



FAST FACTS

COMPANY

Department of Defense

INDUSTRY

Joint After Action Review

GEOGRAPHY

Various Defense Locations

CHALLENGES

- After Action Review (AAR) data stored on heterogeneous external systems must be easily accessible, and translatable into joint AAR knowledge
- Must support multiple operating systems, languages & protocols

SOLUTION

- Recursion Software's *Voyager Edge™* & *Cinergi™* *Multi-language Mobile Agent Integration Platform*

RESULTS

- Implementation period of intelligent mobile agent accessing external system completed in 3 weeks
- Minimal code for inter-process communication and external system access
- Multiple protocol support for direct access to external systems
- Data retrieved, filtered & analyzed at the source

ORGANIZATION

The Government's DoD currently contains multiple After Action Review (AAR) systems for each branch of the military (Army, Navy, Air Force & Marines). After Action Review is defined as the analytical review of training events that enable the training audience to examine the actions and results during a training event.

The Joint After Action Review (JAAR) is a process designed to provide commanders with direct training feedback on the accomplishment of selected Joint Task Articles, conditions, and standards stated in terms of training objectives. This information allows the commander to evaluate training proficiency required in the assessment of a Joint Training system.

CHALLENGES

The JAAR solution entails the assimilation and transformation of a high volume of simulation data from various external AAR systems into knowledge pertinent to joint review across all the Departments of Defense. One such external system, OneSAF, is a Microsoft-based AAR server that communicates using TCP/IP sockets and a custom protocol that involves grammar rules and parsing. The process uses a binary protocol between various operating systems that is custom coded at the TCP/IP socket level.

The OneSAF server described above is written in VB and the intelligent mobile agents are written in Java. The Government is committed to the ubiquity and openness of Java technology. Additionally, they indicate that it would take their engineers a great deal of time and money to expose their VB server to Java unless they use Simple Object Access Protocol (SOAP), which was less than desirable due to its performance limitations.

SOLUTION

INTELLIGENT MOBILE AGENT INTEGRATION PLATFORM

Utilizing *Voyager Edge* Intelligent Mobile Agent Framework from Recursion Software, Inc., the Government streamlined the access, filtering and analyzing of data stored in heterogeneous systems. This solution was implemented in less than four weeks with the necessary code provided for intelligent agent mobility and communication between systems that were written in different platforms (Java and .NET).

Recursion Software worked closely with the Government's software engineers to design and execute a proof of concept using *Voyager Edge* to build Java-based intelligent mobile agents communicating with a Microsoft-based OneSAF server, using auto-generated, fast binary protocol. The intelligent mobile agents accessed, filtered and analyzed the data at the external system, translating the AAR data into joint AAR knowledge. The Government will use this prototype to build the remainder of the JAAR solution.

PROOF OF CONCEPT

GOALS

- Prove that Voyager adds value by shortening development time on agent creation/features thus freeing developers to work true JAAR development.
- Prove that Voyager offers easy integration capabilities between various languages and platforms, which is not an easy task at a binary level.
- Prove that the Voyager Wizard is easy to use, flexible and saves development time.
- Demonstrate that Voyager can meet required performance objectives of the DoD’s JAAR.

OBJECTIVES

- Determine the hardware and software environment in which Voyager is expected to run (*platform, operating system and version, Java version, compilers and versions*).
- Determine the network configuration under which Voyager is expected to run (*multi-homing, firewalls, clustering, NATing, etc.*)
- Determine current and future deployment architecture of the DoD’s JAAR.

DESCRIPTION OF USE CASE

ONESAF INFORMATION ACCESS PROCESS

- The Voyager Adapter Agent establishes a query session with the OneSAF server
- The Adapter Agent signs on with a userid/password
- The Adapter Agent sends a query for OneSAF info
- The Adapter Agent receives XML-based AAR knowledge from the OneSAF server asynchronously
- The Adapter Agent disconnects from the server and converts AAR data into JAAR knowledge

ONESAF SERVER

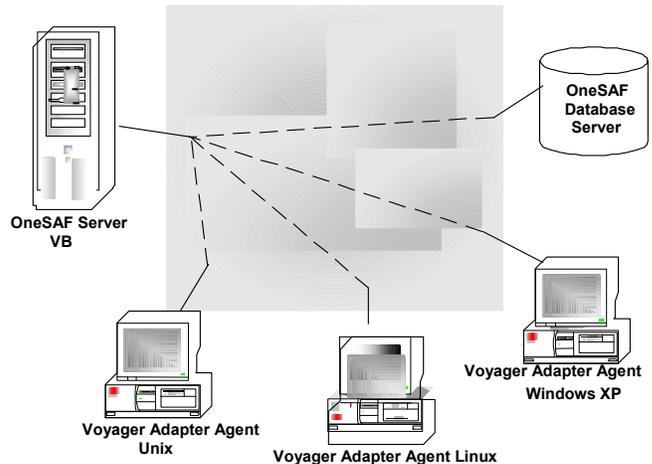
- On encountering a new security, OneSAF server sends a query message to OneSAF database
- The OneSAF server responds with a stream of AAR data
- Adapter Agent filters and analyzes OneSAF AAR data turning it into JAAR knowledge

ONE SAF SERVER

- Windows XP Server
- VB, C++, C#, Java

VOYAGER ADAPTER AGENTS

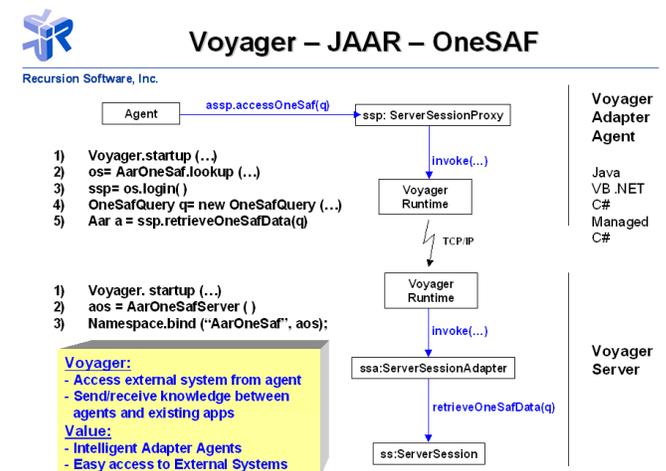
- Linux, Unix, Windows XP



TEST ENVIRONMENT

- “Virtual IP” – point to any of the boxes
- VLAN consisting of “n” LANs (1/AAR System)
- Mix of machines running Linux, Unix and Windows
- Less than 20 data elements made up of: AAR simulation data
- Performance metrics today: # of simultaneous users = 50, Performance = 150 queries/sec

VOYAGER COMMUNICATION



OVERVIEW OF VOYAGER EDGE

Voyager *Edge*™ Intelligent Distributed Computing Platform is a powerful next-generation intelligent distributed computing platform for creating and extending applications seamlessly in Java and .NET on devices and systems ranging from edge devices to enterprise servers

Voyager is regarded for its intelligent mobile agent technology, interoperability, and ability to support a total range of edge devices, including handheld devices, PDAs, sensors and cameras. Voyager Edge intelligent mobile agents can be written in either of the languages listed above and it can communicate with other mobile agents written in either Java or .Net. When processing, it can move from a device running a Java Virtual Machine to a device running the Microsoft CLR and vice-versa. It can also be exposed as a Mobile Web Service, with no coding required for access from any client language regardless of the agents current or future location.

Voyager *Edge* provides a tremendous level of flexibility for developers to quickly and easily establish communication between components (objects) within

an existing application and between applications. This flexibility extends to a variety of development platforms that can be running in different operating system environments. It will communicate across networks using multiple networking protocols and provides a high level of scalability, reliability, security and privacy.

MANAGE DEVELOPMENT AND DEPLOYMENT ABOUT RECURSION SOFTWARE, INC.

Recursion Software, Inc. provides software engineers with powerful enterprise development solutions for constructing distributed systems using proven Java™-based methods and emerging web services technologies. Recursion Software products allow Java, C# and C++ developers to be productive rapidly, by leveraging Recursion's expertise in object-oriented techniques and incorporating industry standards. Recursion Software's customer base includes Defense, Financial, Computer Technology, and Telecommunications industries, among others. Recursion Software is headquartered in Frisco, Texas. To learn more about Recursion Software, Inc., visit www.recursionsw.com.



Copyright © 2006 Recursion Software, Inc.
All rights reserved. Cinergi, Voyager,
C++ Tookits, C# Toolkit and JGL Toolkit
are trademarks of Recursion Software, Inc.
All other trademarks are the property
of their respective owners.

Recursion Software, Inc.
2591 North Dallas Parkway
Suite 200
Frisco, Texas 75034
1.800.727.8674 or 972.731.8800
www.recursionsw.com