reduced (due to signal interference or other demands on available bandwidth), then Robot A can still send semantic information such as the presence of obstacles or other geometric features, derived from the laser data, to Robot B. This would allow Robot B to make some use of the sensor, while still conserving bandwidth. We think this is a very promising research, as bandwidth considerations play a key role in real world robotic deployments, and are often ignored in the limited robotic demonstrations and prototype implementations.

Another equally important extension to the current work is defining methods to share information about mission status, goal completion, and resource availability. These items are critical in effectively distributing and redistributing tasks to enable mission completion for a heterogeneous set of robots operating in the real world. For instance, having a common method of representing and sharing mission plans, goals and sub goals, and robot capabilities, along with completion or functionality metrics, would enable robust, fault tolerant mission execution using an optimal set of robot resources.

## TASK ALLOCATION

Task allocation, a key component of multi-robot systems, addresses the issue of how to assign tasks to different team members. Task assignment algorithms are basically optimization systems, which try to assign tasks to the robots best capable of executing them, leading to a global improvement in team efficiency. There are three primary methods for task allocation [9]:

1. **Centralized** approaches [23] do all task allocation and plan generation in one location, and send low level commands to the team members.
2. **Centralized Goal Allocation** [12] decomposes the task into sub-goals in a centralized fashion, but does local planning at the robot-level.
3. **Auctioning** allows each robot to bid for possible goals, in a purely distributed fashion. Goals can be auctioned in a centralized or decentralized fashion.

Task allocation represents a major component of multi-robot architectures, in terms of defining functionality and performance. Therefore, it would be reasonable to discuss architectures and task allocation in one combined section, and treat task allocation as a component of architectures. However, this concept is important enough to be broken out into its own section. A few general points may help to clarify the relationship between architectures and task allocation systems:

1. Centralized architectures by definition incorporate centralized task allocation schemes.
2. Hierarchical and hybrid architectures can operate with all three methods: 1) centralized task allocation method, by doing task allocation and planning at the highest level 2) goal allocation algorithms, by pushing actual sub-task planning down to lower levels, and 3) auctioning tasks in a centralized or decentralized fashion.
3. Purely reactive and swarm systems typically only do task allocation as a by-product of their reactive behaviors – i.e., it is treated as an emergent property, rather than an explicit coordination.

A task assignment algorithm has to consider the following:

1. The nature of the global task that has to be accomplished (reconnaissance, targeting, search and destroy, etc).
2. How to decompose the global task into subtasks. For instance, search and destroy can be decomposed into 1) multi-robot search, 2) target tracking, 3) weapons fire, 4) damage assessment.
3. How to allocate the subtasks to individual robots. For instance, the search portion should be allocated to robots with the necessary sensors to find targets, weapons discharge should be limited to robots with the necessary weapons, etc.
4. How to handle faults and re-allocation of tasks if a system failure occurs.
5. How to handle resource conflicts, task dependency, and robot interaction and conflict.

One popular approach, the Contract Net Protocol [24] is a method to allow robots to engage in an auction, bidding on tasks that each can complete. The robot bidding the lowest amount (since bidding is usually done in terms of cost to complete the task) gets assigned the task. Robots compute their individual bids by looking at their own cost for completing the subtask. This can include such considerations as distance, time, stealth, energy consumption, presence or absence of necessary sensors and likelihood of completing the task (based on previous experience with that task).

A cost metric is computed using one or more of the above criteria, and this cost (bid) is sent to the task allocator (auctioneer). The task allocator then picks the robot with the lowest cost, and assigns it the task. The winning robot then has to acknowledge that it has received the task, and then goes on to execute it. CNP is fully distributed – any node (robot) is theoretically capable of auctioning off a task, and any node (robot) is able to bid on a task. This allows for a winning robot to re-bid out the task if another opportunity comes along that it is better suited to handle, although the original CNP algorithm does not allow for non-idle robots to bid on tasks. One often cited [10] disadvantage of CNP-based systems is that they produce plans which are globally sub-optimal, as CNP is essentially a greedy strategy. However, in practice, this is not often a concern, and there are techniques which are being developed [1] to address these optimality issues.

### DESIGN RECOMMENDATIONS

The best current task allocation systems are distributed, fault tolerant, and have the (undemonstrated) potential to work with large teams of robots. These systems tend to be developed using auctioning principles based on the Contract Net Protocol (CNP). Real-world performance of these methods on complicated tasks, involving large teams of heterogeneous robots, has not been verified. There are three main issues with CNP-based systems:

1. They are typically most efficient when dealing with homogeneous robots, as the metrics which are

normally used are independent of the robot capability (i.e., often times it is distance to goal, available power, etc, which is used as the metric).
2. Their decentralized nature typically results in solutions which are non-optimal, in a global sense.
3. There is typically little interaction between mission planning systems and auction-based task allocation modules in most architectures. This means that the mission planner is unable to take advantage of explicit resource knowledge when developing the tasks that need to be auctioned off to robots. For example, if a task cannot be auctioned off because there is no robot capable of executing it, the mission planner should be coupled tightly enough with the task allocation module to recognize this fact, and re-plan the mission.

Having said that, auction based systems are still the most promising technology when dealing with large teams. For smaller teams, however, centralized approaches, tightly integrated with mission planners, can be more effective in finding globally optimal solutions that make the best use of available robot resources, if there is sufficient communications capability. Fault tolerance can still be achieved by allowing more than one robot to contain the mission planner, if desired. One issue that comes up, as mentioned above, is the lack of information that most planners have regarding robot capabilities, and how this ties into mission planning and task allocation. Enabling mission planners to have access to more information on robot capabilities requires it to perform some of the functions that are typically associated with task allocation. The planner has to have some notion of which robots can carry out certain tasks, to generate plans which make the best use of available resources. Further research is required in the actual performance of the various task allocation methods in real-world conditions to determine the level of coupling which provides the best tradeoff between fault tolerance (via decentralization) and optimality (in terms of resource utilization).

## LOCALIZATION AND MAPPING

Good localization is a key requirement of single-robot systems (for most applications), and this carries forward to multi-robot teams as well. One advantage multi-robot teams have is that each robot can use information from other robots to improve its own estimate of its position. In heterogeneous teams, this means that robots with expensive, high-accuracy positioning sensors can share that information with other robots, allowing them to position themselves with greater accuracy. Generally, there is one key requirement for cooperative localization: each robot should be able to compute the *relative* location of other robots with good accuracy. i.e., if Robot A knows its position to high accuracy because it has a 2cm GPS capability, and Robot B can measure it's position relative to Robot A with high accuracy, then Robot B can take advantage of the 2cm GPS on Robot A. Once this is accomplished, collaborative mapping and exploration with heterogeneous robots becomes possible, as robots with lower-accuracy positioning sensors can still contribute to the global map. If *global* localization is not necessary, then cooperative localization can be accomplished using only relative measurements to other robots. In that case, the localization problem is treated as an optimization problem, where the goal is to find the most likely position of each robot given a set of observations to other robots.

### COOPERATIVE LOCALIZATION

Nearly all tasks for which a robot could be used require some level of knowledge of its position in the world. This knowledge can take one of three forms:

1. **Global:** The robot knows its position in some global reference frame, normally provided by GPS.
2. **Local:** The robot knows its position relative to some local reference frame, such as a building or other home base.
3. **Relative:** The robot knows its position only relative to other robots with no notion of localization in a global or local frame.

Roumeliotis [20] has extended this framework and developed a Kalman Filter (KF) implementation which fuses *on-board* observations with *off-board* estimates. On-board estimates are position estimates in the "normal" KF sense – measurements from on-board sensors such as GPS and encoders. Off-board estimates are measurements from localization sensors on other robots. This allows a robot with "weak" position sensors to use information from a robot with strong position sensors. A common frame of reference is provided in two manners: 1) through collective shared measurements to known landmarks, and 2) through measurements of the relative position and orientation of neighboring robots. A traditional KF implementation would treat this as a centralized problem, so all position information would have to be transmitted to a central server, which maintains the full Kalman Filter state (in this case, the KF state would include the position and orientation of each robot). This is inefficient, as it relies on each robot having good communication channels to a centralized server. The novel contribution of this work is reformulation of the KF equations to allow for a *distributed* estimation of the propagation phase. This is important because the propagation cycle typically happens at a high rate, and having to communicate it to a central server would require high bandwidth.

### COOPERATIVE MAPPING AND EXPLORATION

Cooperative Mapping, an extension to collaborative localization, requires teams of robots to use their self-consistent positional information to merge sensor readings and to generate large scale maps. There are two main technical issues in collaborative mapping:

1. **Robot deployment** – how to allocate mapping robots to cover the desired area in some optimal fashion.
2. **Sensor fusion** – how can sensor readings from different robots be effectively merged, given that the robots do not have perfect localization.

Another issue, which is not unique to multi-robot mapping, is handling maps with cycles.

Typically, mapping is done by iteratively updating a robot's pose estimate, and using that to register new sensor data into the map. However, unless good localization is available, the robot will suffer from position drift. Over time, as the robot makes a large loop in the world, the start point and end point of the map will not be aligned, as accumulating error will cause the later portions of the map to be mis-registered.

### DESIGN RECOMMENDATION

A common thread between collaborative localization and mapping systems is that the robots either need some form of common position estimate or they need some method of determining the relative pose of other nearby robots. Probabilistic approaches [25] seem to work the best when applied to larger environments, but they still require solving the localization problem.

Another issue with the research to date is that methods for collaborative localization and mapping have primarily been evaluated over relatively small areas, often indoors. Large scale testing of these methods in outdoor and urban environments is still required – especially for methods which make claims of high robustness *and* real-time operation. It is our belief that even over large areas, algorithms based on expectation maximization or maximum-likelihood will work well – but the computational time required to generate detailed maps over acres of terrain could be prohibitively large.

Most of the existing localization systems rely on GPS to some extent. It is difficult to break away from this dependency, due to the heavy investment (technological and financial) that most organizations have made in GPS. While GPS is a necessary component of most localization systems, it has well known failure modes that preclude its use without other complimentary methods. Vision-based localization, including visual odometry, is one such approach. Visual odometry refers to the process of tracking scene features over time, and estimating vehicle location and orientation based on the perceived motion of these features given intrinsic and extrinsic camera parameters. In addition to the pose estimation task, a natural fallout of this process is the ability to place the tracked features in a map for later use by other robots, or on subsequent traversal of the same area by the same robot.

This capability is particularly important for robots that operate in areas that may not have good GPS coverage (indoors, urban canyons, forests) or places where inertial localization systems have problems, such as in mud or rocky/uneven ground. There is a great deal of existing work in this domain in the single-robot area, where it is known as SLAM (Simultaneous Localization and Mapping). Multi-robot SLAM is not as well-studied an area, as the complexity of the problem grows with the number of robots, due to the interrelatedness of the features that each robot tracks, and the issue of optimally allocating robots to map an area, which is dependent on the feature distribution. This is an area that requires additional research, particularly in investigating methods for optimally fusing SLAM-generated maps from multiple robots, and in dynamically adjusting exploration patterns based on partially generated maps.

## HUMAN / ROBOT INTERACTION

Human / Robot interaction (HRI) for teams of heterogeneous robots is perhaps the least mature of the technologies presented here. While there is a great deal of literature dealing with human / system interaction, much of it is slanted towards optimal information display and managing operator workload. While there are lessons that can be drawn from this work regarding operator capability modeling, models of situational awareness, and user interface design, the real time interactive nature of managing teams of outdoor mobile robots provides unique challenges that have not yet been fully addressed. The goal is to develop communications methods that allow humans to interact with robots in the same manner as they interact with other humans.

### ONE USER, MANY ROBOTS

The majority of work to-date has been in the domain of one user controlling many robots. This is the typical model of a battlefield situation, where an operator has a squadron of robots at his disposal, and the goal is to allow the operator to efficiently control and direct the team.

Scholtz [21] proposes a framework for human/robot interaction that has three levels:

1. **Supervisory** relationships where a single human assigns roles and tasks to multiple robots.
2. **Peer** relationships, where a person interacts with a single robot through some form of communications.
3. **Mechanical** relationships, such as teleoperation or more direct control.

Perzanowski [19] looks at various methods of improving communications between humans and robot teams. His approach is directed more towards systems and tasks that require tight interaction, and is targeted towards improving communication efficiency, rather than optimal presentation of data or situational awareness. He proposes three methods for improving communications:

1. **Linguistic cues**, such as prosody (voice inflection) to communicate **intent**. Prosody is used by humans to determine whether a question is being asked, whether a statement is being made, etc. Advances in this area require speech recognition systems which can recognize and provide inflection information.
2. **Visual cues,** such as gestures, can be used to help direct commands to particular robots, and to issue commands to robots when stealth is necessary. Similarly, various visual cues that humans use, such as nodding head to signal assent can be parsed by robot teams in response to queries.
3. **Knowledge cues,** based on situational and capability awareness can be used to make team communications more efficient. As an example, if a sensor on a robot fails, sharing that information to other team members (and controllers) can eliminate the bandwidth of future requests for information from that sensor.

Fong [13] takes a dialog-based approach to remote teleoperation. His model treats the human as a resource which can be used by the robot to resolve queries. Multiple robots can make queries of the human, and an arbitration scheme is provided to prioritize information requests, to avoid overloading the operator.

## MANY USERS, MANY ROBOTS

A more challenging user interaction problem is how to allow multiple users to manage a team of robots. This is desirable as it can allow larger teams to execute a task, as more operators are available to provide input and balance workload. However, the resource allocation problem (considering the operator as a resource) is significantly harder, as is maintaining situational awareness. There isn't much work in this area yet, and what there is, tends to be conceptual in design.

Morris [17] couches the problem in terms of cognitive modeling. He uses an existing cognitive architecture which models cognition as six different resources: working memory, long-term memory, vision, motor, speech, and audition. This approach treats each operator as being equal in capability, which may not be the case. It also doesn't account for the costs that arise in cognitive context switching, and in understanding situational awareness.

Kortenkamp [14] has developed a different architecture for multi-user / multi-robot interaction. His domain is in space construction and monitoring, and deals with how a team of operators can oversee and assign tasks to a team of mobile monitoring robots. This system is being developed at NASA JSC for potential space station and shuttle use. The core of the architecture is a "crewspecific proxy" agent, which provides a standard conceptual interface to various autonomous agents. This proxy represents the user's interests and concerns. Surrounding the architecture is a set of user interfaces (graphical, voice, etc), and services (command and authorization, task status, notification, and location).

## DESIGN RECOMMENDATION

There is a great deal of work that needs to be done to successfully allow human operators to interact with and control teams of robots. It is our belief that the best interaction models will fit naturally into current concepts of operation. This has the potential benefit of reducing training time, allow operators to work in a manner that they are accustomed to, while maintaining efficiency. One way to accomplish this is to develop OCUs which are capable of being mission-specific. While it is not possible to enumerate each and every mission that a multi-robot team may be required to attempt, there is benefit in selecting a small set of likely scenarios, and then tailoring the interaction models (OCUs, command and control methods, task allocation) for those missions. While developing mission specific OCUs can be ineffective in terms of cost and effort, there needs to be work in developing generic OCUs which can have mission-specific "skins". Ideally, this is done in an automatic fasion, by querying the operator for the mission type, and querying the available resources for each robot. In addition, for Army programs, the Army Weapon System Human Computer Interface Style Guide (http://www.pnl.gov/wshciweb/) provides guidance regarding the appearance of HCI components. It would be useful for mission skins to follow these when appropriate. Similarly, there are warrior symbology guidelines at http://home.hiwaay.net/~georgech/Standards/warrior_symbology.htm.

In addition to being mission specific, HRI models have to take into account the relative capabilities of the robots, along with the cognitive capabilities of the human controllers. It is critical that the human operators not be overburdened by requests for information and guidance from the robots. Previous work, such as [13] and [17], treat the human operator as a resource, of which the robot can take advantage. However, this approach, while valid, ignores the costs involved in context switching, changing focus of attention, and maintaining situational awareness. It is significantly easier for one operator to provide guidance when proper situational information is available, and when the guidance which is required is closely related to previous requests. Conversely, it could potentially confuse the operator to receive a request for guidance from a ground vehicle (for instance, requesting permission to fire a weapon), followed by an air vehicle (requesting confirmation that a detected target should be marked for future action), followed again by a ground vehicle (which is experiencing potentially hazardous fault conditions and is requesting permission to continue beyond its safety margin).

HRI models need to take user behavior and workload into account. This can be done via analysis and prioritizing of the pending requests, along with developing methods to better allow multiple users to provide guidance to a team. This way, it is possible for one user to provide UAV targeting guidance, and another user to provide ground robot weapons fire guidance. This fits into the model, used in USAR [8], where individuals are over-specialized for certain tasks. The benefit of this is increased efficiency in task performance, along with better situational awareness maintenance, as operators are not continually asked to switch context.

For smaller teams, where a single operator is likely to be controlling all robots, methods need to be developed to allow the operator to maintain situational awareness of the state of the entire team. This goes beyond graphical display of state on an OCU, and can include other forms of communication, such as audible (for warnings, or high priority status updates) or haptic.

## MARSUPIAL CONCEPTS FOR DEPLOYMENT

Marsupial concepts in multi-robot teams, initially introduced by Murphy [1], have become popular in Urban Search and Rescue (USAR) domains. The core idea behind marsupial robots is that a large "mothership" is capable of carrying and intelligently deploying smaller robots. The benefit of this approach in USAR applications is that a large robot, which is capable of navigating the external wreckage of a building, can deliver smaller robots to ingress points for voids that need to be searched for survivors.

This approach has applications beyond USAR. One which is currently being investigated is the use of a marsupial team of robots to ferry wounded patients off of a battlefield. In this application, a larger vehicle, which is armored and can traverse rugged terrain, ferries smaller vehicles which are deployed on the battlefield. An attending medic finds and triages wounded and attaches them to a modified stretcher, carried by the smaller deployed robots. These robots ferry the stretcher back to the mothership, which transports the patients to a collection point.

Another application for marsupial concepts is in military operations in urban terrain (MOUT). A key capability for successfully completing missions in urban terrain is knowledge acquisition. Urban terrain offers many hiding places for snipers and mobile weaponry. In current concepts of operation, knowledge is gathered by advanced teams of special operations forces, who must place themselves at great risk to identify and remove threats, before the main forces can enter. A large, heavily protected vehicle which can carry smaller reconnaissance robots (including UAVs) could be used to build up threat maps in advance of incoming forces.
Additional work is required to properly characterize the concepts behind successful marsupial systems, as they have not yet been deployed. This would involve:

1. Study of the concept of operations in which they would be used
2. An evaluation of the necessary capabilities of the carrying robot and the passenger robots
3. An evaluation of the required strength – i.e., the number of carrying robots, and how many robots they each need to carry, based on mission scope.
4. Control methods for marsupial teams. In the ideal case, the human operator only has to issue high level commands to the carrying robot, which has the autonomy and tasking abilities necessary to guide and control the passenger robots. Practically, the human operator will likely need to be involved at all levels, given the current state of autonomy.

## CONCLUSION

In this paper, we have presented an overview of two separate surveys of multi-robot systems, along with recommendations for future research. A summary of these recommendations is:

1. Develop architectures which implicitly integrate advanced fault tolerance and recovery capabilities

2. Develop methods for optimizing communications bandwidth by automatically determining the level of necessary abstraction in communications content

3. Develop methods that enable tighter integration of mission planning and task allocation modules, to enable the mission planner to implicitly take advantage of differing capabilities

4. Develop a generic OCU structure that accepts mission-specific skins and communicates using JTA-approved protocols such as JAUS.

5. Develop marsupial concepts for rapid large-scale deployment of robotic teams.

Aside from these recommendations, there are further insights that have been gained through the two surveys and corresponding recommendations. Namely, that there is a great deal of research and knowledge that is embodied in existing publications. Often, these items are lost or ignored, due to the difficulty in conducting thorough background research in a particular area. However, text analysis tools, such as used in this paper, do offer the opportunity to create domain specific knowledge bases from which to draw and improve the efficiency of transferring developed technologies, rather than re-inventing them. Just as we shaped the research groupings of the multi-robot file, figures 1 and 2, by adding specific topical areas of interest (e.g., task allocation), research interests can be mined from prior-art and leveraged in future programs

## REFERENCES

[1] Albus et. al, 4D/RCS: A Reference Model Architecture for Unmanned Vehicle Systems Version 2.0.

[2] Arai, Tamio, Pagello, Enrico and Parker, Lynne: "Guest Editorial, Advances in Multirobot Systems," IEEE Transactions on Robotics and Automation, Vol. 18, No. 5, October 2002

[3] Batavia, P., "A Survey of Collaborative Robotics Technologies,"

[4] Bay, J., Borrelli, L., Chapman, K., Harrold, T., "User Interface and Display Management Design for Multiple Robot Command and Control," Mobile Robots XV and Telemanipulator and Telepresence Technologies VII, Proceedings of SPIE Vol. 4195 (2001)

[5] Bonasso R.P., et al., "A Proven Three-tiered Architecture for Programming Autonomous Robots," Journal of Experimental and Theoretical Artificial Intelligence, 9(2), 1997

[6] Brooks, R., "Intelligence Without Representation," Artificial Intelligence, 47, pp 139-159, 1987

[7] Brooks, R., "Behavioral Synergy Without Explicit Integration," SIGART Bulletin 2 2(4), pp 130-133, 1991.

[8] Casper, J., and Murphy R.R., "Human-Robot Interactions During the Robot-Assisted Urban Search and Rescue Response and the World Trade Center," IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, 33(3), pp 367-385, June, 2003.

[9] Chien, S., et al, "A Comparison of Coordinated Planning Methods for Cooperating Rovers," Proceedings of the 4th Conference on Autonomous Agents, pp 100-101, Barcelona, Spain, 2000

[10] Dias, B., and Stentz, A., "TraderBots: A Market-Based Approach for Resource, Role, and Task Allocation in Multirobot Coordination," Carnegie Mellon University Technical Report, CMU-RI-TR-03-19, August, 2003.

[11] Dixon, K., Dolan, J., Huang, W., Paredis, C., Khosla, P., "RAVE: A Real and Virtual Environment for Multiple Mobile Robot Systems," IEEE International Conference on Intelligent Robotics and Systems, V3, 1999 pg 1360-1367

[12] Estlin, T., et al., "An Integrated System for Multi-Rover Scientific Exploration," In Proceedings of the AAAI-99, 1999

[13] Fong, T., et al., "Robot as Partner: Vehicle Teleoperation with Collaborative Control," *Multi-Robot Systems: From Swarms to Intelligent Automata*, Proceedings of the 2002 NRL Workshop on Multi-Robot Systems, Schultz, A.C., and Parker, L.E, eds., Kluwer Academic Publishers, 2002, pp 195-202

[14] Kortenkamp, D., et al, "User Interaction with Multi-Robot Systems," *Multi-Robot Systems: From Swarms to Intelligent Automata*, Proceedings of the 2002 NRL Workshop on Multi-Robot Systems, Schultz, A.C., and Parker, L.E, eds., Kluwer Academic Publishers, 2002, pp 213-220.

[15] Mataric, M., "Coordination and Learning in Multirobot Systems," IEEE Intelligent Systems, March/April 1998, pp. 6-8

[16] Mataric, M., "Behavior-Based Control: Main Properties and Implications," Proceedings of the IEEE International Conference on Robotics and Automation, Workshop for Architectures for Intelligent Control Systems, pp. 46-54, Nice, France, 1992

[17] Morris, A.C., et al., "Adaptive Multi-Robot, Multi-Operator Work Systems," *Multi-Robot Systems: From Swarms to Intelligent Automata*, Proceedings of the 2002 NRL Workshop on Multi-Robot Systems, Schultz, A.C., and Parker, L.E, eds., Kluwer Academic Publishers, 2002, pp 203-211

[18] Murphy, R.R., "Marsupial-Like Mobile Robot Societies," Proceedings of the 3rd Annual Conference on Autonomous Agents, Seattle, Washington, 1999.

[19] Perzanowski, D., et al., "Communicating with Teams of Cooperative Robots," *Multi-Robot Systems: From Swarms to Intelligent Automata*, Proceedings of the 2002 NRL Workshop on Multi-Robot Systems, Schultz, A.C., and Parker, L.E, eds., Kluwer Academic Publishers, 2002, pp 185-193.

[20] Roumeliotis, S., Bekey, G.A., "Distributed Multirobot Localization," IEEE Transactions on Robotics and Automation, Vol. 18, No. 5, Oct. 2002, pp 781-795.

[21] Scholtz, J.C., "Human-Robot Interactions: Creating Synergistic Cyber Forces," *Multi-Robot Systems: From Swarms to Intelligent Automata*, Proceedings of the 2002 NRL Workshop on Multi-Robot Systems, Schultz, A.C., and Parker, L.E, eds., Kluwer Academic Publishers, 2002, pp 177-183.

[22] Shirkhodaie, Amir, "Supervised Control of Cooperative Multi-Agent Robotic Vehicles," IEEE Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory, 2002.

[23] Simmons, R., et al., "Coordinated Deployment of Multiple, Heterogeneous Robots," Proceedings of the Conference on Intelligent Robots and Systems (IROS), Takamatsu, Japan, 2000

[24] Smith, R., "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," IEEE Transactions on Computers, Vol. C-29, No. 12, 1980

[25] Thrun, S., Burgard, W., and Fox, D., "A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping," in Proc. IEEE International Conference on Robotics and Automation, 2000 pp. 321-328.

[26] Watts, R., Porter, A., Minsk, B., "Automated Text Mining Comparison of Japanese and USA Multi-robot Research," Data Mining 2004, Malaga, Spain, Sept. 2004 (Submitted).

[27] Watts, R.J., Porter, A.L., Courseault, C.: Functional Analysis: Deriving Systems Knowledge from Bibliographic Information Resources, *Information, Knowledge, Systems Management* 1(1), 45-61 (1999).

[28] Zhu, D., Porter, A.L., Cunningham, S.W., Carlisle, J., and Nayak, A.: "A Process for Mining Science & Technology Documents Databases, Illustrated for the

Case of 'Knowledge Discovery and Data Mining'," Internal TOA Paper #94 [available on request].

[29] Zlot, Robert, "Complex Task Allocation for Multirobot Coordination," CMU Robotics Institute Thesis Proposal.