

Generating Self-Reliant Teams of Autonomous Cooperating Robots: Desired Design Characteristics *

Lynne E. Parker

Center for Engineering Science Advanced Research
Computer Science and Mathematics Division
Oak Ridge National Laboratory
P. O. Box 2008-6355
Oak Ridge, TN 37831

“This submitted manuscript has been authored by a contractor of the U. S. government under Contract No. DE-AC05-98OR22464. Accordingly, the U. S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U. S. Government purposes.”

Presentation to Autonomy Control Software Workshop at Agents '99, Seattle, Washington, May 1, 1999.

* Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, U. S. Department of Energy, under Contract No. DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation.

Generating Self-Reliant Teams of Autonomous Cooperating Robots: Desired Design Characteristics

Lynne E. Parker, Ph.D.
Center for Engineering Science Advanced Research
Computer Science and Mathematics Division
Oak Ridge National Laboratory
P. O. Box 2008, Mailstop 6355
Oak Ridge, TN 37831-6355

1. Introduction

The difficulties in designing a cooperative team are significant. Several of the key questions that must be resolved when designing a cooperative control architecture include:

- How do we formulate, describe, decompose, and allocate problems among a group of intelligent agents?
- How do we enable agents to communicate and interact?
- How do we ensure that agents act coherently in their actions?
- How do we allow agents to recognize and reconcile conflicts?

However, in addition to these key issues, the software architecture must be designed to enable multi-robot teams to be robust, reliable, and flexible. Without these capabilities, the resulting robot team will not be able to successfully deal with the dynamic and uncertain nature of the real world. In this extended abstract, we first describe these desired capabilities. We then briefly describe the ALLIANCE software architecture that we have previously developed for multi-robot cooperation. We then briefly analyze the ALLIANCE architecture in terms of the desired design qualities identified.

2. Desired Design Qualities for Cooperative Robot Team Architectures

2.1 Robustness and Fault Tolerance

Robustness refers to the ability of a system to gracefully degrade in the presence of partial system failure; the related notion of *fault tolerance* refers to the ability of a system to detect and compensate for partial system failures. Requiring robustness and fault tolerance in a cooperative architecture emphasizes the need to build cooperative teams that minimize their vulnerability to individual robot outages --- a requirement that has many implications for the design of the cooperative team.

To achieve this design requirement, one must first ensure that critical control behaviors are distributed across as many robots as possible rather than being centralized in one or a few robots. This complicates the issue of action selection among the robots, but results in a more robust multi-robot team since the failure of one robot does not jeopardize the entire mission.

Second, one must ensure that an individual robot does not rely on orders from a higher-level robot to determine the appropriate actions it should employ. Relying on one, or a few, coordinating robots makes the team much more vulnerable to individual robot failures. Instead, each robot should be able to perform some meaningful task, up to its physical limitations, even when all other robots have failed.

And third, one must ensure that robots have some means for redistributing tasks among themselves when robots fail. This characteristic of task re-allocation is essential for a team to accomplish its mission in a dynamic environment.

2.2 Reliability

Reliability refers to the dependability of a system, and whether it functions properly each time it is utilized. To properly analyze a cooperative robot architecture, one should separate the architecture itself from the robots on which the architecture is implemented. Clearly, if the architecture is implemented on robots that only function a small percentage of the time, one cannot expect a very dependable result from trial to trial. One measure of the reliability of the architecture is its ability to guarantee that the mission will be solved, within certain operating constraints, when applied to any given cooperative robot team. Without this characteristic, the usefulness of a control architecture is clearly much diminished.

As an example of a reliability problem exhibited in a control architecture, consider a situation in which two robots, r_1 and r_2 , have two tasks, t_1 and t_2 , to perform. Let us assume that their control architecture leads them to negotiate a task allocation, which results in r_1 performing task t_1 and r_2 performing task t_2 . Further suppose that r_1 experiences a mechanical failure that neither r_1 nor r_2 can detect. While r_1 continues to attempt to complete task t_1 , robot r_2 successfully completes task t_2 . However, although r_2 also has the ability to successfully complete task t_1 , it does nothing further because it knows that r_1 is performing that task. Thus, the robots continue forever, never completing the mission. One would not term such a control architecture *reliable*, since a mere reallocation of the tasks would have resulted in the mission being successfully completed.

2.3 Flexibility and Adaptivity

The terms *flexibility* and *adaptivity* refer to the ability of team members to modify their actions as the environment or robot team changes. Ideally, the cooperative team should be responsive to changes in individual robot skills and performance as well as dynamic environmental changes. This requirement reflects the fact that the capabilities of robots may change over time due to learning, which should enhance performance, or due to mechanical or environmental causes that may reduce or increase a robot's success at certain tasks. Team members should respond to these changes in performance by taking over tasks that are no longer being adequately performed or by relinquishing those tasks better executed by others. Each robot must decide which task it will undertake based on the actual *performance* of tasks by other robots, rather than on what other robots *say* that they are able to accomplish.

Robots must also exhibit flexibility in their action selection during the mission in response to the dynamic nature of their environment. Obviously, in real environments changes occur that cannot be attributed to the actions of any robot team member or members. Rather, outside forces not under the influence of the robot team affect the state of the environment throughout the mission. These effects may be either destructive or beneficial, leading to an increase or decrease in the workload of the robot team members. The robot team should therefore be flexible in its action selections, opportunistically adapting to environmental changes that eliminate the need for certain tasks, or activating other tasks that a new environmental state requires.

2.4 Coherence

Coherence refers to how well the team performs as a whole, and whether the actions of individual agents combine toward some unifying goal. Typically, coherence is measured along some dimension of evaluation, such as the quality of the solution or the efficiency of the solution. Efficiency considerations are particularly important in teams of heterogeneous robots whose capabilities overlap, since different robots are often able to perform the same task, but with quite different performance characteristics. To obtain a highly efficient team, the control architecture should ensure that robots select tasks such that the overall mission performance is as close to optimal as possible.

A team in which agents pursue incompatible actions, or in which they duplicate each other's actions cannot be considered a highly coherent team. On the other hand, designing a coherent team does not require the elimination of all possible conflict. Rather, the agents must be provided with some mechanism to resolve the conflicts as they arise. A simple example of conflict occurs whenever multiple robots share the same workspace; although they may have the

same high-level goals, they may at times try to occupy the same position in space, thus requiring them to resolve their positioning conflict. This can usually be accomplished through a very simple protocol.

Clearly, multi-robot teams exhibiting low coherence are of limited usefulness in solving practical engineering problems. A design goal in building cooperative robot teams must therefore be to achieve high coherence.

3. The ALLIANCE Architecture

We now briefly describe the ALLIANCE architecture that was designed with the above design requirements in mind. Reasons for the design selections we made will be noted in this section, then summarized and compared to our design requirements later in the paper.

ALLIANCE is a fully distributed architecture for fault tolerant, heterogeneous robot cooperation that utilizes adaptive action selection to achieve cooperative control for small- to medium-sized mobile robot teams. Under this architecture, the robots possess a variety of high-level task-achieving functions that they can perform during a mission, and must at all times select an appropriate action based on the requirements of the mission, the activities of other robots, the current environmental conditions, and their own internal states. The missions that the robot team can address under ALLIANCE are restricted to those missions which are composed of loosely coupled **subtasks** that are independent, but may have fixed ordering dependencies among them. We note, however, that even with this restriction, the proposed architecture covers a very large range of missions for which cooperative robots are useful.

In ALLIANCE, individual robots are designed using a behavior-based approach [1], in order to achieve robustness at the individual robot level. Under the behavior-based construction, a number of task-achieving behaviors are active simultaneously, each receiving sensory input and controlling some aspect of the actuator output. The lower-level behaviors, or competencies, correspond to primitive survival behaviors such as obstacle avoidance, while the **higher-level** behaviors correspond to higher goals such as map building and exploring. The output of the lower-level behaviors can be *suppressed* or *inhibited* by the upper layers when the upper layers deem it necessary. This approach has been used successfully in a number of robotic applications, several of which are described in [2].

Extensions to this approach are necessary, however, when a robot must select among a number of competing actions --- actions that cannot be pursued in parallel. Unlike typical behavior-based approaches, ALLIANCE delineates several **behavior sets** that are either active as a group or hibernating. Figure 1 shows the general architecture of ALLIANCE and illustrates three such behavior sets. The j^{th} behavior set, a_{ij} , of a robot r_i corresponds to those levels of competence required to perform some high-level task-achieving function. When a robot activates a behavior set, we say that it has selected the task corresponding to that behavior set. Since different robots may have different ways of performing the same task, and therefore activate different behavior sets to perform that task, we define the function $h_i(a_{ij})$, for all robots r_i on the team, to refer to the task that robot r_i is working on when it activates its j^{th} behavior set, a_{ij} .

Because of the alternative goals that may be pursued by the robots, the robots must have some means of selecting the appropriate behavior set to activate. Thus, controlling the activation of each of these behavior sets is a **motivational behavior**. Due to conflicting goals, only one behavior set per robot can be active at any point in time. This restriction is implemented via cross-inhibition of motivational behaviors, represented by the arcs at the top of Figure 1, in which the activation of one behavior set suppresses the activation of all other behavior sets. However, other lower-level competencies such as collision avoidance may be continually active regardless of the high-level goal the robot is currently pursuing. Examples of this type of continually active competence are shown in Figure 1 as layer 0, layer 1, and layer 2.

The primary mechanism for achieving adaptive action selection in this architecture is the motivational behavior. At all times during the mission, each motivational behavior receives input from a number of sources, including sensory feedback, inter-robot communication, inhibitory feedback from other active behaviors, and internal motivations called **robot impatience** and **robot acquiescence**. The output of a motivational behavior is the activation level of its corresponding behavior set, represented as a non-negative number. When this activation level exceeds a given threshold, the corresponding behavior set becomes active.

The ALLIANCE Architecture

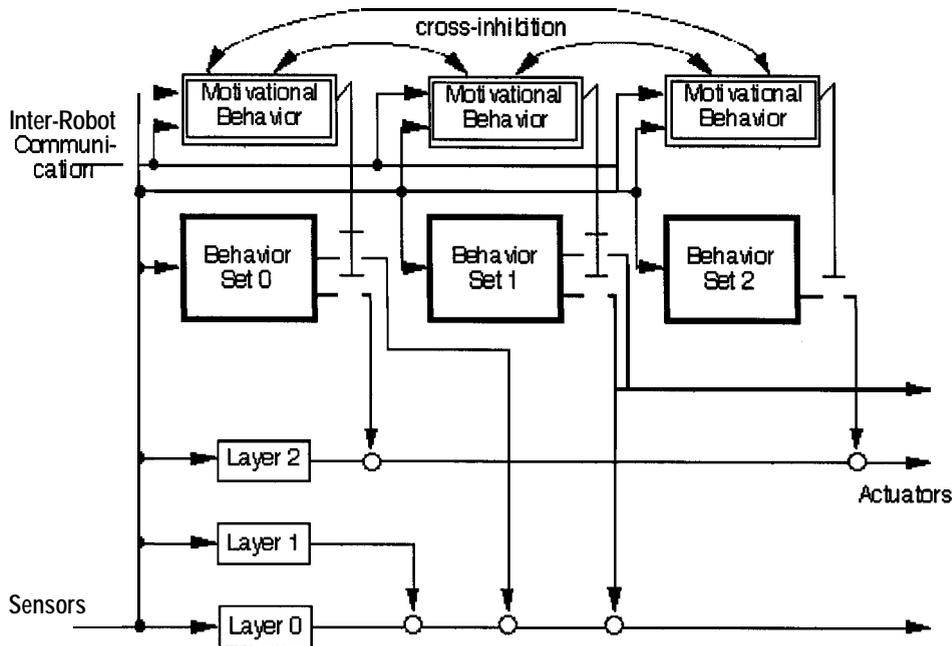


Figure 1. The ALLIANCE architecture.

Intuitively, a motivational behavior works as follows. Robot r_i 's motivation to activate any given behavior set a_{ij} is initialized to 0. Then, over time, robot r_i 's motivation $m_{ij}(t)$ to activate behavior set a_{ij} increases at a "fast" rate (called $\delta_{fast_{ij}}(t)$) as long as the task corresponding to that behavior set (i.e. $h_f(a_{ij})$) is not being accomplished, as determined from sensory feedback. However, the robots must be responsive to the actions of other robots, adapting their task selection to the activities of team members. Thus, if a robot r_i is aware that another robot r_k is working on task $h_f(a_{ij})$, then r_i is satisfied for some period of time that the task is going to be accomplished even without its own participation, and thus go on to some other applicable action. Its motivation to activate behavior set a_{ij} still increases, but at a slower rate (called $\delta_{slow_{ij}}(k,t)$). This characteristic prevents robots from replicating each other's actions and thus wasting needless energy. Of course, detecting and interpreting the actions of other robots (often called **action recognition**) is not a trivial problem, and often requires perceptual abilities that are not yet possible with current sensing technology. As it stands today, the sensory capabilities of even the lower animals far exceed present robotic capabilities. Thus, to enhance the robots' perceptual abilities, ALLIANCE utilizes a simple form of broadcast communication to allow robots to inform other team members of their current activities, rather than relying totally on sensory capabilities. At some pre-specified rate, each robot r_i broadcasts a statement of its current action, which other robots may listen to or ignore as they wish. No two-way conversations are employed in this architecture. Since we have designed the architecture for small- to medium-sized robot teams (i.e. tens of robots rather than hundreds or thousands), the issue of the communication cost for large numbers of robots is avoided. This broadcast form of communication is not designed for applications involving large "swarm"-type robot teams.

Each robot is designed to be somewhat impatient, however, in that a robot r_i is only willing for a certain period of time to allow the communicated messages of another robot to affect its own motivation to activate a given behavior set. Continued sensory feedback indicating that a task is not getting accomplished thus overrides the statements of another robot that it is performing that task. This characteristic allows robots to adapt to failures of other robots, causing them to ignore the activities of a robot that is not successfully completing its task.

A complementary characteristic in these robots is that of acquiescence. Just as the impatience characteristic reflects the fact that other robots may fail, the acquiescence characteristic indicates the recognition that a robot itself may fail.

This feature operates as follows. As a robot r_i performs a task, its willingness to give up that task increases over time as long as the sensory feedback indicates the task is not being accomplished. As soon as some other robot r_k indicates it has begun that same task and r_i feels it (i.e. r_i) has attempted the task for an adequate period of time, the unsuccessful robot r_i gives up its task in an attempt to find an action at which it is more productive. Additionally, even if another robot r_k has not taken over the task, robot r_i may give up its task anyway if it is not completed in an acceptable period of time. This allows r_i the possibility of working on another task that may prove to be more productive rather than becoming stuck performing the unproductive task forever. With this acquiescence characteristic a robot is able to adapt its actions to its own failures.

The behavior-based design of the motivational behaviors also allows the robots to adapt to unexpected environmental changes that alter the sensory feedback. The need for additional tasks can suddenly occur, requiring the robots to perform additional work, or existing environmental conditions can disappear and thus relieve the robots of certain tasks. In either case, the motivations fluidly adapt to these situations, causing robots to respond appropriately to the current environmental circumstances. Refer to [3,4] for more mathematical details of the ALLIANCE architecture, including the formal mathematical model of ALLIANCE, proofs of correction which guarantee that ALLIANCE will allow the robot team to accomplish its mission under certain conditions, and results of physical robot implementations of the ALLIANCE architecture.

4. Meeting Design Requirements

We now review the design requirements outlined earlier and examine the extent to which ALLIANCE meets these design goals. The primary design goal was to develop a cooperative architecture that allowed heterogeneous robots to cooperate to accomplish a mission while exhibiting robustness, reliability, flexibility, and coherence. As these issues are reviewed here, note that the development of a cooperative robot architecture can actually be viewed as the development of an individual robot control architecture that facilitates a single robot's cooperation with other similarly-designed robots. Thus, we describe how each of these performance issues are addressed in ALLIANCE both from the view of an individual robot control strategy and from the view of a collective team strategy.

4.1 Robustness and Fault Tolerance

As described earlier, *fault* tolerance and *robustness* refer to the ability of a system to detect and gracefully compensate for partial system failures. In ALLIANCE, each individual robot is designed using a behavior-based approach, which ensures that lower levels of competence continue to work even when upper levels break down. In addition, individual robots can be given multiple ways to perform certain tasks, allowing them to explore alternative approaches when met with failure.

From the viewpoint of the team, ALLIANCE first enhances robustness by being fully distributed. Unlike hierarchical architectures, since no individual robot in ALLIANCE is responsible for the control of other robots, the failure of any particular robot is not disproportionately damaging. Secondly, ALLIANCE enhances team robustness by providing mechanisms for robot team members to respond to their own failures or to failures of teammates, leading to a reallocation of tasks (such as the leader role in the bounding overwatch application) to ensure that the mission is completed. Third, ALLIANCE allows the robot team to accomplish its mission even when the communication system breaks down. Although the team's performance in terms of time and energy may deteriorate without communication, at least the team is still able to accomplish its mission. Finally, ALLIANCE enhances team robustness by making it easy for robot team members to deal with the presence of overlapping capabilities on the team. The ease with which redundant robots can be incorporated on the team provides the human team designer the ability to utilize physical redundancy to enhance team robustness.

4.2 Reliability

Reliability refers to the dependability of a system and whether it functions properly each time it is utilized. ALLIANCE is designed for applications involving a significant amount of uncertainty in the capabilities of robot team members which themselves operate in dynamic, unpredictable environments. In ALLIANCE, reliability is measured

in terms of the architecture's ability to have the robot team accomplish its mission each time the mission is attempted. It can be shown that under certain conditions ALLIANCE is guaranteed to allow, the robot team to complete its mission, except when robot failures eliminate required capabilities from the team (from which no architecture could recover). While a rigorous presentation of this proof is beyond the scope of this paper (see [3] for more details), we sketch the proof briefly here.

It can be shown that ALLIANCE allows teams of robots that have the following characteristics:

- monotonically increasing motivations to perform the tasks of the mission
- finite thresholds of activation, and
- sufficient task coverage (i.e. for all tasks of the mission, some robot is present that has the ability to perform that task)

to successfully accomplish their mission whenever a condition called *Progress when Working* is true and when no robots with the sole capability to solve certain tasks of the mission fail. *The Progress when Working* condition is true when a robot makes progress toward achieving a task when it activates its behavior sets.

The proof of this claim is based upon the allowable values of the motivations of the robots and the possible interactions of robot task reallocations. The allowable values of the motivations lead to the fact that a robot always has a strictly increasing motivation to perform some incomplete task, which in turn leads to the activation of the corresponding behavior set, and thus the completion of a portion of the task. Evaluating the possible interactions of robot task reallocations in terms of impatience and acquiescence values leads to the conclusion that the tasks of the mission must eventually be completed in finite time. The ALLIANCE action selection mechanism thus gives a means for the robot team to achieve its mission reliably when these conditions hold.

4.3 Flexibility and Adaptivity

Flexibility and *adaptivity* refer to the ability of robots to modify their actions as the environment or robot team changes. The motivational behavior mechanism of ALLIANCE constantly monitors the sensory feedback of the tasks that can be performed by an individual agent, adapting the actions selected by that agent to the current environmental feedback and the actions of its teammates (such as the subgroup selection of robots in the bounding overwatch application). Whether the environment changes to require the robots to perform additional tasks or to eliminate the need for certain tasks, ALLIANCE allows the robots to handle the changes fluidly and flexibly. ALLIANCE enhances the adaptivity and flexibility of a robot *team* by providing mechanisms for robots to work with any other robots designed using ALLIANCE; the robots are not required to possess advance knowledge of the capabilities of the other robots.

One limitation of ALLIANCE is that the parameters of the system (i.e. rates of impatience, acquiescence, and communication) must be provided by the human, thus limiting ALLIANCE's ability to improve its performance over time through adaptivity. An extension to ALLIANCE that we have developed, called L-ALLIANCE [5] (beyond the scope of the current paper) further extends the flexibility and adaptivity of the system by allowing robots to learn about their own abilities and the abilities of their teammates in order to improve their performance on subsequent trials of similar missions whenever familiar agents are present.

4.4 Coherence

Coherence refers to how well the actions of individual agents combine towards some unifying goal. For individual agents, ALLIANCE causes robots to work only on those tasks that the environmental feedback indicates need to be executed. Thus, ALLIANCE will not cause an individual agent to work on some task that is not required by the mission, nor consistent with the current state of the environment.

Obtaining coherence at the team level requires that robots have some means of determining the actions of other robots and/or the effect of those actions on the environment. Without this knowledge, the robots become a collection of individuals pursuing their own goals in an environment that happens to contain other such robots. While we certainly

want the robots to be able to accomplish something useful even without knowledge of other robots on the team, ideally each robot should take into account the actions of other robots in selecting their own actions.

Determining the actions of other robots can be accomplished through either passive observation or via explicit communication. Since passive action recognition is very difficult and is a major research topic in itself, ALLIANCE augments the observation skills of the robot team members through the use of one-way broadcast communication that provides each robot with an awareness of the actions of other robots, plus the ability to act on that information. With this awareness, robots do not replicate the actions of other robots, thus giving them more coherence. We note the importance of this mechanism to achieve team coherence, since when the communications mechanism is unavailable, team coherence is reduced.

Note that a number of issues regarding the efficiency of ALLIANCE are not addressed here. Among these issues include questions of how long robots remain idle before activating a task, how to ensure that robots failing at one task go on to attempt another task they might be able to accomplish, how robots deal with having more than one way to accomplish a task, and so forth. These issues are addressed in the L-ALLIANCE extension to ALLIANCE [5] through the use of dynamic parameter update mechanisms.

5. Conclusions

In this paper, we have addressed the issue of developing robotic technologies that successfully deal with the dynamics and complexity found in real-world applications. We described the design goals of real-world robotics applications, and briefly presented a general architecture --- called ALLIANCE --- that facilitates the fault-tolerant, adaptive control of small- to medium-sized teams of cooperating robots. The key characteristics of this architecture can be summarized as follows:

- Fully distributed (at both the individual robot level and at the team level)
 - . Applicable to robot teams having any degree of heterogeneity
 - . Uses one-way broadcast communication; no negotiation or two-way conversations are utilized
 - . Recovers from failures in individual robots or in the communication system
 - . Allows new robots to be added to the team at any time
 - . Allows adaptive action selection in dynamic environments
 - . Eliminates replication of effort when communication is available
- Provably terminates for a large class of applications

We examined the ALLIANCE architecture in light of the design characteristics identified in the first part of the paper. While these design characteristics are useful from a qualitative perspective, much work is yet to be done to develop specific means of quantifying these characteristics in terms of metrics of cooperation. Ongoing work is examining this issue of quantifying performance metrics for cooperative teams.

Acknowledgements

This research was funded in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-91-J-4038 at the Massachusetts Institute of Technology's Artificial Intelligence Laboratory, and in part by the Office of Engineering Research, Basic Energy of the U.S. Department of Energy, under contract No. DE-AC05-96OR22464 with Lockheed Martin Energy Research, Inc.

References

- [1] Rodney A. Brooks, "A robust layered control system for a mobile robot", *IEEE Journal of Robotics and Automation*, RA-2 (1): 14-23, 1986.
- [2] Rodney A. Brooks, "Elephants don't play chess", *Robotics and Autonomous Systems*, 1990, No. 6: 3-15.

[3] Lynne E. Parker, "ALLIANCE: An architecture for fault tolerant multi-robot cooperation", *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 2, April 1998: 220-240.

[4] Lynne E. Parker, "On the design of behavior-based multi-robot teams", *Advanced Robotics*, Vol. 10, No. 6, 1996: 547-578.

[5] Lynne E. Parker, "L-ALLIANCE: Task-oriented multi-robot learning in behavior-based systems", *Advanced Robotics*, Vol. 11, No. 4, 1997: 305-322.