Suitability of Agent Technology for Military Command and Control in the Future Combat System Environment

Dr. Thomas Potok and Dr. Andy Loebl
Computational Sciences and Engineering Division
Oak Ridge National Laboratory

Laurence Phillips and Robert Pollock
Advanced Information and Control Systems
Sandia National Laboratories

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY
The 2020 Army Challenge

Unit spotted
Missile Fired
Sarin Detected
Agent Vision

- Multiple T-72s tanks; Possible sarin filled 122M rockets (Picture)
- Apache Longbow can engage in under 15 minutes (COP)

Apache Longbow destroy the intruder
Forward forces to MOPP4
Issues

- Can current software technology solve the FCS Command and Control problem?
- Are software agents a better approach to solving this problem that traditional technology?
Qualitative Approach

- Derive the needed software capabilities from the TRADOC FCS C2 requirements
- Review these capabilities against the current software technology to determine the limitations of the current technology
- Review the limitations of current technology against the capabilities agents technology
FCS Requirements

- Common Operational Picture
- Mobile Command
- Mission-Centric IS
- Decision Support/Planning
- 3D Visualizations
- Continuous Mission Planning
- Synchronized C2
# Functional and Software Requirements

<table>
<thead>
<tr>
<th>TRADOC Requirements</th>
<th>Distributed Computing</th>
<th>Fault Tolerance</th>
<th>Security</th>
<th>Mobile Code</th>
<th>Information Fusion</th>
<th>Information Analysis Summary</th>
<th>Decision Support</th>
<th>Software Productivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Operational Picture</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Mobile Command</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Mission-Centric IS</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Decision Support/Planning</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3D Visualizations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Continuous Mission Planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Synchronized C^2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1** A mapping of TRADOC FCS functional requirements to expected software requirements.

*Oak Ridge National Laboratory  Applied Software Engineering Research Group*
Needed Software Capabilities

- **Distributed computing** over an unreliable, ad hoc, dynamic physical network
- **Fault tolerance** over a system in which, at any given time, it is unclear what nodes are available within the network
- **Network security and accessibility.** Warfighters will need immediate access to the network, but adversaries need to be prevented from accessing or corrupting it.
- **Data fusion.** Data from a wide range of systems and sensors will need to be correctly related
- **Information analysis and summary** of enormous amounts of data from the C2 network on the basis of user needs
- **Decision support.** A network capable of supporting C2 decision making
- **Software development improvements** to reduce the complexity and risk in creating the proposed system
## Current Software Limitations

<table>
<thead>
<tr>
<th>Software Requirements</th>
<th>Distributed Computing</th>
<th>Fault Tolerance</th>
<th>Mobile Code</th>
<th>Security</th>
<th>Information Fusion</th>
<th>Information Analysis Summary</th>
<th>Decision Support</th>
<th>Software Productivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Higher-level Interfaces</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asynchronous Interaction</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sporadic Network Support</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peer-to-peer Models</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Productivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

**Figure 3** A mapping of the software requirements to the limitations of the current software technology
Limitations

- Providing higher-level interfaces to distributed objects.
- Allowing asynchronous object interaction
- Providing message support for sporadic network connections
- Providing secure object communication and information system operation
- Providing support for richer peer-to-peer programming models
- Increasing software development productivity
Agent Definition

- Agents are typically described as possessing human characteristics,
  - autonomous, adaptable, social, knowledgeable, mobile, and reactive, ...
- For the purposes of this study,
  - we are more interested in the computer science novelties of the technology
  - focus strongly on the comparative benefits of agent technology
Representative Agent Architectures

- Sycara et al. [i] proposes planning, communication and coordination, scheduling, and execution monitoring of agent activities.
  - Agents access shared information through a coordination model that can be domain specific or domain independent.
- Griss et al. [ii] who provide an architecture for locating and communicating with moving and unconnected agents, and for gathering information about groups of agents.
  - This architecture provides services that include support for mobility, security, management, persistence, and naming of agents.

Agent Novelty

- Communication and control aspects of agent systems
  - Peer-to-peer topology
  - Agent coordination models that provide encapsulated and asynchronous messaging with the use of blackboards, and tuple space models and associated pattern-matching
  - High-level messages are typically written in an agent control language (ACL) such as KQML or the FIPA ACL. These languages provide a structured means of exchanging information and knowledge among agents.
Why Agents?

Traditional Software

- Client-server
- Low-level messages
- Synchronous
- Can not do the job!

Agent breakthroughs
- Peer-to-peer topology
- Blackboard coordination model
- Encapsulated messaging
- High-level message protocols

Intelligent Agents
- Function(Parameters)
- Return(Parameters)

Intelligent Agents
- Latest on bin laden?
  - He dead.
  - In Pakistan
  - Unknown

Intelligent Agents

Oak Ridge National Laboratory
U.S. Department of Energy

Applied Software Engineering Research Group
Agents against the limitations

- Providing higher-level interfaces to distributed objects.
  - Agents support
  - No universally accepted ACL standard
- Allowing asynchronous object interaction
  - Agents support
  - Performance of large blackboard systems unclear
- Providing message support for sporadic network connections
  - Agents support
  - Need store and forward, and rollback capabilities
- Providing secure object communication and information system operation
  - Agents support
  - How easily agents can be “turned” is an issue
- Providing support for richer peer-to-peer programming models
  - Agent Support
  - Topologies much be carefully built to ensure performance
- Increasing software development productivity
  - There may be improvements through reuse, but no evidence to support this
So should agents be used for C2?

- **Good news:**
  - Agents appear to have several technological advantages over traditional programming, mainly in communications with other agents.
  - This clearly would benefit FCS, or any large distributed software project.

- **Bad news:**
  - Traditional software has major limitations in an FCS environment, and may not be suitable.
  - Agent technology may be suitable, but there are no large reference systems to validate this.
Recommendations

- Large-scale experimentation is needed to validate an agent architecture for FCS C2.
- Main areas:
  - Scalability
  - Survivability
  - Security
Conclusion

- Traditional technology need significant enhancement to meet the needs of FCS
- Agent technology is the best suited technology for FCS, but need to be validated though experimentation
Contact Information

- Contact Information

Thomas E. Potok, Ph.D.

Potokte@ornl.gov

865-574-0834