Appeared in *Journal of Experimental and Theoretical Artificial Intelligence*, 13 (2001): 95-98.

Evaluating Success in Autonomous Multi-Robot Teams: Experiences from ALLIANCE Architecture Implementations

Lynne E. Parker [1]
Center for Engineering Science Advanced Research, Computer Science and Mathematics Division
Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6355

1 Introduction

Quantitatively evaluating the effectiveness of software architectures for multi-robot control is a challenging task. Exacerbating the problem is the fact that architectures are typically constructed to address different design goals and application domains. In the absence of benchmarks that capture the variety of issues that arise in multi-robot coordination and cooperation, the system developer can only evaluate an architecture for its own qualities. In this short article, we summarize the metrics of evaluation that we utilized in applying our ALLIANCE architecture [2] to eight different application domains for multi-robot team control. We explore the implications of the metrics we have chosen and offer suggestions on future productive lines of research into metrics for multi-robot control architectures.

2 Brief overview of ALLIANCE

We developed the ALLIANCE architecture to enable fault tolerant action selection in multi-robot teams. The focus was on an approach that operated successfully amidst a variety of uncertainties, such as sensory and effector noise, robot failures, varying team composition, and a dynamic environment. The ALLIANCE architecture is a behavior-based, distributed control technique. Unlike typical behavior-based approaches, ALLIANCE delineates several behavior sets that are either active as a group or are hibernating. Each behavior set of a robot corresponds to those levels of competence required to perform some high-level task-achieving function. 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. This action selection is controlled through the use of motivational behaviors, each of which controls the activation of one behavior set. Due to conflicting goals, only one behavior set is active at any point in time (implemented via cross-inhibition of 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.

The motivational behavior mechanism is based upon the use of two mathematically-modeled motivations within each robot – impatience and acquiescence – to achieve adaptive action selection. Using the current rates of impatience and acquiescence, as well as sensory feedback and knowledge of other team member activities, a motivational behavior computes a level of activation for its corresponding behavior set. Once the level of activation has crossed the threshold, the corresponding behavior set is activated and the robot has selected an action. The motivations of impatience and acquiescence allow robots to take over tasks from other team members (i.e., become impatient) if those team members do not demonstrate their ability – through their effect on the world – to accomplish those tasks. Similarly, they allow a robot to give up its own current task (i.e., acquiesce) if its sensory feedback indicates that adequate progress is not being made to accomplish that task.

3 Evaluation of metrics in ALLIANCE applications

In [1], the ALLIANCE architecture was demonstrated to have the important qualities of robustness, fault tolerance, reliability, flexibility, adaptivity, and coherence, which we identified as critical design requirements for a cooperative multi-robot team architecture. These demonstrations were achieved through the implementation of ALLIANCE in a wide variety of applications in the laboratory on several different types of physical and simulated robot systems[2]. Table 1 summarizes the metrics we used to analyze the performance of multiple robot teams in eight different ALLIANCE implementations. In these applications, concrete indicators of mission success were used, such as numbers of objects moved, distance traveled, or number of targets within view. Improved mission quality was based upon the time taken to achieve these indicators; this is natural, since a primary benefit of multiple robot teams is using parallelism to achieve mission speedup. In these implementations, no single metric was found to be most useful. The need for a variety of metrics suggests that system performance measures are application-dependent. These examples

[1] This article has been authored by a contractor of the U. S. Government under Contract No. DE-AC05-00OR22725. 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. Research sponsored in part by the Engineering Research Program of the Office of Basic Energy Sciences, U. S. Department of Energy. Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the U.S. Dept. of Energy under contract DE-AC05-00OR22725.

[2] Refer to "http://saturn.epm.ornl.gov/~parkerle/publications" for a list of articles describing these implementations.

Application domain	# Robots	Metric description	Metric definition
1. "Mock" hazardous waste cleanup	2-5 (P)	a. Time of task completion	t_{max}
		b. Total energy used	$\sum_{t=1}^{t_{max}} \sum_{i=1}^{m} e_i(t)$, where $e_i(t)$ is energy used by robot i through time t (m robots)
2. Box pushing	1-2 (P)	Perpendicular dist. pushed per unit time	$d_\perp(t)/t$, where $d_\perp(t)$ is $\perp$ distance moved through time t
3. Janitorial service	3-5 (S)	a. Time of task completion	t_{max}
		b. Total energy used	$\sum_{t=1}^{t_{max}} \sum_{i=1}^{m} e_i(t)$, where $e_i(t)$ is energy used by robot i through time t (m robots)
4. Bounding overwatch	4-20 (S)	Distance moved per unit time	$d(t)/t$, where $d(t)$ is distance moved through time t
5. Formation-keeping	4 (P & S)	Cumulative formation error	$\sum_{t=0}^{t_{max}} \sum_{i \neq leader} d_i(t)$, where d_i = distance robot i is misaligned at t
6. Simple multi-robot manipulation	2-4 (P)	Number of objects moved per unit time	$j(t)/t$, where $j(t)$ is number of objects at goal at time t
7. Cooperative tracking	2-4 (P) 2-20 (S)	Avg. number of targets observed (collectively)	$A = \sum_{t=1}^{t_{max}} \sum_{j=1}^{n} \frac{g(B(t),j)}{t_{max}}$, where $B(t) = [b_{ij}(t)]_{m \times n}$, ($m$ robots, n targets) $b_{ij}(t) = 1 \Longrightarrow$ robot i observing target j at t, $g(B(t),j) = \begin{cases} 1 & \text{if exists } i \text{ s.t. } b_{ij}(t) = 1 \\ 0 & \text{otherwise} \end{cases}$
8. Multi-vehicle production dozing	2-4 (S)	Quantity of earth moved per unit time	$q(t)/t$, where $q(t)$ is quantity of earth moved through t

Table 1: Summary of metrics used in ALLIANCE implementations. (In the second column, "P" refers to physical robot implementations; "S" refers to simulated robot implementations.)

also illustrate that, for typical applications, the most important issues are *whether* and *how well* the robot team completes its mission.

By focusing on application-specific metrics, however, the broader-perspective qualities of robustness, fault tolerance, adaptivity, etc., are not made explicit. Instead, these characteristics are hidden in the application-specific measures. Thus, any shortcomings in a robot team's ability to operate robustly or with a high degree of fault tolerance, for example, would be measured by an increased time to complete the mission (or by never completing the mission at all), a decreased distance traveled, fewer objects moved, etc. It would be difficult, therefore, to determine the relative levels of contribution of the various broader-perspective qualities (e.g., fault tolerance vs. adaptivity) to changes in the application-specific quantitative measures (e.g., distance traveled). Thus, if one wants to explicitly measure fault tolerance across several control architectures, and/or several application domains, these metrics are not suitable. An important goal of research in the quantitative evaluation of robot control architectures is, therefore, the development of metrics that enable quantitative measurement higher-level characteristics, including fault tolerance, reliability, flexibility, adaptivity, and coherence. By averaging the results across multiple application domains, we would then be able to explicitly compare alternative control architectures in terms of these important application-independent characteristics. Our continuing research is aimed at developing these higher-level metrics for the evaluation of robot team performance.

References

[1] L. E. Parker. On the design of behavior-based multi-robot teams. *Journal of Advanced Robotics*, 1996.

[2] L. E. Parker. ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.