AFRL-IF-RS-TR-2002-148
Final Technical Report
July 2002

MULTI-AGENT COMMON OPERATING ENVIRONMENT (MACOE)

Lockheed Martin Advanced Technology Laboratory

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J377

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

AFRL-IF-RS-TR-2002-148 has been reviewed and is approved for publication.

APPROVED:
MICHAEL D. SCHUYLER, Lt., USAF
Project Engineer

FOR THE DIRECTOR:
MICHAEL L. TALBERT, Technical Advisor
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JULY 2002	Final Jun 98 – Dec 01

4. TITLE AND SUBTITLE
MULTI-AGENT COMMON OPERATING ENVIRONMENT (MACOE)

5. FUNDING NUMBERS
C - F30602-98-C-0162
PE - 63760E
PR - AGEN
TA - T0
WU - 17

6. AUTHOR(S)
Ken Whitebread, Martin Hofmann, and Lori Pridmore

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lockheed Martin Advanced Technology Laboratory
1 Federal Street
Camden New Jersey 08102

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFTB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2002-148

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Michael D. Schuyler, Lt. USAF/IFTB/(315) 330-3910/ Michael.Schuyler@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
The Multi-Agent Common Operating Environment is an effort funded under the CoABS program executed by prime contractor Lockheed Martin Advanced Technology Labs (LM ATL). ATL's mission under the CoABS MACOE program is to advance the application and transition of agent based systems into the Military through the definition, test and evaluation of technology requirements, prototypes, and applications of human-agent teams and multi-agent teams.

14. SUBJECT TERMS
Intelligent Agents, Agent Control, Time Critical Targeting

15. NUMBER OF PAGES
27

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

Executive Summary

The objective of the DARPA Control of Agent-Based Systems (CoABS) program is to develop and evaluate several control strategies that will allow military commanders and planners to automate relevant command and control functions such as information gathering and filtering, mission planning and execution monitoring, and information system protection through the use of agent-based systems. Through the effective control of agent systems, the intelligent agents will work in harmony to strengthen significantly military capability by reducing planning time, automating and protecting Command and Control (C2) functions, and enhancing decision-making. The CoABS program will develop and evaluate a wide variety of alternative agent control and coordination strategies to determine the most effective strategies for achieving the benefits of agent-based systems, while assuring that self-organizing agent systems will maintain acceptable performance and security protections.

The Multi-Agent Common Operating Environment is an effort funded under the CoABS program executed by prime contractor Lockheed Martin Advanced Technology Labs (LM ATL). ATL's mission under the CoABS MACOE program is to advance the application and transition of agent-based systems into the Military through the definition, test and evaluation of technology requirements, prototypes, and applications of human-agent teams and multi-agent teams.

Under the CoABS program, ATL worked cooperatively with the Naval Warfare Development Command (NWDC) to test and evaluate the application of agent technology to key Navy challenges. Technology was applied in the area of the time-critical strike domain and tested in multiple fleet-battle experiments. The CAST Technology framework was developed and refined through a series of Fleet Battle Experiments developed in coordination with NWDC efforts. The CAST technology provides a framework to compose multi-agent systems from cooperating, heterogeneous agents and legacy systems. Time Critical Strike applications were developed utilizing the CAST framework and were tested and evaluated in five Fleet Battle Experiments. In addition, the CAST framework was utilized to support several experiment and transition paths including the CoABS Mobility Technology Integration Experiments (TIE), Coalition experiments, and the JIATF-East Drug Interdiction and the NWDC Anti-Terrorism Force Protection Command & Control domain applications.

CAST has shown the utility of agent-based decision support in domains where teams of human operators and decision makers work on a common problem, such as Time-Critical Targeting. CAST ensures that all users see the same information and that users cannot duplicate each others actions with respect to agent's tasking. ATL will exploit the work accomplished under the MACOE program to develop an Intelligent Interoperable Agent Toolkit (I2AT) that supports the wide-scale development of agent-based system within the software engineering community.

1 Background

1.1 DARPA Control of Agent-Based Systems (CoABS) Program

Today's software has increased the user's ability to access and manage information, edit, present ideas, and generally control the work environment. However, this high level of control comes at the expense of increased work for the user. Furthermore, current systems lack the ability to get contextual information or use it to automate filtering as well as lack of protection from self-destructive behaviors or attack from outside sources. To solve these problems and make effective use of our growing information infrastructure, we need a fundamentally new kind of software technology for automating our processes. Requirement and the need exist for customized software technology that can be rapidly developed at low cost and execute on readily available hardware without overloading conventional processors. With this technology, the military will be able to adapt decision-making processes quickly and cheaply to automate access to information, generate alternative courses of action, communicate ideas, and protect the information infrastructure. To this end, the industry and government are leveraging previous research in distributed artificial intelligence to develop intelligent agents that are software modules designed to provide these capabilities.

Even though the agent technology promises to lead to a revolutionary new model of computing beyond client-server architecture, significant research and development remains to be accomplished before true benefits of agent-based systems can be realized.

For example, agents or agent systems produced by different developers cannot cooperate in any meaningful way. Cooperation among agents is critical to building powerful applications to support military capability because without cooperation, each new task must be handled by a monolithic agent designed for it. Control strategies are needed to build small teams of agents that can cooperate in a robust and flexible manner, as well as a very large number of agents that exhibit macro scale behavior without attending to the detailed behavior of individual agents.

Furthermore, there are no sufficient algorithms, policies, or mechanisms that prevent a large heterogeneous set of agents from exhibiting dangerous or chaotic behavior on a network. This lack of control can lead to clogged networks, wasted resources, poor performance, system shutdowns, and security vulnerabilities.

What is needed and what CoABS proposes to develop are technologies for the control of multi-agent systems with predictable behavior for automating military command and control in a cost-effective manner. If successful, the systems of cooperating agents and agent ensembles are expected to dramatically reduce the information systems workload for the entire spectrum of military forces from the national command authority down to the small-unit level as well as provide a framework for resource management in a dynamic hostile or unpredictable environment in which software systems are adaptable, self-configuring, self-healing and evolvable.

Specifically, CoABS proposes to develop:

1. A simple agent programming methodology supported by sophisticated component libraries so that we can automate complex functions cheaply and easily with agents assembled from powerful pieces.

2. Compatible agent behavior models.

3. Interoperable agent communication languages.

4. Advanced, fully protective agent services for protecting both agents and hosts/current servers/existing data sources. (Examples of partial solutions: Agent-Tcl and CMU's yellow pages)

5. Simple methods of understanding agent behavior. (e.g. visual programming language, graphic simulation, and rationale trace)

The Control of Agent-Based Systems program is organized into four areas:

Cooperative Control Strategies
Development and demonstration of alternative agent control strategies for coordinating, controlling, and managing agents' collections, ranging from simple tasks involving the cooperation of small agent teams to highly complex interactions involving thousands or millions of agents. Research topics include: models of collaborative behavior, the role of competition, policy and mechanisms for competition and cooperation, semantic representation and translation methods, and agent facilitation, brokering, and mediation.

Reliability Assurance Methods
Development and demonstration of methods of resource allocation and control, security mechanisms, appropriate methods of agent creation and deletion, distribution of agents on the network, agent system behavior, user interfaces to identify agent behavior, and trade-offs between such control mechanisms and collaboration

Computer Systems Architectures
Development and demonstration of computer system architectures appropriate for both multi-agent systems and legacy software applications in current architectures. Areas of research are architecture design implications and trade-offs, communication protocols, standards for agent interoperability, system integration, and application programmer interfaces.

Related Technologies
Development of cost-effective agent development languages, tools and environments, testing and demonstration environments, evaluation methods, and component capabilities. Component capabilities include distributed artificial-intelligence-based techniques such as planning, scheduling, execution monitoring, machine learning, user interfaces, knowledge-sharing, and acting.

1.2 MACOE Team

The Multi-Agent Common Operating Environment (**MACOE**) program is an effort funded under the CoABS program. The prime contractor, and primary contributor, is Lockheed Martin Advanced Technology Laboratories. The Lockheed Martin Advanced Technology Laboratories (LM ATL) is an advanced-computing asset of Lockheed Martin Corporation—a global enterprise with principal business areas in aeronautics, space, systems integration, and technology services. The Advanced Technology Laboratories' mission is to enhance the LM Corporation's competitive edge, developing and applying computing innovations in artificial intelligence, distributed processing, and embedded processing. Key to leading-edge innovation is an aggressive internal research and development program and contractual relationships with the Defense Advanced Research Projects Agency (DARPA), military laboratories, and other government agencies.

In earlier DARPA-funded agent work, ATL developed the agent-based Domain Adaptive Information System (DAIS), that automated and accelerated critical information flow through the echelons of the 201st Military Intelligence Brigade at Ft. Lewis, WA. DAIS was fielded in thirteen field-training exercises that let users push and pull intelligence records at any network node. The results from these experiments proved that the automation and robust execution of our agents significantly reduced the workload of system operators and gave decision makers more time to understand and control their environment. The greatest challenges were interoperability with legacy stove-piped systems and the control of a potentially large number of agents. Another key result of the DAIS project was the development of the Extensible Mobile Agent Architecture (EMAA), a flexible platform for the development of mobile agent systems. EMMA provided key leverage for the MACOE program effort.

Under the CoABS program, ATL continued to advance the application and transition of agent-based systems into the Military through the definition, test and evaluation of technology requirements, prototypes, and applications of human-agent teams and multi-agent teams. To further support application test and evaluation and operational transition, ATL teamed with Logicon and Litton PRC. These key subcontract relationships provided focused support on specific military applications and transitions including:

- **Logicon** –Supported insertion of our JIATF-E Case Agent (JECA) capability into Logicon's WebTAS system. Logicon is performing the integration work and will deploy JECA with a new release of WebTAS at JIATF-E and additional future WebTAS installations.

- **Litton PRC** – Supported development of the Modernized Integrated Database (MIDB) wrapper. With this wrapper, we integrate MIDB with the CoABS Grid and CAST.

2 Project Objective

ATL's MACOE objectives are threefold:

- Provide a framework to compose multi-agent systems from cooperating, heterogeneous agents and legacy systems.
- Experiment with these systems in military exercises.

- Transition multi-agent decision support technology into wide-spread use in the military services.

To this end, we envisioned MACOE, the Multi-Agent Common Operating Environment, where heterogeneous systems interact through adapters or "drivers" that hide and extend native system interfaces. We aligned this vision with the CoABS Grid vision and embarked on a development path that coupled technology development with experimentation in military contexts.

Our experimental objective is to develop and field multi-agent Grid-aware systems, connect them to legacy C4ISR systems, such as the Global Command and Control System – Maritime (GCCS-M), and demonstrate automation and decision support capabilities. We are continuing development on our Cooperating Agents for Specific Tasks (CAST) decision support framework and its application in USN Fleet Battle Experiments (FBEs). During experiment and exercise participation, our goal was to learn where agent systems can provide the most benefits compared to traditional systems architectures, and to capture requirements for interoperability capabilities.

Our transition objective is to define and transition an agent-based computing life-cycle model and a toolkit that supports compliant multi-agent system development. A new life-cycle model will make it possible to reap the benefits of the CoABS interoperability technology. It will allow much faster system maintenance, enhancements, and specialization, because tools can be built that simplify these activities to the point where subject matter experts and IT support staff can perform most or all of them.

3 MACOE

3.1 Operational Community Partners

The increasingly sophisticated asymmetric tactics of overmatched opponents have prompted research and experimentation into processes that accelerate and improve the U.S. military response. The USN Navy Warfare Development Command[1] (NWDC) in Newport, RI, addresses warfare innovation in terms of developing new doctrine and concepts, by war gaming and experimentation. The Maritime Battle Center department of NWDC coordinates the execution of Fleet Battle Experiments with the numbered fleets where operators and NWDC personnel jointly exercise these innovative warfare concepts. FBE concepts and initiatives have included Network-Centric Warfare, Theater Ballistic Missile Defense, and Time-Critical Strike.

Under the CoABS program, ATL worked cooperatively with NWDC to test and evaluated the application of agent technology to key Navy challenges. Technology was applied in the area of the time-critical strike domain and tested in multiple fleet-battle experiments.

[1] http://www.nwdc.navy.mil

3.2 Challenge Problem (Time Critical Targeting)

The Time-Critical Strike (TCS) concept development aims to shorten the time to detect the fleeting threat, to decide if and how to engage it, to engage with the chosen weapon, and to assess the damage inflicted. Decision makers and supporting operators are challenged to collect and interpret the available data and imagery, to analyze and choose courses of action, to coordinate among the multiple operators on distributed platforms, i.e. ships, naval aircraft, and Marine land forces, and to monitor for unexpected changes in the situation. Fleet Battle Experiment architects deploy advanced systems for target detection, mensuration, weapon-target pairing, and fires coordination. However, each of these systems requires human operators and the TCS process results from human operators coordinating among each other with simple tools, such as Internet Relay Chat (IRC). The opportunity exists to accelerate the TCS process through greater levels of automation.

FBE initiative leads team with industry and design novel system architectures that specifically support the topics of experimentation. This architecture is composed of systems ranging from experimental systems to systems near transition, such as GISRS (Global Intelligence, Surveillance, and Reconnaissance System)[2] to systems of record in the Global Command and Control System – Maritime (GCCS-M), such as the Modernized Integrated Database (MIDB). Interconnecting these diverse constituent systems for the experiment requires significant ingenuity and resources. There is a need for a more flexible approach to system connectivity and interoperability. With the hypothesis that intelligent software agent technology will to lead to a revolutionary new model of computing beyond client-server architecture, the DARPA Control of Agent Based Systems (CoABS) program[3] is investigating control and coordination of distributed, heterogeneous agents and non-agent services. Its goal is to develop the technologies that make possible systems of cooperating agents and agent ensembles that dramatically reduce the information systems workload for the entire spectrum of military forces.

4 Technology overview

4.1.1 LM ATL EMMA Agent Platform

The Extensible Mobile Agent Architecture (EMAA) was developed by the Lockheed Martin Advanced Technology Laboratories. EMAA provides a rich component framework for developing or integrating distributed systems using autonomous mobile agents. The central component for EMAA is the agent Dock, which acts as a daemon process within a Java Virtual Machine (JVM) and supplies the hosting infrastructure and foundation for software agents and services. Mechanisms for reliable agent migration and authentication amongst Docks, as well as service lookup and discovery, are built into the framework.

---

[2] GISRS has since transitioned into the GCCS-M as GISR-C (GCCS-M Intelligence Surveillance and Reconnaissance Capability).

[3] http://www.darpa.mil/ito/research/coabs/index.html

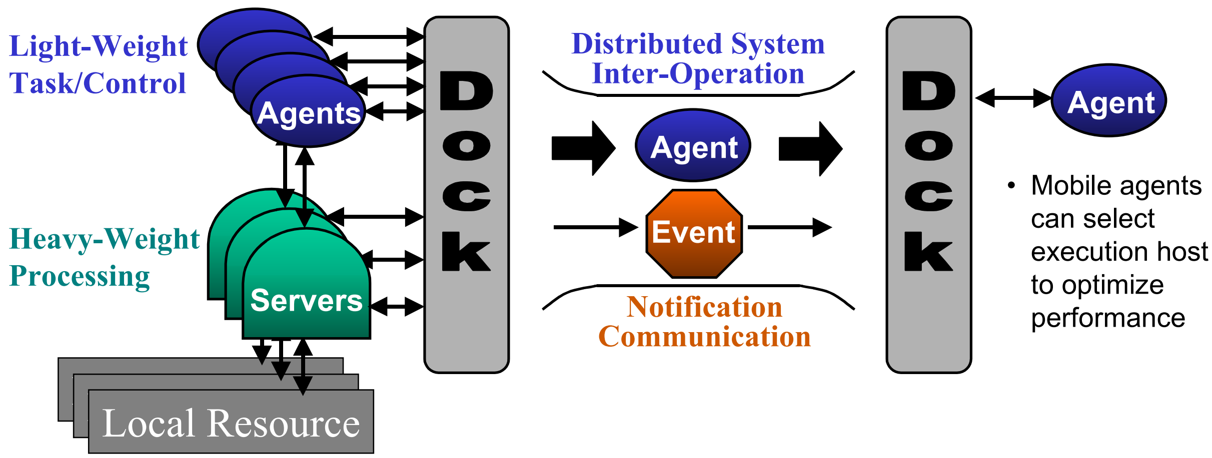


Figure 4-1-1 : EMAA Architecture

A mobile intelligent agent's primary responsibility is to achieve the processing objective represented by an itinerary and assigned to it from a user or system. Mobility is an added capability, enabling the agent to relocate its processing to at defined points within its itinerary. Because an agent receives an execution thread exclusively from the Dock, a mobile agent may only migrate to hosts that have an agent Dock running within a JVM. Agents often represent the business logic of an application, which may need to change based upon current circumstances. Services are components loaded into the Dock for use by agents. Typically, a service may provide a standardized interface to a resource, such as a database, or a computational engine. Reusable agent tasks representing the most common use cases of the service included. The intent of services is to encourage reusability and reduce the size of the agent when it must migrate.

EMAA has currently been incorporated as agent architecture for over 15 ATL contract efforts. EMAA has also been licensed for use with universities, Lockheed Martin departments, and LM ATL business partners.

4.1.2 CoABS Grid

The CoABS Grid, developed by Global InfoTek, allows EMAA agents to cooperate with other CoABS agent platforms. ATL's MACOE work has produced several agent system prototypes that leverage the interoperability features of the Grid. EMAA agents interoperated with D'Agents from Dartmouth College in NWDC Fleet Battle Experiment - Foxtrot and in the Mobility Technology Integration Experiment (TIE), with Nomads and KAoS from the University of West Florida in the Mobility TIE, with BBN's OMAR in a sentinel agent system for Air Mobility Command.

ATL has also used the Grid to connect its agent prototypes to legacy systems and databases. Several re-usable Grid wrappers resulted from these efforts, including a wrapper for the Modernized Integrated Database (MIDB), a component of GCCS-M, the Air Operations Database (AODB), a component of TBMCS, the Image Product Library (IPL), etc.

ATL contributed to the early design of the Grid and remains one of the major application developers using the CoABS Grid.

4.1.3 CAST Agent-Based Decision Support Framework

CAST is our decision support agent framework based on EMAA and the CoABS Grid. CAST, Figure 4-1-3-1, provides agent behavior and cooperation patterns generally useful for decision support in C4ISR networks. CAST agents consist of Java bean-like tasks that developers compose into workflow patterns and configure for the specific installation.

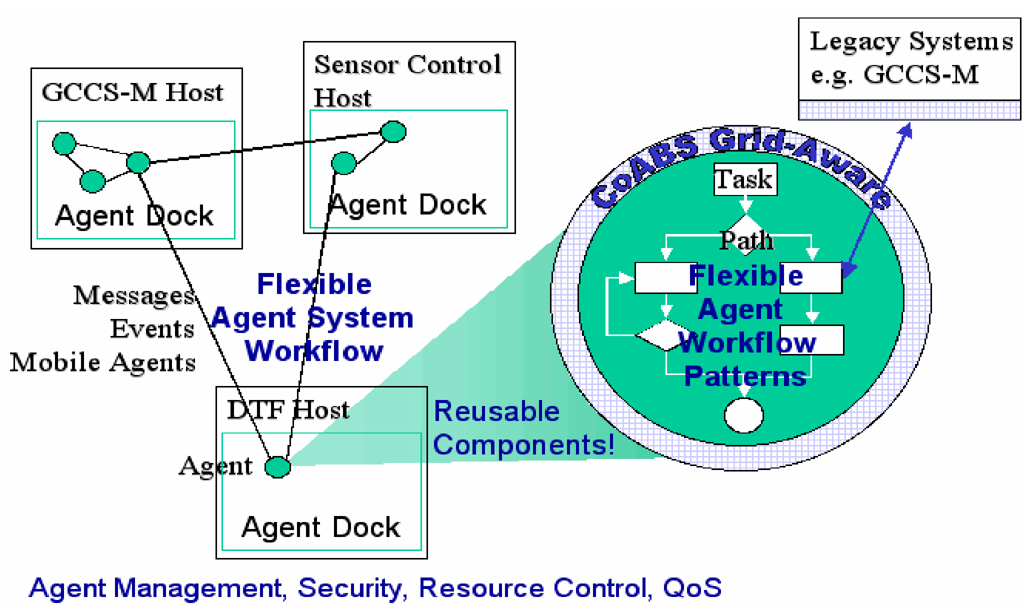


Figure 4-1-3-1. The CAST multi-agent framework architecture invites reuse of agent tasks, agents, and workflow patterns. It employs the CoABS Grid to ease system integration and enable interoperability.

CAST supports patterns of cooperation among specialized agents. In FBE-E we demonstrated a Theater Ballistic Missile (TBM) launch detection system composed of four diverse agent types that cooperated through a blackboard. The four agent specialties were data source monitoring, data correlation, distributed data search, and user alerting.

We are increasing the flexibility of CAST by delaying the configuration and even some composition steps until the system is installed. We have successfully used the CAST framework to tailor CAST applications for FBE-E through FBE-I, for the Joint Grid-Based Integrated Targeting (JGIT) demonstration, and for the 6th Fleet Distributed TCS Limited Objective Experiment. Figure 4-1-3-2 shows the Grid-supported integration of CAST into the FBE-I system architecture.

In each of these deployments, we have benefited from the interoperability provided by the CoABS Grid. We have developed Grid-based agent-service communication design patterns and stubs that normalize our solutions and accelerate our development. Using these stubs, we rapidly integrate CAST into the varying C4ISR system environments.

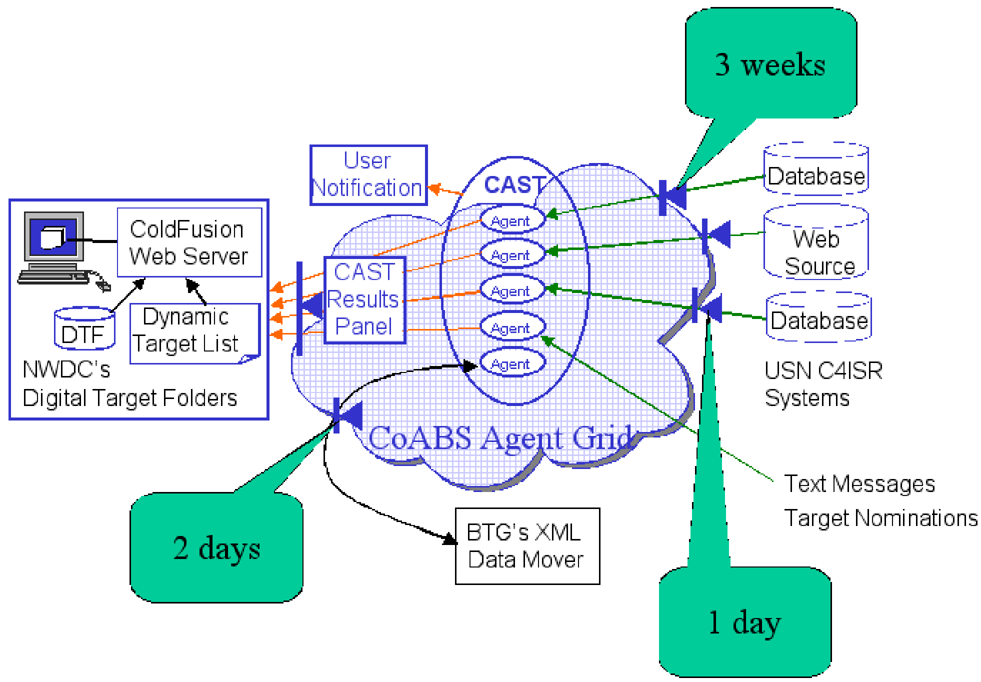


Figure 4-1-3-2. The CAST architecture for FBE-I highlights the benefits of Grid-supported integration. The green callout boxes show how quickly systems can be integrated via the Grid. The 3 weeks required to integrate the Modernized Integrated Database (MIDB) includes 2.5 weeks of custom development to wrap the complex MIDB data model. It took only 2 days to integrate CAST with a data mover system developed by another contractor, BTG, through which agents prompt the data mover to update specific target records. A Grid connection to a simple database reduces to a few configuration steps that we routinely complete, including testing, in a single day.

4.1.4 Operator/Agent Teaming

CAST has shown the utility of agent-based decision support in domains where teams of human operators and decision makers work on a common problem, such as Time-Critical Targeting. CAST ensures that all users see the same information and that users cannot duplicate each others actions with respect to agents taskings. For example, check boxes are greyed out as soon as one user performs a global action, such as a database insert. Today CAST presents a uniform interface to these operators and agents cooperate with each other while servicing user requests. Through our experimentation, we have learned some of the limitations of our approach. Operational users have identified the need to personalize information presentation and to monitor the workflow among users. Analysis of agent behavior points out the need for better self-coordination among task agents to gain efficiency while servicing overlapping information requests from multiple users.

4.2 Application Prototype/Experimentation

4.2.1 A Series of CAST Grid-Enabled Multi-Agent Decision Support Systems

From FBE-H (2000) onward and culminating in FBE-I (June 2001), all interoperability between agents and C4ISR systems is provided by the CoABS Grid. In FBE-I, CAST interacted with seven C4ISR systems relevant to TCS operations. CAST proved technically sound and operationally capable. After FBE-H, NWDC stated that LM ATL's CAST agent system "showed promise, replacing redundant manual operations" and, after FBE-I, that "CAST … is a situational awareness multiplier."

The latest draft of NWDC's assessment of CAST in FBE-I starts with this paragraph:

> **1.5.1.2. Assessment.** It was apparent that operators that were trained to use the Smart agents thought that the technology was extremely valuable. The main reason was that it allowed them to rapidly assemble many pieces of information about a threat and associated processing in one place. There were a number of knowledge discovery functions demonstrated in this initiative. CAST kept a complete log of agent activities.

Our CAST configurable agent-based decision support framework accelerates our development of specific implementations for similar but different exercise requirements. The CAST framework provides a development model, a decision support architecture, and a set of configurable components. With CAST and the CoABS Grid, we are one step closer to our vision of composing multi-agent applications instead of programming them, and of composing specific agent behaviors instead of coding them

4.2.2 Heterogeneous Agent Mobility Experiments

As a member of the Mobility Technology Integration Experiment (TIE) team, LM ATL contributed to the design and implementation of the Grid Mobile Agent Service (GMAS) and in the demonstration that showed agents migrating among heterogeneous mobile agent platforms via GMAS. Our GMAS agents migrated between ATL's EMAA host, a Dartmouth College's D'Agents host, and a University of West Florida's Nomads host.

Also on the Mobility TIE, LM ATL collected experimental data on the performance of EMAA and contributed to the publications that use the experimental results to verify the conditions where mobile agents perform better than traditional client-server solutions. The experiments proved that LM ATL's EMAA implementation is highly efficient, especially in low bandwidth "last mile" networks.

4.2.3 Agents for Coalition Operations Experiments

LM ATL provided a team of monitoring agents to the multi-agent coalition experiments (CoAX). We were among the first agent developers to contribute agent components to this experiment, because of the maturity of our technology.

4.3 Experimentation

4.3.1 Fleet-Battle Experiments

Our participation in the USN Fleet Battle Experiments (FBE) provided the leading edge for transition of CoABS technology. FBE decision support needs as well as the need for rapid, impromptu configuration of heterogeneous systems turned out to be a perfect match for CoABS agent interoperability technologies. CAST proved the benefits of agent functionality to operational users, and our use of the CoABS Grid demonstrated a more dynamic and scalable approach to systems integration. Data collected during the FBEs provides qualitative evidence that agents do not interfere with manual operations, a major concern of the operational community.

During FBE-H, for example, CAST operated on board the USS Mt. Whitney, the 2nd Fleet command ship. Figure 4-3-1 shows Navy Warfare Development Command's experiment hypothesis and our complementary agent technology hypothesis.

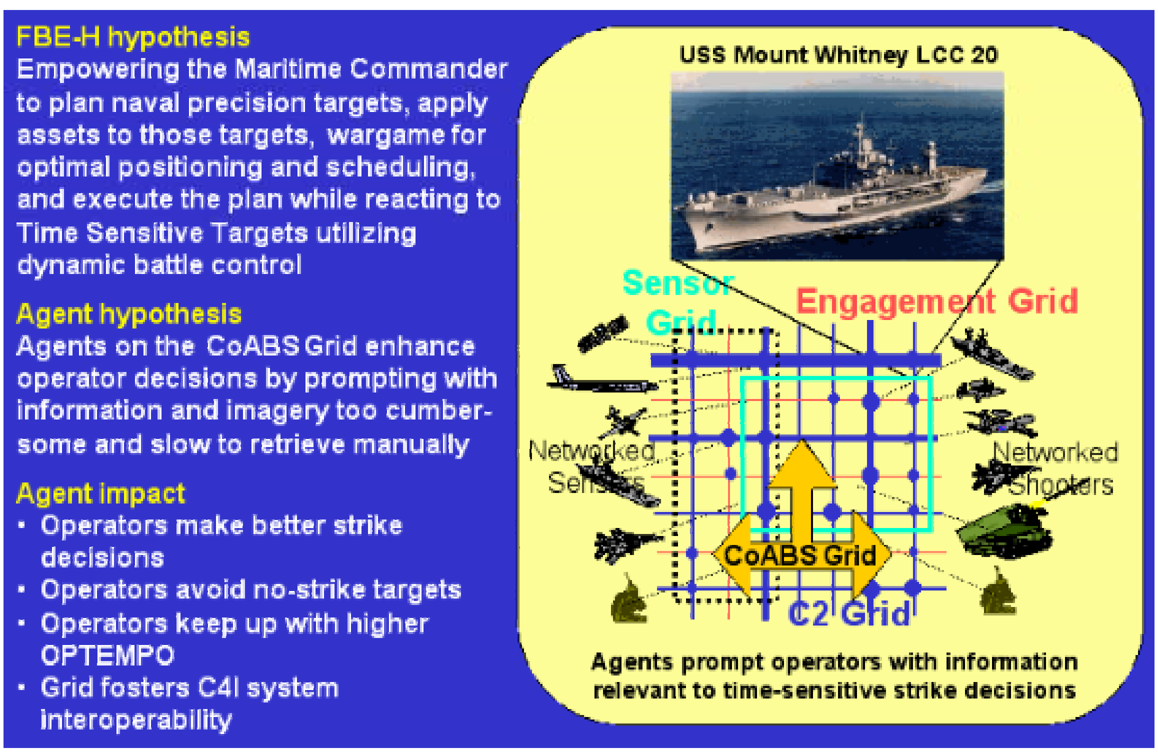


Figure 4-3-1. CAST demonstrated the benefits of intelligent, autonomous agent assistance and interoperability in FBE-H. CAST

reduced operator workload even though experiment constraints limited CAST functionality.
The success of CAST in the FBE series has greatly furthered the interest of the Naval Warfare Development Command (NWDC) and SPAWAR in the results of the CoABS program, including the CoABS Grid and the CAST agent application.

4.3.2 Joint Experiments

At the request of General Myers, Chairman, Joint Chiefs of Staff, we have expanded the US Navy oriented CAST system to the joint environment working with the Grid Military Users Group and Global InfoTek, Inc. We developed the Joint Grid-Based Integrated Targeting (JGIT) demonstration as an initial implementation of such a capability. In the JGIT configuration, CAST agents cooperate with a Joint Battlespace Infosphere (JBI)-compliant publish/subscribe service that connects to the US Army's Maneuver Control System (MCS)-Lite resource. The JGIT configuration of CAST also interacts with a US Air Force Air Operations Database (AODB).

4.3.3 Additional Transition Opportunities

In September 2001, LM ATL installed CAST as part of the COMSIXTHFLEET (C6F) Distributed TCS Limited Objective Experiment (LOE). We traveled to Gaeta, Italy, and successfully installed and configured CAST on the C6F Web server. Unfortunately, the events of September 11 interrupted the LOE and the USS LaSalle left Gaeta. However, all the systems installed for the LOE, including CAST, remain installed and 6th Fleet intelligence personnel plan to perform the LOE as time permits.

LM ATL is integrating its Joint Inter-Agency Task Force – East (JIATF-E) Case Agent (JECA) capability and the CoABS Grid in the WebTAS system developed by Logicon and installed at JIATF-E in Key West, FL. As part of WebTAS, JECA will be routinely used by operators to monitor SeaLink data for critical events. The new WebTAS version including JECA is also scheduled to be installed in at least 10 Air Operations Centers.

As part of our Hunter Standoff Killer Team (HSKT) Advanced Concept Technology Demonstration (ACTD) work with Army Applied Aviation Technology Directorate (AATD) we are developing intelligent agents that support situation and threat assessment for the airborne battle management system. We will deploy the CoABS Grid to provide system interoperability and will explore linking the HSKT systems to the JGIT configuration.

For the Logistics Command and Control Advanced Technology Demonstration (LogC2 ATD) sponsored by Army Communications-Electronics Command (CECOM) we developed intelligent agents to recognize, alert, and suggest to the user when alternative courses of action might be necessary. Ongoing work focuses on supporting closer cooperation between operational and logistics command and control, using the CoABS Grid as the interoperability layer.

Promising future transition avenues include participation in the SPAWAR Information Operations Center of the Future (IOCOF) demonstrations of TCS capabilities and integration of CAST and the Grid in the GCCS-M 4.X horizontal integration effort. Figure 5-3-3 summarizes our transition schedule and plans.

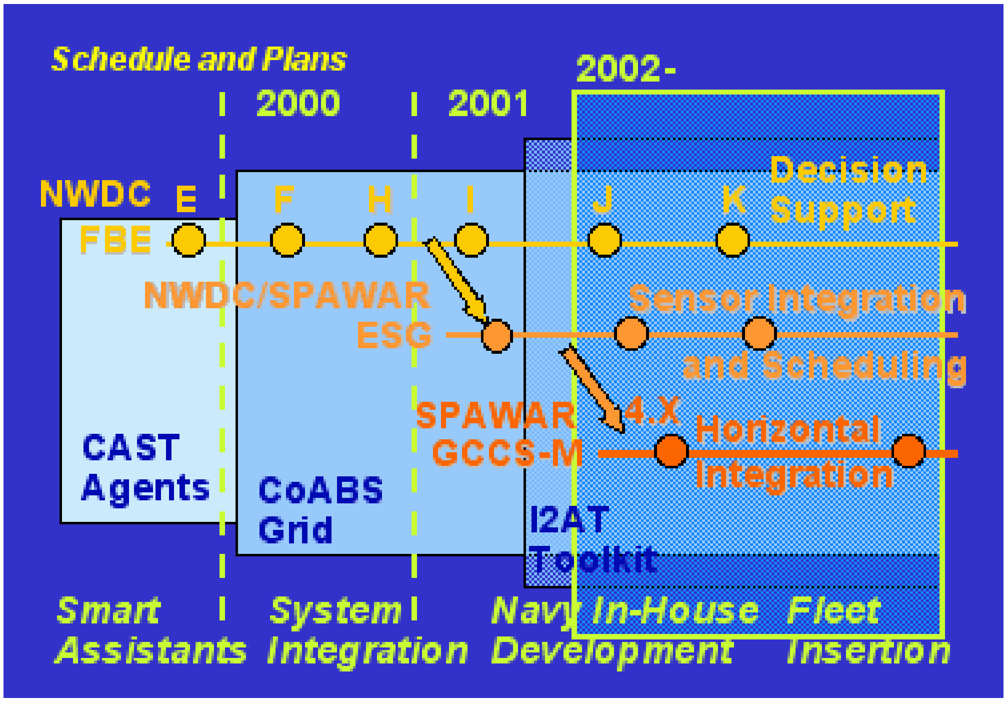


Figure 4-3-3. Transition roadmap for our CAST agent technology. We hope to transition agent-based decision support and sensor integration capabilities to GCCS-M 4.X.

4.3.4 Mobility TIE

Dartmouth College and Lockheed Martin Advanced Technology Labs (ATL) are working together under the DARPA Control of Agent Based Systems (CoABS) program to demonstrate the results of mobile agent research in military applications. As a participant in the Mobility TIE, ATL compared the performance of its EMAA agent platform with a traditional client server approach and several other CoABS agent systems. The experimental task had up to twenty clients retrieve data from a single server. The experiments showed that agent solutions outperform the client server arrangement over a large and useful operating range. EMAA agents perform particularly well and excel in low bandwidth networks, as illustrated in Figure 5-3-4 below.

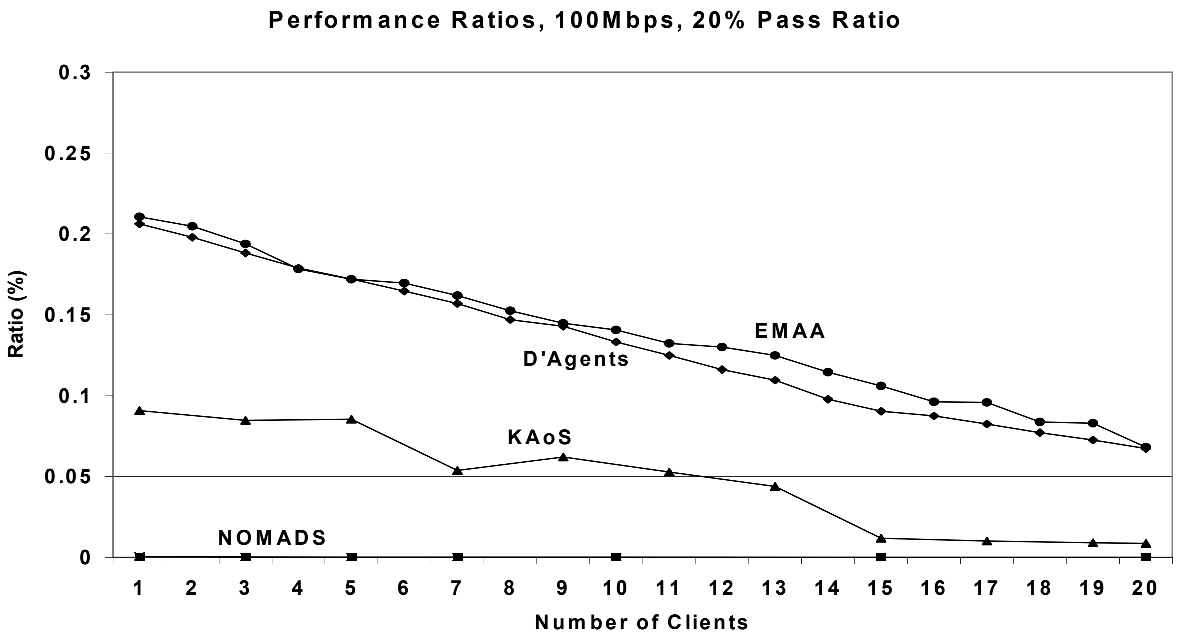


Figure 4-3-4 EMAA Agent Performance

5 Transition

5.1.1 Transition Process

Successful technology innovation is often measured in terms of transition impact. The scope and scale of a transition may take several forms and is dependent upon the type of technology under development and the desired impact. Technology transition can be divided into four categories. An analysis of the characteristics of a technology and the desired impact define the categorization of technology transition into one of the following four categories:

- **Category 1:** Technology innovation in the most basic form may not directly transition into a product but instead paves the way for further innovation. This **"foundation transition"** is a successful transition even though no one transitioned product can be identified.

- **Category 2:** Technology innovation may produce a product that performs a particular function or solves a particular problem faster, better or cheaper. This technology is typically integrated into various programs that require that function and is transparent to the user. This "**core technology transition**" is successful if one or more programs integrate the core technology into their overall program and achieve the envisioned benefits. The development of a new multi-sensor fusion algorithm is an example of a core technology that would fall into this category. The quality of the information produced in the system will be better but the user does not fundamentally change the way he performs his task.

- **Category 3:** Some technology products not only require a core transition but also require a shift in the way the user performs his/her job to achieve true benefits. An “**operational transition**” requires both a core technology transition and a fundamental change in the way that an operational user performs his task, requiring the additional transition of both a concept of operation and training products. Technologies such as [EXAMPLES] are examples of technologies that have made successful operational transitions.

- **Category 4:** The most complex type of technology transition requires core transition and a shift in the way systems are designed and developed or a “**paradigm transition**”. A paradigm transition may or may not be accompanied by an operational transition. The advent of object-oriented technology is an example of a category 4 transition that did not necessarily require an operational shift, while the invention of JAVA-based Web-browsers is an example of a category 4 transition that did require an operational change.

A successful transition and the degree of technology impact depends not only on the worth of the technology but recognizing the type of technology transition required for success and putting into place the right process to proactively support the transition. A category one is the simplest type of transition to successfully achieve. The transition is usually to a program being executed by the original developers of the technology itself.

A category two and a category three transition are the most common types of technology transition executed. Both are typically executed successfully through the transition into a single (primary) target application where the benefits are clearly visible. A successful category three transition is also accompanied by a close working relationship with the operational users to define, develop and execute a change in the overall concept of operation to incorporate the new technology. This traditional technology transition process may transition products into an ACTD or an operational program. The benefits achieved by the use of the technology in the single target application are often enough to promote the wide spread use of the technology in other applications with similar functional requirements.

The fourth and last category is harder and requires an approach that encompasses a more end-to-end transition to facilitate the paradigm shift in the way people develop systems to the way that the technology is used. This type of technology and a successful transition often reaps the most widespread and long-term benefits. The characteristics of the technologies that require this type of transition and a business process model to execute category four transitions are described in this section. While particular agent application and technologies may be transitioned via a category two and three transitions, the true power and benefits of agent technology will be realized only through a category four transition.

5.1.1.1 Characteristics of Category Four Technology

Recognizing the type of transition that must be executed and laying a foundation for the execution is equally as important as the development of the technology itself and requires just as much though and in some cases luck and timing. Most technology products can be transitioned though a category two or three process but in some instances the scope of the envisioned impact is not fully achieved. Technologies that fit into this category require a category four-transition process and often share the following characteristics:

- Require a paradigm shift to fully achieve benefits
- Benefits include both operational performance and business process benefits (cheaper maintenance, code reuse, etc…)
- Benefits are measurable within a single scope (program) and across programs (improved interoperability)
- Potential benefits have broad applicability and are wide reaching (lower development cost)
- Technology tends to be more general in nature (infrastructure vs. functional)

In addition, technologies of this type which have been **successfully** transition also share an addition set of characteristics:

- Enables a drastic improvement in process and in overall functional capability
- Provides an attractive capability to users without dictating form and content of application
- Supports and extends the concept of interoperability
- Clear vision of technology use
- Clear vision of technology benefits.

5.1.1.2 Category Four Transition Process

Two examples of technologies that fit the above criteria are OOA/OOD and Java. Analysis of the evolution of these technologies yields several key factors in the success of both of these technologies transitions.   Critical transition elements include:

The following transition model has been derived from several "successful" category four transitions. The technology transition model utilizes proof-of-concept prototypes coupled with a business plan to define an end-to-end process. Success is defined through achieving the required paradigm shift within a major program and setting into motion the institutionalization of the paradigm shift through more formal methods. The process is divided into seven key steps:

- Define a Vision
- Define and execute an initial challenge application
- Define and develop design, development and support tools
- Develop Proof-of-Concept Application,

- Define and develop a Critical Transition Application,
- Support the Institutionalization of changes
- Support General Use.

5.1.1.2.1 Technology Vision

The first step in the transition process is to create a strong technology vision that clearly illustrates the characteristics of the technology, the use of the technology and the potential impact of the technology. The vision should be a living documented and must be refined in light of the realities of the business aspects. Some areas that appear to be the right areas to apply the technology may not in fact be ready or nimble enough to support the transition.

In the early days of Java (Oak), the developers envisioned a software platform that was portable, architecture-neutral, reliable, secure and interoperable. The early impact of this technology was described in term of effect on the small consumer electronics business. Although that particular industry was not ready to use the technology at the time, a reassessment of the business plan produced several options including the WWW.

5.1.1.2.2 Initial Application

The second step is to create an initial application or set of applications to act as a test and evaluation platform for the technology and to support the requirements definition. This initial application also provides a way to illustrate the technology and potential impact to solicit feedback from the potential transition communities. This application or set of application is typically focused on providing a particular capability that is new or drastically improve some existing capability and focuses on illustrating the functional benefits. For the Java developers, the development of the "StarSeven" device fulfilled this role.

5.1.1.2.3 Tools

Another key step is to make the transition as easy as possible to accomplish. Any type of paradigm shift is difficult to accomplish because it involves not only doing something a different way but thinking about it in a new way. Technologies that have successfully made paradigm shift transition have overwhelming be accompanied by a set of authoring tools to support the use of the technology. In the case of JAVA, the emergence of the JDK played a significant role in easing the transition to using JAVA, amplified buy the fact that it is free. A key focus of the follow-on program to MACOE, I2AT, is on tool support.

5.1.1.2.4 Proof-of-Concept Super Application

The next step is to define and develop a large-scale application that will demonstrate not only the functional benefits of the technology but the business aspects also. This type of application involves not only a single application buying into the paradigm shift but multiple users and is often hard to execute.

Typically at this point in the transition the basic tenants of technology have been formed and while some technical issues remain, the basic technology has been proven. Many programs it's too early and risky to buy into the whole scale use of the technology. The proof-of-concept is often a demonstration program outside product programs. The demonstration application domain should be broad and general (Planning, monitoring), clearly demonstrate new or improved functional capabilities and new or improved "business capabilities" (quicker development time, code reuse, etc…). For the Java community the Proof-of-Concept application was the WebRunner/HotJava Browser. It clearly demonstrated the new capabilities of making Web pages "live" but also demonstrated the "business" advantages of the "write once, run anywhere" concept. Potential technology users saw both the benefits of the technology and the reaction of the customer community.

5.1.1.2.5 Program/Product Transition (Netscape)

Once the Proof-of-Concept has been successfully achieved, the next step is to transition the technology into a major program/product. Potential program/product transition options should be identified and worked in parallel with the proof-of-concept demonstration. This step is a critical juncture in the transition and most often technology transition fails because the proper foundation for the transition has not been laid. In the Java example, because of the success and the customer community reaction to WebRunner, Sun executive convinced Netscape to incorporate Java technology into their Browser product. The Netscape Browser acted as an early example of a Super-Application framework, enabling customer to program their Java applets that operate within the Netscape framework.

5.1.1.2.6 Institutionalization

As technology is transitioned into use via the transition application, the desire to grow the technology and apply it elsewhere emerges. For the technology to move forward in both use and capability it has to do so in both an organized fashion and one that was inclusive of the potentially wide technology user community. In the Java transition, Sun recognized that for the last pieces to fall into place they could not be the sole controller of the technology and supported the creation of committees to support the "institutionalization" of changes by an independent group.

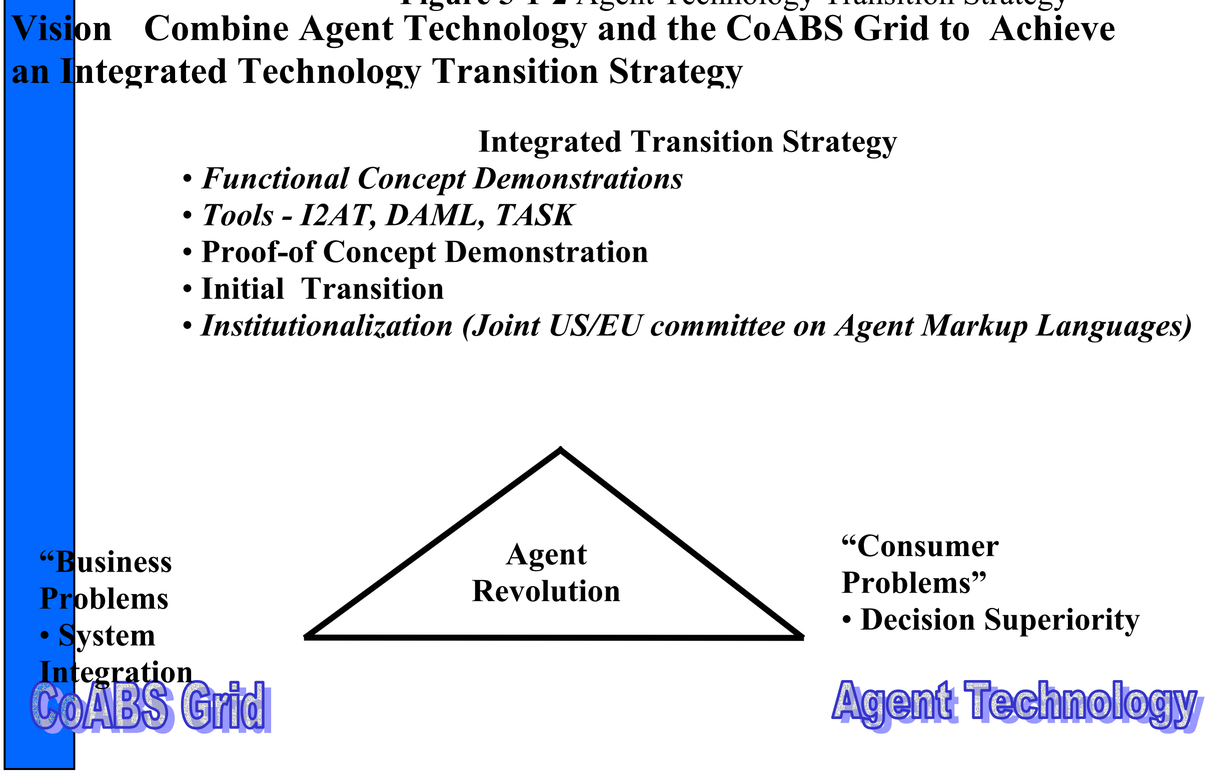
5.1.1.2.7 Support General Use

After the benefits and the technical soundness of the Java technology was demonstrated through the Netscape transition, developers of all types of application were wondering if the technology could be applicable to their applications. Sun clearly promoted and supported the widespread use of the technology outside the Web. An organization that promotes the use of this technology and provides services to promote widespread use is the last step in achieving a full-scale category four transition.

5.1.2 Recommendations

Agents Transition requires a paradigm shift, requiring end-to-end view of transition. The paradigm transition that agent technology must achieve to realize their full benefits and the transition that Java took above are very similar. For many application to say that a Web interface is not useful is absurd, similarly we want this same sentiment to be the view of the Grid. This section presents options for executing the business process model defined above for a category four transition.

Figure 5-1-2 Agent Technology Transition Strategy



Several of the transition steps are currently in process but a plan incorporating these elements must be derived and several key step must be put in place, such as the Proof-of-Concept application, for the true benefits of Agent technology to be realized.

5.1.3 ESG

Under MAOCE CoABS, and follow-on funding, LM ATL work with NWDC, SPAWAR, DARPA, and Global InfoTek to deploy the Grid, Toolkit, and CAST agent technology as part of the Expeditionary Sensor Grid (ESG). The ESG is one of a few high priority initiatives of NWDC and will be a component of Millennium Challenge '02/FBE-J, to be held in June/July 2002.

LM ATL has installed the CAST system used in FBE-I at the SPAWAR Information Operations Center of the Future (IOCOF). CAST is integrated via the CoABS Grid to a SPAWAR supplied MIDB instance and a C2PC track database. Installation and system configuration was accomplished in less than a day. SPAWAR personnel plan to use the CoABS Grid to connect additional information sources to the CAST system.

5.1.4 JIATF-East Case Agents (JECA)

The drug interdiction domain is a very information rich environment. Information needs to be collected from a variety of resources in order to form a comprehensive picture of the events that are in the process of unfolding. The absence of any individual piece of data could imply a totally different scenario. The intelligent agent work developed under the MACOE program has proven to be an excellent mechanism for collecting information from heterogeneous data resources. This, coupled with the CoABS Grid work, is a powerful combination for developing a reliable and comprehensive picture. Having identified the opportunity have an impact at JIATF-East, and recognizing the contribution that Case Agents can have on providing analysts the information they need to develop a more comprehensive and timely picture of domain related events, the goals of this program are:

- Apply Case Agents towards the Drug Interdiction Domain Space.
- Transition Intelligent Agents and CoABS Grid into a real-world operational environment.
- Leverage work done under the MACOE and CoABS programs to provide analysts with more comprehensive and timely picture of domain related events.
- Integrate Case Agents into WebTAS to meet the goals defined above.

5.1.4.1 Approach

Our approach to this effort was to employ the "Iterative Development" approach to application development. This approach generally consists of three stages, which are repeated until the application meets the needs of the operator. These stages include: knowledge acquisition, software development and customer feedback. Generally after the software development stage, a leave-behind system is installed and evaluated by one or more operators. Feedback is then collected from the operators and is used to better tailor the application to meet the task specific needs of the operators. In this case, it became obvious that capabilities such as monitoring agents, persistent query agents, nested queries, and drill down were essential to the operators. It also became apparent that a mechanism for domain abstraction was required since there were such a broad variety of heterogeneous data resources to query against. Since several of the data resources were web-based, introduction of "web scraping" techniques was required. Finally, to insure successful integration into the JIATF-East environment, a teaming arrangement with Logicon was sought to develop and deploy an embeddable JECA client into their WebTAS application. The client provided WebTAS the capability to task JECA agents to retrieve data for WebTAS operators.

5.1.4.2 Results/Status

In May 2000, the initial proto-type was completed and demonstrated at JIATF-East. The onsite analysts greeted the features and functionality of the initial proto-type warmly. The second iteration proto-type leave-behind system was developed using the feedback collected from the analyst. New data sources were added as well as new features requested by the analysts. This system was delivered, but never demonstrated due to in-surmountable issues raised by the newly instated JIATF-East Command leadership. In order to proceed, we cultivated a relationship with Logicon who expressed interest in the capabilities we were providing JIATF-East. The new approach that grew from that relationship was to integrate JECA into WebTAS. Since this approach matched the goals set forth by this program and Logicon's goals, we proceeded to develop an embeddable client that allowed WebTAS to task JECA agents. The client has been embedded into WebTAS and has been deployed as a back-fit to WebTAS version 2.1 in October 2001. JECA is currently planned as base functionality in WebTAS version 2.2 which is expected to be deployed CY 2002.

JECA is currently deployed as part of the WebTAS version currently running at JIATF-East. Some minor problems related to the CoABS grid have been identified and are currently being evaluated, however, the problems are not serious enough to prevent deployment. We have discussed the issues with GITI and expect that they will be worked out in the near term. JECA is now a functional piece of a DEPLOYED system.

5.1.4.3 Future Improvements

In the process of developing an integration plan with WebTAS, several features were identified that could not be developed due to funding limitations, schedule, and relative risk. Should the opportunity for continued work in this domain space, this features could help provide more timely and complete information to the analyst. These features include: automation of the watch-list processing, re-integration of the Tipper e-mail processing, enhanced monitoring agents in support of WebTAS' truth maintenance, and configurable monitoring agents for managing user defined alerting criteria. Watch-list and Tipper e-mail process automation provide new data resources that the intelligent case agents can access. The addition of the monitoring and alerting features gives analysts the capability to monitor the plethora of data resources for individual events in near real-time, providing a timely picture of events as they unfold. These features also provide the analyst the ability to set up alerting criteria that can act as a triggering method for activating new cases. In addition, monitoring agents used for truth maintenance will provide a more accurate picture and can stand watch 24/7 providing the most up to the moment status.

6 MACOE Follow-on (Interoperable Intelligent Agent Toolkit I2AT)

6.1 The Problem: One Size Fits One

As we fielded multi-agent systems in military command and control operations, users ranging from operators to decision makers to commanders recognized the potential power of agent technology and enumerated a plethora of additional capabilities they would like to see delivered. Desired capabilities ranged from small changes, e.g. retrieve images from a second IPL server, to complex decision support, e.g. prioritize TCS. With the CoABS promise of rapid configuration of specialized systems, such enhancements should not take years to filter through the procurement chain and individualized system configurations should be easy to create and maintain.

6.2 The Answer: The Interoperable Intelligent Agent Toolkit

Starting in 2001, we set the objective to develop an Interoperable Intelligent Agent Toolkit to answer this requirement. With the help of the toolkit, military users and their support staff will be able to modify and enhance a basic multi-agent system without, or with minimal, support from contractors. The toolkit will produce interoperable systems, because it will leverage the CoABS Grid. It will offer Grid-enabled functional and infrastructure components as building blocks. It will leverage ATL's workflow models of agent behaviors. This technology will put useful decision support applications into the hands of users faster and more affordably than currently possible. The first release of the toolkit is planned for Spring 2002.