Autonomous Agents with Application to the Evaluation of Organizational Structures[1]

Michael L. Curry
ALPHATECH, Inc.
50 Mall Rd., Burlington, MA. 01803
e-mail: michael.curry@alphatech.com

Abstract

Experimental investigation of adaptive command and control organizations is limited in scope by the availability of qualified subjects and the complexity of the experimental design and analysis for large organizational structures. These limitations challenge the study of adaptive architectures for command and control (A2C2) to represent a realistic command and control environment with a small number of human participants. This paper presents a method of representing large organizations by introducing autonomous agents that simulate additional decision-makers. These agents not only interact with the human participants via message communication, but also interact with the environment, which indirectly affects the human participants and contributes to a more realistic environment. Since the agents act as additional uncontrolled factors and increase the variability of the experiment, it is important to be able to control their actions such that the variability is minimized (or at least controlled). In this paper, the controllability issue is addressed by scripting agents' actions. This paper also identifies some of the challenges and suggests a course of action in the development of truly interactive and collaborative agents for developing, assessing, and training large C^2 organizations.

1. Introduction

The study of command and control architectures in the A2C2 program involves the theoretical development and experimental analysis of organizational structures. To date the size of the organizations studied have been small, limited by the complexity of command and control system and the logistics associated with experimentation. As a better understanding of C^2 organizational structures is developed, experimental limitations will significantly handicap further studies. In particular, the low levels of communication, coordination, and command authority present in small organizations do not correspond to those that exist in typical C^2 organizations. Therefore, in order to understand command and control architectures, larger organizations need to be considered.

In recent experiments, the Distributed Dynamic Decision-making (DDD) simulation environment [Kleinman *et al*, 1996] has been used to experiment with relatively small organizations. Experiments with larger organizations require many properly trained participants. However, only a small number of qualified subjects are available. In order to increase the size of the organizations, this paper proposes the use of autonomous agents to simulate additional decision-

[1] This work was supported by the Office of Naval Research under contract #N00014-95-C-0125

makers in the DDD environment. These agents will interact with the environment, communicate with other decision-makers, and support experimental goals. Unfortunately, these agents complicate experimental logistics by introducing additional uncontrolled factors in the same way that additional human participants would. One way of controlling these factors is to constrain the agents' activity such that they contribute similarly under each experimental condition. In this case, the impact of the agents will be uniform across all the experimental conditions and will not bias the independent experimental variables. Scripting the agents' actions provides a near-term solution for current organizational experiments. However, this limits the flexibility and autonomy of the agent. The development of intelligent agents for experimentation will expand the capabilities and role available to agents in future experiments.

Section 2 introduces the DDD environment and develops a framework for representing the basic decision process. Based on this framework, scripted agents are introduced in Section 3. Intelligent agents are introduced in section 4 by expanding the collaborative and interactive capabilities of the scripted agent. The paper is concluded in section 5.

2. The Decision Process

In the DDD environment, each decision-maker is able to observe the area of interest (AOI), control a set of platforms[2], and communicate with other decision-makers. A visual display characterizes the AOI including coarse geographical features and an iconic representation of tasks[3] and platforms. Tasks and platforms are further characterized by a set of attributes; some are represented graphically and others numerically. A set of platforms controlled by each decision-maker allows interaction with the environment. These platforms provide a capability to prosecute tasks as well as obtain information via sensors. Communication is facilitated via electronic messaging and data transmission (e.g., transferring sensor data). The DDD also simulates the scenario activity that participants must interact with in order to achieve the objectives outlined in the operational orders for the mission.

To achieve mission objectives, a sequence of decisions are made regarding the manipulation of platforms (i.e., move, attack, pursue), the gathering of information, and communication/coordination with other decision-makers. Specifically, a decision-maker needs to be able to determine when and where to move platforms and under what circumstances to attack or pursue the enemy. Platform movement is restricted both logistically (i.e., limited fuel) and geographically (i.e., air, sea, or ground), and each platform has a limited range from which it may obtain sensor information or attack. Therefore, a decision-maker must coordinate platform movement with other activities (i.e., identify, attack) in order to achieve the desired objective. In the decision process listed below, the objective is to identify and prosecute an unidentified task.

1) **Can the platform be moved within sensor range?**
2) **Move to (x_1, y_1).**
3) **Evaluate sensor data (friend or foe).**
4) **Identify type (helicopter, fighter aircraft).**
5) **Is the platform equipped to prosecute?**

[2] Platforms indicate the resources available to each decision-maker (e.g., Infantry, CAS aircraft, Engineering).
[3] The term "task" will be used categorically to refer to tasks, and neutral or enemy platforms.

6) Can the platform be moved within attack range?

7) Move to (x_2, y_2).

8) Attack.

9) Return home.

This decision process may be translated directly into a simple decision tree shown in Figure 1. Each line in the decision process above is represented as a box in the decision tree.

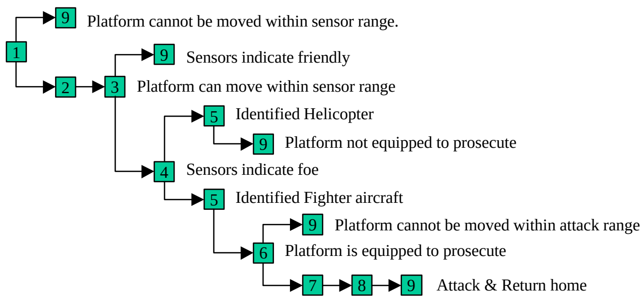

Figure 1 Decision Tree

The decision tree in Figure 1 directs the decision-maker through a sequence of decisions. First the decision-maker must determine whether the platform can be moved within sensor range. If the platform can be moved within sensor range (based on geographic and logistic capabilities), the sensors are evaluated. If the task is identified hostile, further evaluation will indicate the task type. In this example, the platform is not equipped to prosecute a helicopter. However, if the task is identified as a fighter aircraft, the platform is moved into attack position and an attack is made. This simple representation of a decision process is used in the next section to develop scripted agents.

The decision process may be augmented to include complicating factors such as multiple decision-makers and uncertainty. Multiple decision-makers may be cooperative or non-cooperative (enemy). In either case, additional decisions are added to the decision tree. In multiple decision-maker processes, only a subset of the decisions is available to be made by each decision-maker. Uncertainty also introduces additional branches to the decision process. These decisions are often attributed to “nature”.

The sequence of activities outlined in the decision tree must have two basic properties, *completeness* and *correctness*. Completeness of a decision policy is the ability of the policy to handle all the possible decision states. Correctness describes the flow of the strategy. A correct strategy will proceed through a sequence of feasible decision states (i.e., the result of each decision will lead to a valid and well-defined decision state) and terminate at a feasible terminal state in a finite number of steps. Both correctness and completeness are mathematically defined and can be rigorously applied to an appropriately defined strategy.

3. **Scripted Agents**

Scripted agents simulate the behavior of human decision-makers in the DDD environment by performing a sequence of activities that are identified in a predefined script. This script corresponds to a decision process and guides the agent to achieve a specific objective. The agents' activity is initiated via an electronic message that specifies a particular script. Once the script is specified, the agents operate independently and interact with the DDD environment similar to their human counterparts.

Each script is a combination of observation commands, action commands, and simple conditional logic. Observation commands allow the agents to extract information (e.g., geographical location, attributes, and messages) from the DDD environment. Action commands allow the agent to interact with the environment and other decision-makers (e.g., *move*, *pursue*, *attack*, and *send* message). Evaluation commands provide a basic reasoning capability based on propositional logic. This simple logic allows basic conditional statements within the script (e.g., if-then-else).

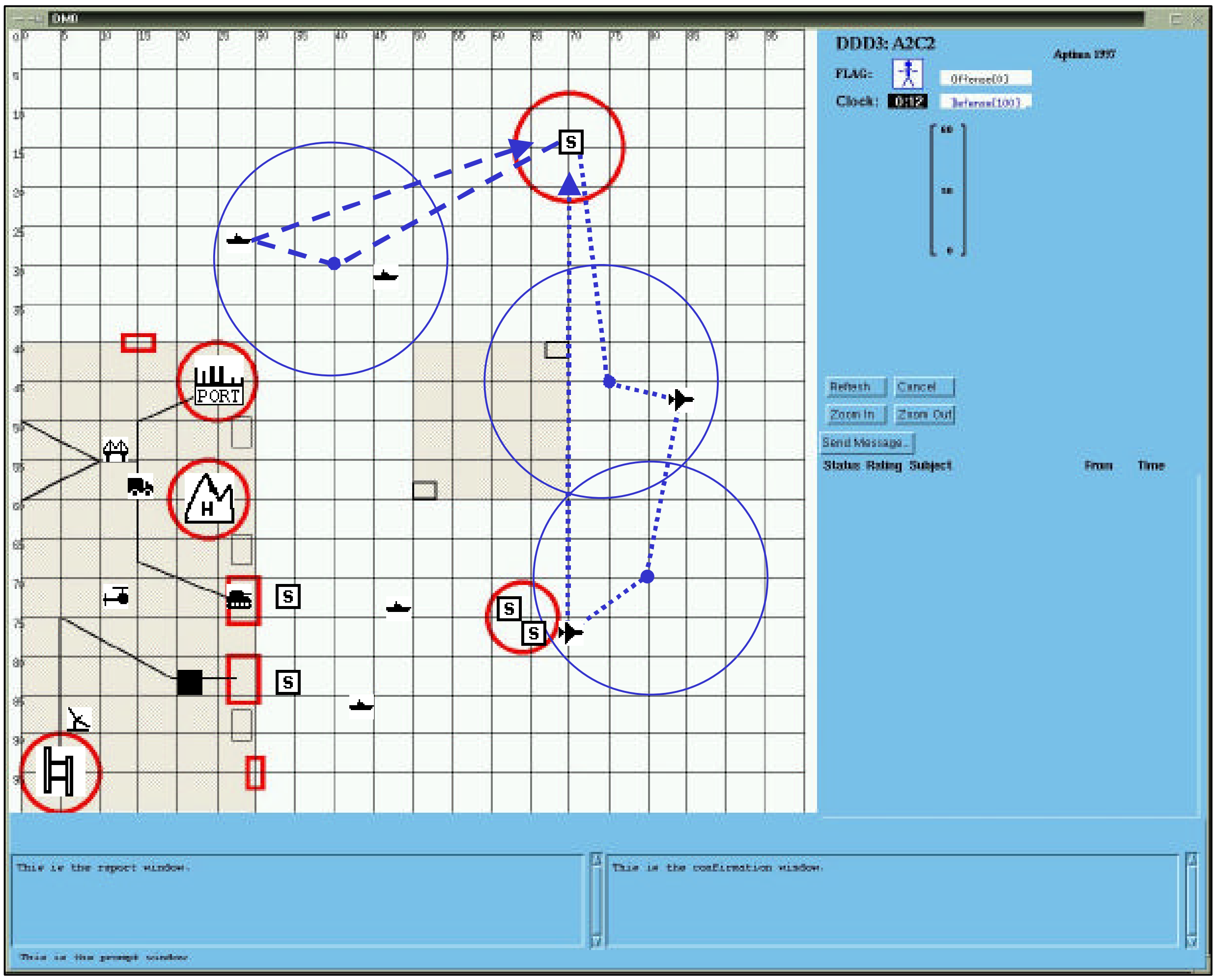

Figure 2 Sample Scenario: Script A (dashed line), Script B (dotted line)

The scenario in Figure 2 is representative of a joint amphibious landing scenario used in recent A2C2 experiments. The blue objects have been added to annotate the activity of the scripted agent that is stationed on an aircraft carrier in the northern region of the scenario (denoted by an "S" in the top portion of the figure). This agent controls an air asset that is used to search and destroy enemy aircraft in the region. This agent has two scripts (strategies) available. The agent will be prompted to carry out one of these strategies at a time.

The objective of Script A is to search an area in the vicinity of location (40,30) just west of the aircraft carrier. If an enemy aircraft is found within sensor range the agent will attack. A status message is sent prior to leaving the area.

Script A (denoted by a dashed line)
Move **from aircraft carrier to position (40, 30)**
Evaluate **objects within sensor range**
If **object is enemy (i.e., attribute values exceed predefined thresholds)**
Attack **enemy**
Report **status**
Return **to aircraft carrier**

The objective of Script B is to search two areas south of the aircraft carrier. If an enemy aircraft is identified, it is attacked. A status message is sent prior to leaving each area.

Script B (denoted by a dotted line)
Move **from aircraft carrier to position (75,45)**
Evaluate **objects within sensor range**
If **object is enemy (i.e., attribute values exceed predefined thresholds)**
Attack **enemy**
Report **status**
Move **from aircraft carrier to position (80,70)**
Evaluate **objects within sensor range**
If **object is enemy (i.e., attribute values exceed predefined thresholds)**
Attack **enemy**
Report **status**
Return **to aircraft carrier**

The impact of the agent on the DDD environment and the awareness provided to other decision-makers by the agent simulates the collaborative effects of a larger C^2 organization.

The script outlines a fixed sequence of activities to accomplish a specific objective. Therefore, the agent does not need to look-ahead beyond the current activity. Although, these activities include interaction with the DDD environment and communication to other decision-makers, the script is carried out independent of other DDD activities. The independent activities of the agents help to ensure the correctness of the script since dependent actions could leave the agent in a waiting state indefinitely. The script is complete since only binary decisions are used. If the attack command were based on the type of task identified, correctness would require that every possible type be accounted for.

There are several ways that the scripts can be used to control the experimental impact of the scripted agents. Therefore the development of the script as well as the identification of agent responsibilities and activities within the experimental scenario have become significant components of the experimental design. In general, assigning agents to perform relatively independent tasks reduces the impact of the agents on independent experimental variable.

4. **The Foundation for Intelligent Agents**

Scripted agents provide the necessary foundation for the further development of truly interactive and collaborative agents. These "intelligent" agents provide a more sophisticated capability for reasoning in and interacting with complex C^2 environments. Applications for these intelligent agents include:

- developing and assessing C^2 organizations
- training and training tools
- decision making and decision aids

Intelligent agents expand the capabilities of the scripted agents by selecting from a set of available actions at each control point rather than relying on a predefined script. An intelligent agent will select the most appropriate action based on the current and future benefit of that action. These agents will behave in a fully autonomous manner. This autonomous behavior limits direct control over the agents' actions, making experimental control significantly harder than with the scripted agents. For intelligent agents, control must be applied by limiting the available activities and carefully assigning responsibilities and objectives to the agents.

Another challenge is evaluating the benefit of each potential action so that the most appropriate action can be taken. This challenge can be considered in two parts. First, the agent must account for concurrent activity within the environment, including other decision-makers, the enemy, and natural (or random) events. Second, the agent must be capable of evaluating the future benefit of an action in the context of the current and concurrent activity. In both cases, the intelligent agents will rely in cognitive models that represent the way humans handle these problems. The representation of concurrent activity is based on a model of the agents' awareness. Future benefit is evaluated based on features of the environment.

The development of intelligent agents is further complicated by a need for real-time (dynamic) decision-making. Neuro-dynamic programming (NDP) [Bertsekas *et al*, 1996] is well suited for this type of problem. NDP uses an approximation architecture (e.g., Least Squares, Neural Network, etc.) to capture complex functions of the environment such as the characteristic features necessary for the intelligent agents to evaluate future states. The approximation architectures are trained off-line in representative situations using advanced optimization techniques. This training determines, in the form of a complex function, how to view the environment in order to make appropriate decisions. Once the approximation architectures is trained, the values necessary for decision-making can be generated in real time.

The basic methods presented in this paper for scripted agents will provide the practical and theoretical foundation for the development of intelligent agents, as well as provide an intermediate step to address current experimental needs.

5. **Conclusion**

The approach presented in this paper addresses the challenge of providing a realistic command and control environment by simulating large organizations with autonomous agents. In order to minimize the variability of individual experiments due to the autonomous agents, the agents' actions are scripted and controlled.

Scripted agents provide means of simulating a decision-making node in a large C^2 organization. This allows experiments to be performed with a small number of human participants. Experimental control is exercises via the script that provides the experiment designer with a direct mechanism to control the agents' activity.

The development of scripted agents paves the way for truly intelligent agents by establishing an interface to a simulation environment (in this case, DDD) and breaking the ground in terms of the development and application of autonomous agents in C^2 environments. The scripted agents manipulate and observe the simulation environment in a manner that is consistent with human decision-makers. The development of intelligent agents will build more sophisticated reasoning capabilities and more advanced evaluation techniques into the basic framework provided by the scripted agent.

References:

[Bertsekas *et al*, 1996] Bertsekas, D., Tsitsiklas, J.N., *Neuro-dynamic Programming*, Athena Press, Belmont MA, 1996.

[von Winterfeldt *et al*, 1986] von Winterfeldt, D., Edwards, W., *Decision Analysis and Behavioral Research*, Cambridge University Press, Cambridge, 1986.

[Kleinman *et al*, 1996] Kleinman, D.L., Young, P.W., and Higgins, G.S., "The DDD-III: A Tool for Empirical Research in Adaptive Organizations," *Proceedings of the 1996 Command & Control Research & Technology Symposium*, Monterey CA, June 1996, pp.827-836.

[Russell *et al*, 1995] Russell, S., Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River NJ, 1995.

[Rubinstien, 1998] Rubinstien, A., *Modeling Bounded Rationality*, The MIT Press, Cambridge MA, 1998.