INVITED PAPER

Building Multirobot Coalitions Through Automated Task Solution Synthesis

A group of robots can move to, or push boxes to, specified locations by sharing information when individual robots cannot perform the tasks separately.

By Lynne E. Parker, *Senior Member IEEE*, and Fang Tang, *Student Member IEEE*

ABSTRACT | This paper presents a reasoning system that enables a group of heterogeneous robots to form coalitions to accomplish a multirobot task using tightly coupled sensor sharing. Our approach, which we call ASyMTRe, maps environmental sensors and perceptual and motor control schemas to the required flow of information through the multirobot system, automatically reconfiguring the connections of schemas within and across robots to synthesize valid and efficient multirobot behaviors for accomplishing a multirobot task. We present the centralized anytime ASyMTRe configuration algorithm, proving that the algorithm is correct, and formally addressing issues of completeness and optimality. We then present a distributed version of ASyMTRe, called ASyMTRe-D, which uses communication to enable distributed coalition formation. We validate the centralized approach by applying the ASyMTRe methodology to two application scenarios: multirobot transportation and multirobot box pushing. We then validate the ASyMTRe-D implementation in the multirobot transportation task, illustrating its fault-tolerance capabilities. The advantages of this new approach are that it: 1) enables robots to synthesize new task solutions using fundamentally different combinations of sensors and effectors for different coalition compositions and 2) provides a general mechanism for sharing sensory information across networked robots.

KEYWORDS | Coalition formation; information invariants; multirobot teams; schema theory; sensor sharing; task allocation

I. INTRODUCTION

Researchers generally agree that multirobot systems have several advantages over single-robot systems [1], [5]. The most common motivations for developing multirobot system solutions are that: 1) the task complexity is too high for a single robot to accomplish; 2) the task is inherently distributed; 3) building several resource-bounded robots is much easier than having a single powerful robot; 4) multiple robots can solve problems faster using parallelism; and 5) the introduction of multiple robots increases robustness through redundancy. The issues that must be addressed in developing multirobot solutions are dependent upon the task requirements and the sensory and effector capabilities of the available robots. The earliest research in multirobot systems focused on swarm intelligence approaches using homogeneous robot teams, inspired by insect societies (e.g., [3], [28]). In these approaches, individual robots typically perform the same type of subtask in the same environment, resulting in global group behaviors that emerge from the local interaction of individual robots. The fundamental research challenge in these systems is designing the local control laws so as to generate the desired global team behavior.

Other types of robot systems involve heterogeneous robots, which have differing sensor and effector capabilities. In these teams, the mapping of tasks to robots is much more important to the efficiency of the system, since robots vary in the quality of their solutions to tasks. Traditionally, this problem has been called the multirobot task allocation (MRTA) problem. Gerkey [14] has developed a taxonomy for describing these problems, distinguishing robots as either *single-task* (ST) or *multitask* (MT), tasks as either *single-robot* (SR) or *multirobot* (MR), and assignment types as either *instantaneous* (IA) or *time-extended* (TA). The vast majority of prior work on MRTA (e.g., [4],

---

Manuscript received June 1, 2005; revised June 1, 2006.

The authors are with the Distributed Intelligence Laboratory, Department of Computer Science, University of Tennessee, Knoxville, TN 37996 USA (e-mail: parker@cs.utk.edu; ftang@cs.utk.edu).

Digital Object Identifier: 10.1109/JPROC.2006.876933

0018-9219/$20.00 ©2006 IEEE

[8], [16], [23], [31], [45], [47], [48]) has addressed single-task robots executing single-robot tasks using either instantaneous assignment (denoted ST-SR-IA) or time-extended assignment (denoted ST-SR-TA).

In this paper, we address a different problem in the MRTA taxonomy—namely, single-task robots performing multirobot tasks using instantaneous assignment (ST-MR-IA). In other words, we are addressing the development of heterogeneous robot coalitions that solve a single multirobot task. While this problem has been addressed extensively in the multiagent community (e.g., [24], [36], [37]), it has been noted by Vig [44] that most of the multiagent approaches to coalition formation cannot be directly transferred to multirobot applications, since robot capabilities and sensors are situated directly on the robots and are not transferable between robots. Our approach is aimed at enabling sensor-sharing across robots for the purpose of forming coalitions to solve single-multirobot tasks.

More generally, multirobot *coalition formation* deals with the issue of how to organize multiple robots into subgroups to accomplish a task. *Coalitions* are typically considered to be temporary organizations of entities that bring together diverse capabilities for solving a particular task that cannot be handled by single robots. Coalitions are similar to the idea of *teams*, except that they typically have a shorter duration and can change frequently over time. We are particularly interested in automated techniques for coalition formation, especially when the specific task solution is highly dependent upon the available capabilities of the heterogeneous robots, and thus cannot be specified in advance. This is especially challenging in heterogeneous robot systems, in which sensory and computational resources are distributed across different robots. For such a group to accomplish the task as a whole, it must determine how to couple the appropriate sensory and computational capabilities from each robot, resulting in automatically formed coalitions that serve specific purposes.

To address this challenge, we present our approach called ASyMTRe (which stands for "Automated Synthesis of Multirobot Task solutions through software Reconfiguration," pronounced like the word "asymmetry"), which we first introduced in [41], [42]. This approach is aimed at increasing the autonomous task solution capabilities of heterogeneous multirobot systems by changing the fundamental abstraction that is used to represent robot competences from the typical "task" abstraction to a biologically inspired "schema" [2], [27] abstraction, and providing a mechanism for the automatic reconfiguration of these schemas to address the multirobot task at hand.[1] In doing this, we are able to simultaneously obtain a number of significant new benefits in multirobot coalition formation that have previously been difficult to achieve. These benefits include: 1) enabling robots to automatically generate task solutions based on sensor-sharing across robot coalition members, in configurations not previously explicitly defined by the human designer; 2) providing a way for robots to develop coalitions to address multirobot tasks; and 3) enabling flexible software code reuse from one multirobot application to another through the task-independent schema abstraction that is viewed as a generator of semantic information content which can be combined in many ways by various diverse tasks. Eventually, we expect that the ASyMTRe approach can be layered with prior task planning/allocation approaches, with ASyMTRe serving as a lower level solution generator for generating a coalition to solve single-multirobot tasks. The coalitions would then compete (with other coalitions or single robots) for task assignments using the higher level, more traditional task planning/allocation strategies.

The basic ASyMTRe approach is an anytime centralized reasoner, generating multirobot coalitions using complete information, with solution quality increasing as more time is available for the reasoning process. In order to allow for increased robustness, we also present a distributed version of ASyMTRe, called ASyMTRe-D (which we first introduced in [43]), which uses communication to enable distributed formation of coalitions. This distributed version offers a tradeoff of increased robustness versus solution quality compared to the centralized version. Our ultimate objective in this research is to eventually enable the human designer to specify the desired balance between solution quality and robustness, enabling the reasoning approach to invoke the appropriate level of information-sharing among robots to reach the specified solution characteristics.

The rest of this paper is organized as follows. Section II describes the centralized ASyMTRe solution approach. Section III analyzes the theoretical soundness, completeness, and optimality of this approach. In Section IV, we describe the distributed version of ASyMTRe, called ASyMTRe-D. Section V describes the experimental results that validate this approach. We then present a review of related work in Section VI and conclude in Section VII.

II. THE ASYMTRE APPROACH

A. Schema Theory and Information Types

The basic building block of our approach is a collection of *schemas*, inspired by the work of Lyons and Arbib [27] and Arkin [2], which first applied schema theory to cognitive and agent systems. In [27], a formal model of computation is constructed, called robot schemas. In this earlier work, the schema includes a list of input and output ports, a local variable list, and a behavior, which defines how the input is processed to generate the output. A network of schemas can be built by manually connecting

---

[1] Our approach does not necessarily require a schema implementation, and could alternatively be implemented using traditional "behaviors." We selected the schema approach because schemas tend to be more fine-grained and less sensor- and task-specific than behaviors. Additionally, schemas have a more formal, commonly accepted interface definition, as defined by [27], which facilitates the formal ASyMTRe specification.

the outputs of one schema to the inputs of another schema. At the higher level, a nested network is established to represent the collaboration among robots. Arkin [2] further develops these ideas by presenting schema-based control for mobile robots. In his approach, perceptual schema interpret sensory stimuli, feeding the results to one or more motor schemas. Computations from multiple motor schemas are summed and normalized to generate the overall robot behavior.

Based upon this prior work, the fundamental building blocks of our approach are collections of environmental sensors (ES), perceptual schemas (PS), and motor schemas (MS). Our ASyMTRe approach extends this prior work by introducing a new component, called *communication schemas* (CS). Perceptual schemas process input from environmental sensors to provide information to motor schemas, which then generate output control vectors representing the way the robot should move in response to the perceived stimuli. Communication schemas transfer information between various schemas distributed across multiple robots. All of these schemas are assumed to be preprogrammed into the robots at design time, and represent fundamental low-level capabilities of individual robots.

Our ASyMTRe approach further extends this prior work on schema theory by autonomously and dynamically connecting the schemas at run time instead of using predefined (manual) connections. In our approach, schemas are situated in each robot, but are not connected to each other at the beginning of a task. Instead, they are configured using our automated ASyMTRe approach. Our automation is based upon recognizing and defining the fundamental information that is needed to accomplish the task. The information needed to automatically activate a certain schema remains the same regardless of the way that the robot may obtain or generate it. Thus, we label inputs and outputs of all schemas with a set of information types that are unique to the task. Note that we use the term *information types* as distinct from *data types*. This distinguishes our approach from Lyon [27]. *Information types* have semantic meaning and define the specific sensing or computational data of a schema or a sensor, such as the *global position* of a robot, rather than just the format of the data. Semantics of the information is built into these information types, and does not just refer to a data type (such as boolean or integer). These information labels provide us a method for automating the interconnections of schemas, enabling robots to share sensory and perceptual information as needed to accomplish the multirobot task.

We define the inputs and outputs of the schemas to belong to the set of information types $F = \{F_1, F_2, \ldots\}$. For schema S_i, I^{S_i} and $O^{S_i} \subset F$, represent the input and output sets of S_i, respectively. As in [27], we assume that each schema has multiple inputs and outputs. There are two types of inputs to a schema (see Fig. 1). The solid-line arrows entering a schema represent an "OR" condition, meaning that it is sufficient for the schema to only have one of the specified inputs. The dashed-line arrows represent an "AND" condition, where all the indicated inputs are needed to produce a result. For example, in Fig. 1, MS_1 can calculate output only if it receives both F_1 and F_2. However, PS_2 can produce output based on either the output of PS_1 or CS_1. An output of a schema can be connected to an input of another schema if and only if their information types match. Using the mapping from schemas to information types, robots can collaborate to define different task strategies in terms of the required flow of information in the system. Once the interconnections between schema are established, the robots have executable code to accomplish their task.

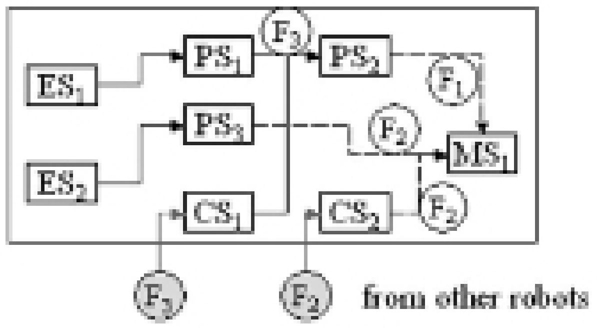

Fig. 1. *An example of how the schemas are connected to accomplish a task. (For clarity, we have eliminated the superscripts in this figure.)*

Table 1 Connection Constraints for Schemas

Sensor/Schema	Input Sources:	Output Feeds into:
ES	Sensor Signals	PS
PS	ES, PS, or CS	PS, CS or MS
CS	PS, or CS	PS, CS, or MS
MS	PS, CS, or ES	Actuators

Given a set of n robots and a task T, the solution configuration problem can be represented as (R, T, U), where $R = \{R_1, R_2, \cdots, R_n\}$ is the set of n robots, $T = \{MS_1, MS_2, \cdots\}$ is the set of motor schemas that define the group-level task to be achieved, along with application-specific parameters as needed,[2] and U provides utility information to be defined later. A robot R_i is represented by $R_i = (ES^i, S^i)$. ES^i is a set of environmental sensors that are installed on R_i, where $O^{ES^i_j} \subset F$ is the output of ES^i_j (that is, the jth ES on robot R_i). S^i is the set of schemas that are preprogrammed into R_i at design time. Each schema is represented by $(S^i_j, I^{S^i_j}, O^{S^i_j})$. A schema can be activated if and only if its input can be obtained from the output of another schema or sensor. A set of *Connection Constraints* regulate the connections between schemas. As shown in Table 1, these constraints specify the restrictions on correct connections between various schemas.

---

[2] More complex task definitions, including task sequences, can be defined in a manner similar to the formal, schema-compatible, specification of tasks in [13].