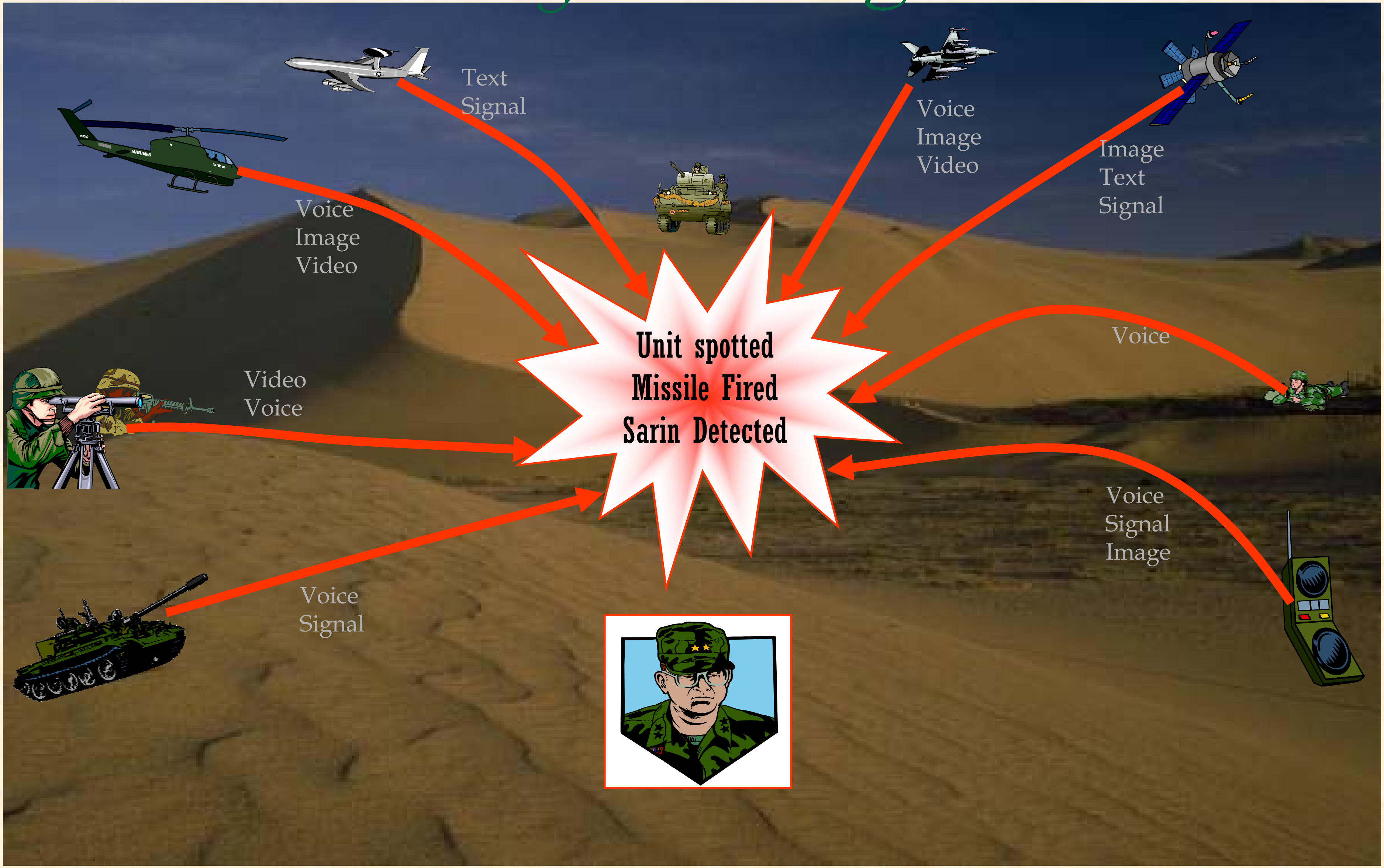
Suitability of Agent Technology for Military Command and Control in the Future Combat System Environment

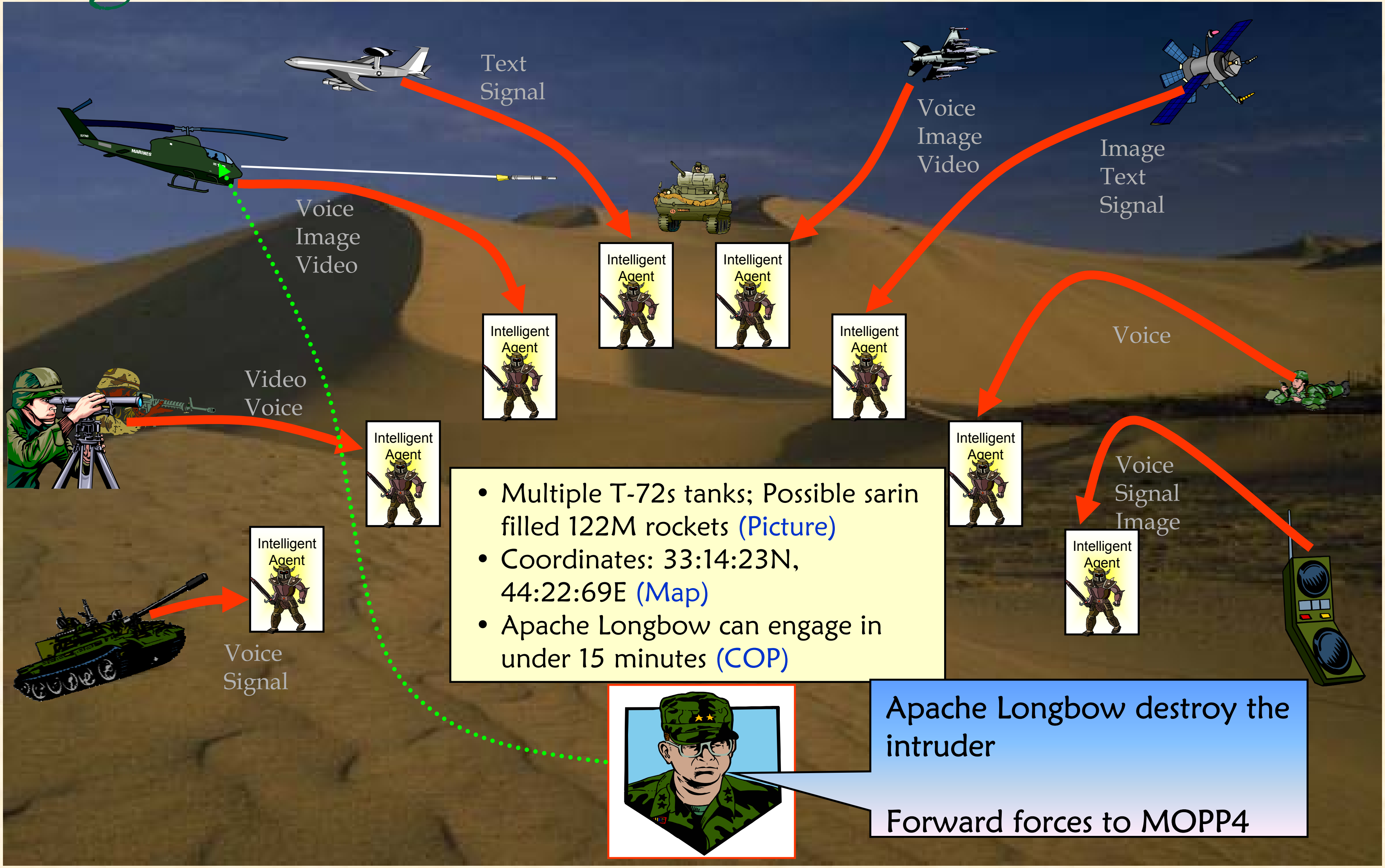
Dr. Thomas Potok and Dr. Andy Loebl
Computational Sciences and Engineering Division
Oak Ridge National Laboratory

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

The 2020 Army Challenge



Agent Vision



- Multiple T-72s tanks; Possible sarin filled 122M rockets (Picture)
- Coordinates: 33:14:23N, 44:22:69E (Map)
- 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

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

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

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

Software Requirements Software Limitations	Distributed Computing	Fault Tolerance	Mobile Code	Security	Information Fusion	Information Analysis Summary	Decision Support	Software Productivity
Higher-level Interfaces	X			X				
Asynchronous Interaction	X							
Sporadic Network Support	X	X	X					
Security			X	X				
Peer-to-peer Models	X	X						
Software Productivity								X

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.

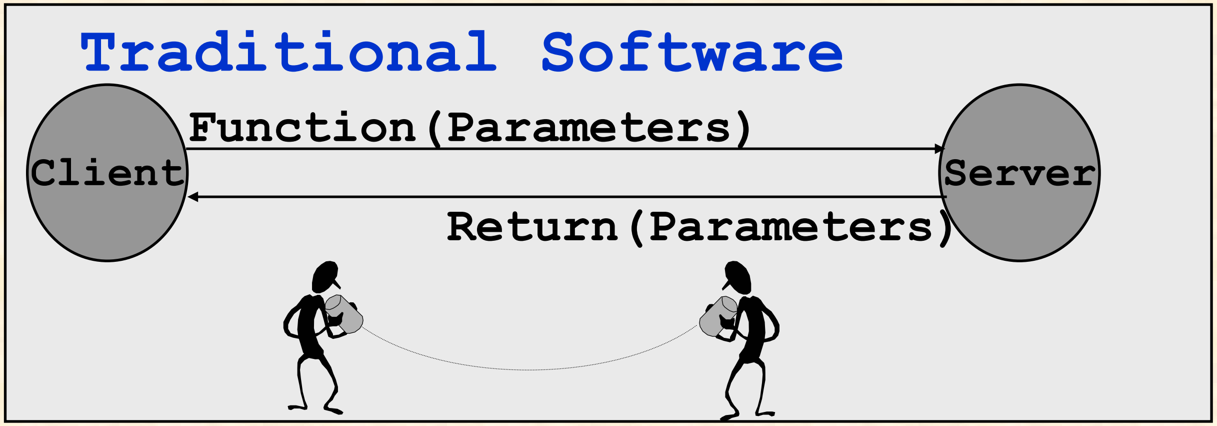
[i] K. Sycara, A. Pannu, M. Williamson, and D. Zeng, "Distributed Intelligent Agents," *IEEE Expert* 11, no. 6 (Dec. 1996): 36-46

[ii] M. Griss and G. Pour, "Accelerating Development with Agent Components," *IEEE Computer* 34 no. 5 (May 2001): 37-43.

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?



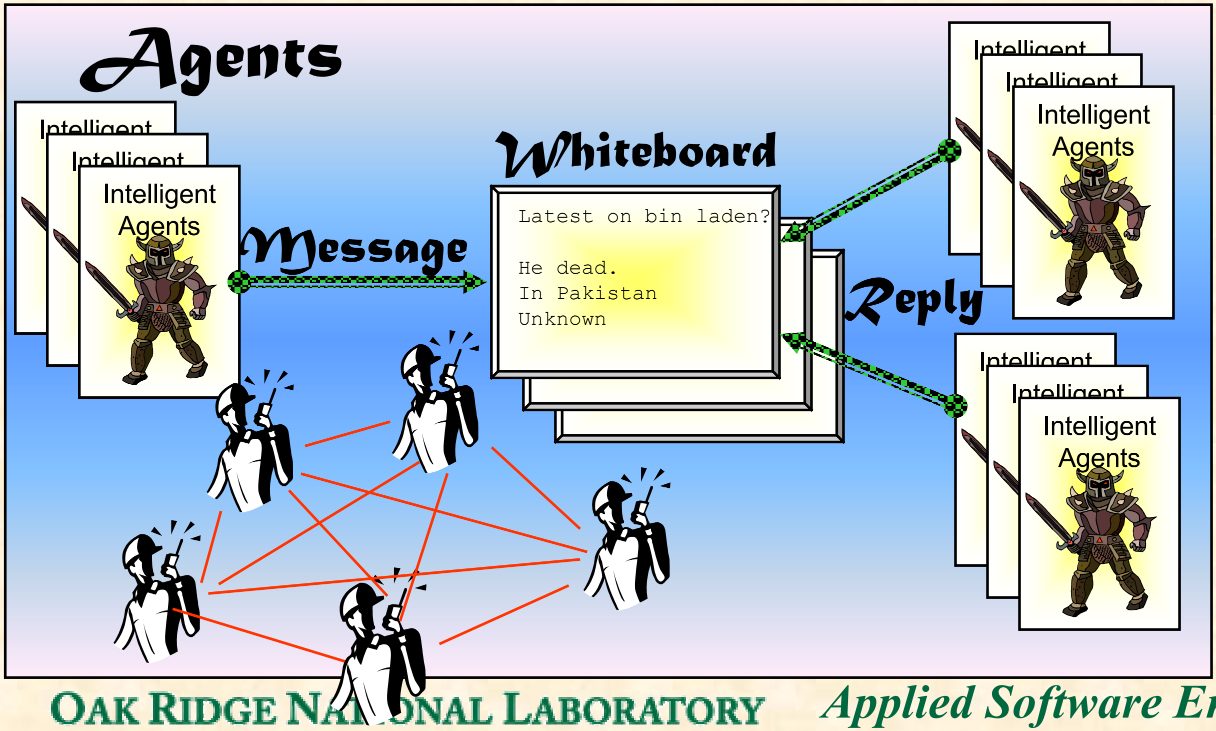
Traditional Software

Client

Function(Parameters)

Server

Return(Parameters)



Agents

Intelligent Agents

Message

Whiteboard

Latest on bin laden?

He dead.
In Pakistan
Unknown

Reply

Intelligent Agents

Intelligent Agents

- Traditional
  - 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

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, main 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 my 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
  - Survivatibility
  - 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