Organizational Patterns For Data Management in Large-Scale Distributed MultiAgent Systems

William Wright, Dana Moore, Michael Thome

BBN
1300 N 17th Street
Arlington, Virginia 22209
703-284-4600
[wwright, damoore, mthome]@bbn.com

ABSTRACT

One of the important advantages of a distributed multi-agent system comes from the parallelism that can be realized by distributing data over a network of agents. A properly designed distributed agent system can scale to handle arbitrarily large data sets by adding more computers, network bandwidth, and storage. However, this scalability comes at a price. In this paper, we show how a naive approach to data management in a distributed agent system unnecessarily limits its scalability and present our approach to data management and visualization using the Cougaar agent architecture. Finally, we examine the implementation of a distributed MAS realized as three systems of increasing scale and analyze how these principles were and were not followed in an actual, functional system.

Categories and Subject Descriptors
I.2.11 [Artificial Intelligence]: Multiagent systems

General Terms
Management, Measurement, Performance, Design, Reliability, Experimentation

Keywords
Patterns, Scalability, Messaging, Interaction

1. INTRODUCTION

There are a few canonical patterns[3] of message data flow within all distributed systems[1], including distributed multiagent systems (MAS). These patterns facilitate certain types of interactions[4], but have limitations when applied to other types. Software agents are well suited to adapting the data flow pattern to the immediate requirements of the MAS. We will show how a robust agent architecture can support all of the interaction modes and discuss an implementation using the Cougaar MAS.

It is important to note that our focus is not on the efficacy of organizational design as in the field of Computational Organization Theory (COT)[2] but rather on the effects of the implementation of agent organizations on their scalability. So while we consider the key COT concepts of agents, the structure of their relationships, tasks, and stressors, our focus is on the effects of the implementation of the system on its performance as a whole. This implies that our analysis should be focused on the physical messaging between agents rather than the abstract relationships between them.

To that end, we studied a set of MAS that were developed over the course of several years by many different engineers to see how the organizational implementation choices affect the scalability characteristics of the MAS application. These MAS were not designed with this analysis in mind, so they do not reflect an attempt to implement canonical organizational structures, but rather reflect the design choices made by software engineers in the field in an effort to meet application requirements.

We divide agent group message flow patterns into three categories.

- **Hub and Spoke** – The hub and spoke pattern of data flow uses a central switch to route messages from sender to receiver. This pattern is effective in implementing publish-subscribe interactions and one-to-many communications. However, the resources of the switch limit its scalability.

- **Peer to Peer** – In this data flow pattern, entities communicate directly with one another. This has desirable properties with regard to scalability as there is no central switch bottleneck, but publish-subscribe and one-to-many flows require the sending peer to manage the list of receivers and possibly transmit multiple messages. This can be prohibitive for resource-constrained peers.

- **Hierarchical** – This pattern is a combination of the other two. In this pattern, a hierarchy of hub switches route data from its source to its destination. Many natural systems and human organizations display this kind of data flow structure.

2. CASE STUDY OF AN MAS

For the last six years, we have been developing applications using the Cougaar MAS and the patterns described above. This work represents scores of man-years of work developing, integrating, and testing distributed multiagent systems. In this section, we examine one of those applications and see how the messaging
patterns were applied and the effects of their application on the scalability of the system.

This application performs logistics planning for a set of military units for a particular contingency. This application can be scaled to solve incrementally larger problems by adding more agents to represent more military units. It was developed over several years by many people to solve a logistics planning problem, so it can be taken as a representative application in its domain and not a contrived example. The following table shows the configurations used for these experiments and the overall communications resources used.

<table>
<thead>
<tr>
<th>Agents</th>
<th>Tasks</th>
<th>Message Count</th>
<th>Byte Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>30,000</td>
<td>767</td>
<td>41,385,075</td>
</tr>
<tr>
<td>52</td>
<td>35,000</td>
<td>1,126</td>
<td>48,131,049</td>
</tr>
<tr>
<td>133</td>
<td>137,000</td>
<td>3,016</td>
<td>173,074,580</td>
</tr>
</tbody>
</table>

Table 1: Message metrics for various size agent applications

The first column of table 1 shows the size of the MAS in terms of the number of agents representing military units present in the system. The second column shows the approximate total number of tasks generated by the planning system and gives an indication of the size of the problem space. Message count is the total number of messages sent between agents in creating the plan. Byte count is the sum of the sizes of all of those messages. The largest of the three MAS is composed of approximately 2,400 unique business processes and generates over 2 million agent interactions during the construction of the logistics plan.

2.1.1 Relationship Analysis

The nature and number of relationships in an agent system is an indicator of its scalability. The following figures show relationship graphs for the three agent systems under consideration. The vertices represent agents and the edges represent communication paths between them that are used in the application.

In Figure 1 (right), clusters of related agents are starting to become apparent in the 52-agent system. While, the agent in the center is still related to all other agents, subgroups are starting to become apparent in the upper left and lower right corners.

By the time the MAS reaches 133 agents, clusters of agents are readily apparent. The left side of Figure 2 (left) contains agents that (with the exception of the center hub agent) relate primarily among themselves. The upper right and lower right areas contain other agent clusters that are representative of the planning domain modeled by the MAS application. This becomes clearer if we remove those relationships that are lightly used (that is, those that carry less than 10% of the traffic carried on the average link) as shown in the right side of Figure 2. Figure 2 (right) shows that the agents tend to cluster into local hierarchies and peer groups as can be seen especially well in the lower right corner of the figure.

3. SUMMARY

We have seen through examination of a representative application how communications patterns are used in practice. Analysis of the application under consideration displayed tendencies toward scalability limitations that were borne out by experience.

4. ACKNOWLEDGEMENTS

This work is sponsored by the US Defense Advanced Research Projects Agency (DARPA) and is managed under DARPA’s Joint Logistics Technology Office (JLTO).

5. REFERENCES