A Multiagent System for Structural Health Monitoring

Gina Bullock
Dept. of Computational Sci. & Eng.
North Carolina A&T State University
Greensboro, NC 27411
1-336-334-7717
glbulloc@ncat.edu

Albert Esterline
Dept. of Computer Science
North Carolina A&T State University
Greensboro, NC 27411
1-336-285-2440
esterlin@ncat.edu

ABSTRACT

This faculty —paper reports on work on structural health monitoring done for NASA at North Carolina A&T State University. We are developing a system where a hierarchically structured collection of monitor agents monitors the health of a structure by interpreting acoustic signals. This paper addresses the problem of how the main agents (the "monitor" agents) can get organized into a hierarchy that reflects the hierarchy in the monitored structure. The agent framework used is JADE. The alternative approaches are worked out for a structure much simpler than an aircraft, namely, a plate equipped with two kinds of sensors. The approaches we consider are the contract net protocol (CNP), the iterated CNP, and coalition game theory. The first two are distributed approaches provided with the JADE framework. The third is centralized; we implement it in the Python programming language. The iterated CNP generally produces better solutions than the basic CNP, and coalition game theory generally produces solutions better than the iterated CNP but requires more resources (including information). And the iterated CNP requires more resources than the basic CNP. We suggest when use of the various approaches is appropriate.

The faculty advisor for the work reported here is Dr. Albert Esterline, Dept. of Computer Science, North Carolina A&T State University, esterlin@ncat.edu.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents*, *multiagent systems*.

General Terms

Design, Experimentation

Keywords

Multiagent systems; Contract net protocol; Coalition game theory

1. INTRODUCTION

Safety is a major concern in today's society. Whether it is police securing a city or a teacher monitoring a class room, certain protocols have to be in place to carry out what is needed. In the case of an aircraft, this need is life critical. The research reported here addresses monitoring the health of a mechanical structure with a multiagent system. The target structures have been aircraft, but other important targets include bridges and buildings. The fundamental techniques interpret acoustic signals for clues regarding the formation and growth of cracks, but the information on cracks in specific members must be interpreted within the context of the entire structure. Our multiagent system is a hierarchical structure that has a collection of agents that monitor the health of an aircraft structure. The monitor agents in the lowest part of the hierarchy are associated with sensors on given regions. They negotiate with agents that advocate for specialize techniques to find one suitable for classifying the events associated with the signals. The hierarchy reconfigures dynamically, and problem solving follows the structure of the hierarchy on an as-needed basis.

In monitoring the structure of an airplane, the data must be processed not only in an intelligent and flexible way but also in near real-time. Our solution involves a multiagent system that sends a message to a workflow engine with the specification of the tasks. We set up our workflow in a sequence (pipeline) pattern where each task (stage) is its own individual process. In our setup, agents are advocates for certain computation-intensive techniques. The agents must have enough intelligence so that they can work out which techniques are appropriate for the current situation and how these techniques must connect to each other in a workflow. An agent submits a multitask request to the workflow system and attaches input and output sources. So agents are the "brains" and the workflow is the "brawn."

This paper addresses the problem of how the main agents (the "monitor" agents) can get organized into a hierarchy that reflects the hierarchy in the monitored structure. The agent framework used is JADE. The alternative approaches are worked out for a structure much simpler than an aircraft, namely, a simulated plate equipped with two kinds of sensors. The approaches we consider are the contract net protocol (CNP), the iterated CNP, and coalition game theory. The first two are distributed approaches provided with the JADE framework. The third is centralized; we implement it in the Python programming language.

As far as we know, the wok reported here involves the first use of iterated CNP or coalition game theory to assign tasks to agents in the context of structural health monitoring..

In the rest of this paper, we first describe, in the following section, the architecture of our system independent of particular implementation issues. The next section describes the software used to implement the system, and the section after that concludes.

The rest of this paper is organized as follows. The next section briefly introduces multiagent systems and presents the CNP. Section 3 presents the iterated CNP, and Section 4 presents coalition game theory. The simulated plate is described in Section 5. Section 6 describes how the various approach for organizing the agents are applied for monitoring the plate. The last section compares the three approaches in this section and concludes.

2. MULTIAGENT SYSTEMS AND THE CONTRACT NET PROTOCOL

According to Wooldridge [1], multiagent systems address five main trends that have driven the advances in computing: ubiquity, interconnection, intelligence, delegation, and human-orientation. An agent is autonomous because it has control over its actions and internal state, social because it works with others in order to achieve goals, reactive because it responds to changes that occur

in its environment, and proactive because it has goal-oriented behavior.

The Contract Net Protocol (CNP) is a task-sharing protocol for a collection of software agents that form the 'contract net' [2]. It was developed to specify problem-solving communication and control for nodes in a high-level distributed problem solving protocol by a negotiation process. Each agent in the network can, at different times or for different tasks, be a manager or a contractor. The CNP controls the flow of contracting and subcontracting throughout the network of agents. The CNP is designed to allow agents to break down tasks (or problems) that they cannot efficiently handle on their own into more manageable, smaller subtasks (or sub-problems) using delegated agents to complete the task. Agents can bid on tasks based on their capabilities.

In the CNP, the manager initiates the protocol by announcing the subtask, and the participants (or prospective contractors) are those agents willing to perform the subtask. The manager decomposes the problem into sub-problems and *announces* each sub-problem to all agents. This is a *call for proposal*s (*cfp*). All available contractors may reply with a *bid* that includes information about the resources and time needed to solve the sub-problem. The manager reviews the bids and *awards* the contract to the best (from its point of view) bid. The contractor reports the solution back to the manger, which compiles a solution to the overall problem from the solutions to the sub-problem.

If the task decomposition leads to multiple levels of subtasks, the CNP determines a hierarchy of the agents defined by the manager-contractor relation. Not only can the CNP be used to set up a hierarchy of agents to carry out a particular task over long a period of time (such as monitoring an area of a structure), but it can also be used to solve problems on the fly.

3. ITERATED CONTRACT NET PROTOCOL

The iterated contract net protocol is an extension of the basic FIPA CNP and is another FIPA standard for negotiation between agents [3]. The major difference from CNP is that it allows multi-round iterative bidding and allows multiple contractors to be chosen for a subtask (and thus team formation). From the participants who send proposals, the manager may accept the intended number and thus complete the protocol. But it may decide to issue a revised call-for-proposals to a subset of those that bid. Note that the iterated CNP requires more time, communication, and computation than the simple CNP.

In the iterated CNP, the manager issues the initial call for proposal. Participants answer with their bids. The manager has a choice to accept one or more bids and reject the others. It iterates by sending a revised CFP to get better bids. The protocol terminates if the manager declines all proposals and does not issue a new CFP, the contractors all decline to bid, or one or more bids are accepted.

4. COALITION GAME THEORY

There is a greedy aspect to the CNP, which may result in clearly sub-optimal arrangements. A more radical alternative we consider is coalition (or cooperative) game theory [4]. In this, unlike the better known (competitive) game theory, binding agreements are made. Utility in the first instance accrues to groups (coalitions), not individuals. Coalition game theory addresses such questions as which are (stable) coalitions that might be formed by rational agents, and how the pay-off received by a coalition might be reasonably divided among its members.

A coalitional is a set of agents, and we consider what coalitions could be formed. The key is to find how well each group can do for itself. We also consider how the coalition allocates the team's payoff to the member agents. We are not concerned with how the agents make individual choices within a coalition and how they coordinate. We assume transferable utility: a coalition can redistribute the value it tries to achieve arbitrary among members. For example: if a coalition is paid its value in money, it will be possible to divide the money and to make side payments amongst the members in anyway.

A coalitional game is a pair, (N,v), where N is a finite set of players and v is the characteristic function for the game. For every coalition $S \subseteq N$ that can be formed, v(S) is the payoff that S can achieve (and divide amongst its members). An *outcome* of a transferable utility game $G = (N, v)$ is a pair $(CS, \underline{x})$, where $CS = (C_1, ..., C_k)$ is a coalition structure, (i.e., partition of N into coalitions), and $\underline{x} = (x_1, ..., x_n)$ is a payoff vector, which distributes a value x_i to each coalition C_i in *CS*. The *core* of a game is the set of all stable outcomes, i.e., outcomes in which no player is motivated to defect from its coalition. The core is a very attractive solution concept, but some games have empty cores.

A fair payment scheme would reward each agent according to his contribution, and this is captured by the notion of the Shapley value φ_i of player i. φ_i is *i*'s average marginal contribution to the coalition of its predecessors, over all permutations of the order in which players enter coalitions. If we choose a permutation of players uniformly at random, among all possible permutations of *N*, then φ_i is the expected marginal contribution of player *i* to the coalition of his predecessors. The vector made up of φ_i for all $i \in N$ is a payoff distribution and, in fact, this scheme is the only payoff distribution scheme with several important desirable properties.

When coalition game theory is used in artificial intelligence, probably the most pressing issue is finding an optimal coalition structure, that is, a structure with the greatest overall payoff. Approaches include dynamic programming that exploits optimal sub-structures (Rahwan and Jennings 2008) and representations of the search space that guarantee that, after a certain amount of search, the solution is within a certain bound from the optimal (Sandholm, Larson, Andersson, Shehory and Tohme, 1999). Dynamic coalition formation has been studied as a Markov chain (Dieckmann and Schwalbe, 1998).

5. THE MONITORED PLATE

The current, less ambitious architecture involves a single member, as in the schematic diagram in Fig. 1. The two kinds of sensors are piezoelectric sensors (which record acoustic waves and so detect cracks forming at a distance) and optic fiber (which capture the strain distribution across its length). We may include strain gauges. Sensors are distributed over the plate, and there are more sensors than there are observed data streams. Agents determine what data streams are used in response to inferred conditions. The different kinds of sensors are combined in intelligent ways, and areas are combined hierarchically.

6. APPLICATION OF THE CNP, ITERATED CNP, AND COALITION GAME THEORY

This section sketches how the CNP, the iterated CNP, and coalition game theory have been used in our prototype. We consider the simple case of the plate with sensors attached depicted in Fig. 1. For the CNP and iterated CNP, we used the Java Agent Development Environment (JADE) framework [5] to implement our agent system. JADE was preferred over other agent frameworks because it is the most widely used and mature agent framework and uses FIPA standards. JADE allows communication between agents either on the same or different platforms. For coalition game theory, we used the Python programming language to find coalitions. We intend to integrate this mechanism with the JADE implementation of the multiagent system.

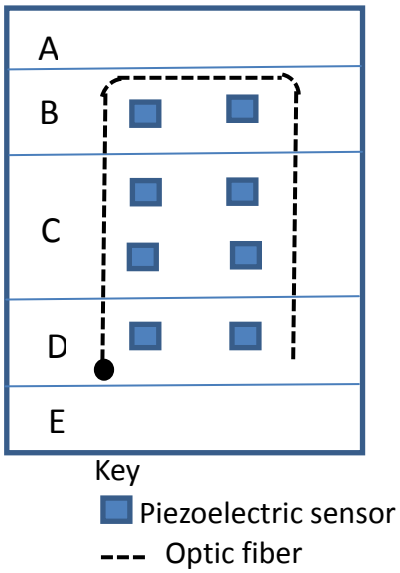

Figure 1: Current Architecture: Single Member with Sensors

6.1 Application of the CNP

The monitor agents are given access to the schema of the plate: locations of devices, functional blocks, spatial layout, and an expression for the likelihood of crack initialization that is a function of the location on the plate. There are three subtasks here: monitor the plate where it is supported at A and again at B, and monitor it in the middle, at C. The middle is the most likely area for crack initialization, but the regions where the plate is supported are also vulnerable. The sensors in B provide the best data on region A although the top two sensors in C can provide some information. Similarly, the sensors in D provide the best data on F although the bottom two sensor in D provide some. We provide a high-level description of how the system works.

The topmost monitor agent will first announce the task of monitoring the center of the plate. The agents that tend to bid for this task and whose bids will be most promising are those that are proxies for computational tasks that combine a reasonable number of classification streams derived from PZT data as well as strain data from the optic fiber. The top monitor agent will then announce the subtask of monitoring one of the support areas, say, A. The monitor agents that will do well here will be ones good at combining a small classification streams from PZT sensors but able to add a few more streams if required. (The optic fiber is not useful for monitoring A or F since it does not record waveforms.) After a certain monitor agent is awarded the task of monitoring region C, it will determine what kind of information it needs and how much. It will issue announcements for classification streams. Various monitor agents that are appropriate for classifying events from streams of PZT data (which involves using the CNP to find appropriate feature extraction and classification agents/techniques) will associate with the various data agents paired with the sensors. These monitor agents will participate in the CNP for classification streams. One agent-stream pair might be enough initially, or the monitor agent for C might decide two are needed, and initiate another CNP instance. The monitor agents for regions A and E will similarly perform instances of the CNP to award monitoring tasks to one or two monitor-senor pairs each.

If one of the classification streams from region C suggests crack formation, the monitor agent for C might initiate another instance of the CNP to obtain another classification stream. Likewise, if one of the classification streams in B or D suggests crack formation in A or E, the monitor agent for A or E might try to get another data stream by awarding a subtask to a new monitor-sensor pair. If the situation is critical, one of these monitor agents may allocate a third agent-sensor pair, taking a neighboring one from region C. Note that, if things are quiet, the monitor agent for a region may release an agent from an agent-sensor pair to conserve computational and communication resources. The sensor thus freed may be needed by a monitor agent for another region. Where there is contention for sensors, the parent (or lowest common ancestor) of the two contending monitor agents decides.

Note that two monitor agents higher in the hierarchy may tap information from the same agent lower in the hierarchy, and in this sense the so-called hierarchy is not tree-shaped (as expected of a hierarchy). But there are certain control relations that are maintained (e.g., requesting specific data), and each monitor agent engaged in the monitoring has a unique parent that is the ultimate arbiter of what it may do.

If all contractors persist throughout (i.e., there is no notion of completing the task) and there are more subtasks than agents in the system, then some subtasks will go unassigned. For performance, we do not allow an unbounded number of agents. We can try various policies for creating additional agents as needed, or we can have a policy for which tasks may be left unassigned.

6.2 Application of the Iterated CNP

Consider again the case where the monitor agent for region C initiates instances of the CNP to reallocate agent-sensor pairs. This is a perfect opportunity for using the iterated CNP. The monitor agent could accept several bids in light of which it could refine its cfp and eventually accept the one or several agent-sensor pairs required. The agents have some measure of how well they collaborate with the other agents. Note that this may avoid sub-optimal allocations due to the greedy nature of the CNP.

6.3 Application of Coalition Game Theory

For the Python code, we define a class **Coalition**. An instance has attributes for (1) the coalition's high-level monitor agent, (2) its set of level-one monitor agents, and (3) its characteristic function value (a non-negative real number). Using coalition game theory, we work from the bottom up. This is somewhat like the approaches (such as (Rahwan and Jennings 2008)) that have used dynamic programming to construct optimal coalition structure (i.e., the structure with the greatest overall value) from optimal solutions to sub-problems. In our case, however, possible lowest-

level coalitions are severely constrained by the need to form teams of agents with complementary competences. The payoff for a coalition is some measure of its contribution to the overall assurance of the integrity of the monitored structure minus a measure of the resources used. For the plate in Fig. 1, we would expect that, for a structure with the greatest overall value, the sensor-agent pairs in B would form one coalition, those in D would form another, and those in C would form a third.

Since the coalition structure will be hierarchical, each of these coalitions will include an agent that works as their representative. This is exactly the monitor agent in the CNP served as the manager for the contractors that are now seen as members of the same coalition.

7. CONCLUSION

The motivation for this work is the problem of monitoring the health of mechanical structures, especially aircraft structures. We aim for a multiagent solution, where agents are generally proxies for services provided by a workflow engine. Monitor agents form the back bone of the system, and we expect them to arrange themselves into a hierarchy mirroring the hierarchy of the monitored structure. A monitor agent generally is responsible for the union of the regions monitored by its children in the hierarchy and may collaborate with agents with special functionalities. For example, monitor agents lowest in the hierarchy would generally collaborate with feature-extraction and classification agents. In this arrangement, agents are the "brains" (determining structure and communication) while the workflow engine is the "brawn." This paper consider three approaches for structuring the monitor-agent hierarchy: the Contract Net Protocol (CNP), the iterated CNP, and coalition game theory. The first two are distributed, and we have implemented them in the JADE agent framework. Agent allocation done by coalition game theory, in contrast, is centralized, and we have implemented it in Python. Implementations monitor a plate instrumented with several sensors of two types.

The lowest-quality agent organizations are produced by the CNP, which takes a greedy approach to assigning agents. Generally better quality organizations are produced by the iterated CNP, which mitigates the greedy nature of the basic CNP. The best organizations are generally produced with coalition game theory. Ordering these approaches according to the computational resources required reverses the order, with coalition game theory requiring the most, and the CNP require the fewest resources. This tradeoff is to be expected. A similar ordering arises if we consider just the information required. With the iterated CNP, we assume (beyond the CNP) that the agents have some measure of how well they collaborate with the other agents they might be expected to work with. With the coalition-game-theory approach, the agents need considerably more information (such as the current context and exactly what competences other agents provide) since there is no direction from a manager agent. The centralized nature of coalition game theory might also be considered a disadvantage.

From these considerations, it is clear how these approaches may be effectively combined. We would use coalition game theory for the initial organization. Adjustments would be made using the iterated CNP. And mall, ad hoc changes would be made with the basic CNP. In future work, we shall implement all three approaches in one multiagent system. The main challenges here are how to use resources, including information.

8. ACKNOWLEDGEMENTS

The authors would like to thank the Army Research Office (proposal number 60562-RT-REP) and NASA (Grant # NNX09AV08A) for financial support. Thanks are also due to members of ISM lab (and Dr. Mannur Sundaresan of the Mechanical Engineering Dept.) at North Carolina A&T State University for their assistance.

9. REFERENCES

[1] Wooldridge, M. 2009. *An Introduction to Multiagent Systems*. John Wiley & Sons.

[2] Smith, R. The contract net protocol: Highlevel communication and control in a distributed problem solver, 1980. *IEEE Trans. on Computers, C, 29*, 12.

[3] Foundation for Intelligent Physical Agents (FIPA), "FIPA Iterated Contract Net Interaction Protocol Specification," 2002, http://www.fipa.org/assets/XC00030G.pdf.

[4] Osborne, M. J., & Rubinstein, A. 1994. *A Course in Game Theory*. MIT press.

[5] Bellifemine, F.L., Caire, G., and Greenwood, D. 2007. *Developing Multi-Agent Systems with JADE*. Wiley, Hoboken, NJ.