Enhanced Maritime Domain Awareness through
Web-based Configurable Software

Richard Dickinson, Michael Orr, and Martin O. Hofmann

Lockheed Martin Advanced Technology Laboratories
3 Executive Campus
Cherry Hill, NJ 08002
USA

Phone: 856-792-9717
e-mail: richard.j.dickinson@lmco.com

Abstract

Maritime domain awareness (MDA) of vessel traffic in a large operational area requires automated tools for watchstanders and analysts. Additionally, the users must be able to tailor the tools to keep pace with changing operational needs. Lockheed Martin Advanced Technology Laboratories has fielded web accessible software that enables operators to configure and manage intelligent agents that monitor live maritime data feeds to detect user prescribed conditions and analyze historic data. The tools are compliant with interoperability standards and realized in a Service Oriented Architecture (SOA), allowing external applications to invoke the analytic capabilities provided by the underlying intelligent agents.

The software has been incorporated into daily operations at Commander, Naval Forces Europe/Commander, Sixth Fleet (CNE-C6F) and NATO Component Command Maritime (CCMAR) in Naples, Italy where operators are now able to transform an overwhelming amount of data into meaningful, actionable knowledge, resulting in improved situation awareness and successful maritime intercept operations.

This paper will describe the novel capabilities of the software and how it supports MDA activities in a web-based environment. The paper includes an overview of the Defense Advanced Research Projects Agency (DARPA) program that funded the development of the software and how the program applied technological innovation to user requirements to create an operational product.

Introduction

Over the past thirteen years, Lockheed Martin (LM) Advanced Technology Laboratories (ATL) has worked closely with the Defense Advanced Research Projects Agency (DARPA) in the investigation of agent-based computing technologies. Initially, as part of the DARPA Control of Agent Based Systems (CoABS) program, LM ATL and DARPA researched and demonstrated fundamental concepts of agent-based computing. In its original phase, the CoABS program investigated techniques to safely control, coordinate, and manage systems of autonomous agents. Subsequent CoABS studies demonstrated how intelligent agents could be used in military environments to automate command and control functions such as information collection and dissemination, mission planning, and situation assessment. The studies
showed that software agents provided an efficient and effective development paradigm for automating portions of user workflow. However, a key enabler for ubiquitous adoption of the agent paradigm was still missing; the ability for end users to specify agent behaviors without programming. Additional research within the program examined the possibility of providing end-users with an agent toolkit to allow them to compose agent-based systems for their specific needs. The results of the agent toolkit assessment indicated that while an agent workflow composition approach was somewhat too complex and demanding for time-pressured end-users, an effective middle ground might exist between end-user system composition and hard-coded system development; perhaps a way to allow end-users to configure pre-fabricated agent components to effectively apply agent-based capabilities to their dynamic operational environments.

Encouraged by the results of the CoABS program, and to continue the investigation of the application of agent technologies in an operational environment, DARPA funded the Fast Connectivity for Coalitions and Agents Project (FastC2AP). The project endeavored to demonstrate that agent-based configuration tools could be employed by operational end-users to effectively tailor agents that were adaptable to new situations and emergent operational demands. As part of the FastC2AP development effort, ATL created the Fast Agent Configuration Tool (FACT), a web-accessible, agent-based software application that allowed operational end-users to configure, execute, and manage intelligent agents that assisted with daily operational tasks such as information retrieval and data analysis. The FACT software targeted the perceived sweet spot between end-user system composition and hard-coded system development by providing a set of agent templates that allowed non-technical end-users to quickly configure agents to conduct operationally relevant tasks without having to know or understand the agent’s underlying workflow(s).

DARPA coordinated with the U.S. Navy Warfare Development Command (NWDC) to identify an appropriate operational realm in which to conduct the FastC2AP effort. After examining several candidates, the Commander, Naval Forces Europe/Commander, Sixth Fleet (CNE-C6F) in Naples, Italy was selected as the FastC2AP experimental domain. The FastC2AP development team worked closely with CNE-C6F staff over the course of the project to develop and evaluate the FACT software. Through knowledge engineering sessions and technical interchanges with members of the CNE-C6F staff, the FastC2AP engineering team identified a broad set of information processing needs within the Command that could be addressed with agent-based computing capabilities. From the broad set of needs, the FastC2AP team identified associated work tasks to be represented and implemented as configurable agent templates within the FACT tool suite. During the entire software development process, the FastC2AP team coordinated with CNE-C6F to ensure that the FACT application would meet the operational needs of the Command.

CNE-C6F participated in the FastC2AP initiative with the agreement that if the resulting software proved to be effective, it would remain installed at their facility beyond the project’s assessment and evaluation activities for long term employment by the CNE-C6F staff. During the assessment period, CNE-C6F validated the operational utility of the FACT software and elected to retain the software and incorporate it into their Watchfloor’s daily operations. Additionally, based on a recommendation from CNE-C6F, the software was installed within the Mission Operations Center (MOC) at the NATO Component Command Maritime (CCMAR) facility, also in Naples, Italy. The FACT software is currently fully operational at both CNE-C6F and CCMAR and continues to enhance maritime situation awareness at these sites. The FastC2AP development team supports and maintains the current software release and is working closely with the operational users to define enhancements, improvements, and requirements for future software releases.
Capabilities Description

The FACT software enables operational end-users to assemble and manage intelligent agents that conduct information retrieval and analysis tasks on the user’s behalf. The FACT intelligent agents can be configured to monitor live maritime data feeds to detect prescribed conditions and to analyze historic maritime data. A web-enabled user interface permits operational end users to configure and manage the agents using any web browser. Additionally, a web service application programming interface (API) allows external applications to programmatically access the FACT agent capabilities from within a Service Oriented Architecture (SOA).

Agent Configuration Process

In order to understand the capabilities of the FACT system, it is important first to understand the way in which an operational user assembles a FACT agent. The configuration process, Figure 1, begins when a user selects one of the prefabricated FACT templates. The templates are programmatic implementations of data processing functions typically performed manually by watchstanders and analysts in an operational maritime domain. The user tailors or customizes the template to his/her specific operational needs by specifying pertinent parameters and values. Once the template data entry is completed, the user can immediately execute the customized template as an agent and/or save it as a configuration for later use. Saved configurations can be recalled subsequently by any FACT user to quickly create similar type agents.

Figure 1. Agent Configuration Process
The agent configuration process can be likened to the process of new home construction. The act of a FACT user selecting a template is similar to a new home buyer selecting a blueprint from a builder’s catalog of available home plans. The FACT template configuration process is akin to a home buyer taking the selected stock plan and customizing it with specific options (e.g., adding an additional room, selecting exterior color and trim, or interior options) that suit his/her needs. Finally, the resulting FACT agent is comparable to the house itself—the physical manifestation of the construction process. And just as the home buyer is not involved in the physical construction of the house, the FACT user does not have to build the agent or even know how the agent is built.

Templates

Ten template types are included in the current FACT software installation. Each template was designed to perform a specific operational task. In the development of the templates, ATL engineers worked closely with maritime operations subject matter experts to identify operational data processing tasks that could be automated through the employment of agent technologies. The templates, when properly configured by an operator, can be used to produce agents that perform operationally relevant tasks for the operators. Table 1 provides a list and descriptions of the template types.

Table 1. FACT Agent Template Types

<table>
<thead>
<tr>
<th>Template Name</th>
<th>Description</th>
<th>Operational Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abnormal Vessel Speed</td>
<td>Produces agents capable of detecting vessels whose calculated rate of movement in a given geographic area is below a specified minimum speed threshold.</td>
<td>Typically used to detect vessels that are loitering in open water or any area designated to be a “hot zone.”</td>
</tr>
<tr>
<td>Chat Monitoring</td>
<td>Produces agents capable of monitoring IRC chat channel communications to detect exchanges containing specified keywords or phrases.</td>
<td>Typically used to simultaneously monitor multiple maritime and intelligence chat channels to detect messages of interest.</td>
</tr>
<tr>
<td>Course Proximity</td>
<td>Produces agents that generate a list of vessels that had been in geographic proximity to a specific target vessel during a specific period of time.</td>
<td>Typically used to obtain all waypoints of all positions of all vessels that had been within a specified distance of a target vessel during a specified time period.</td>
</tr>
<tr>
<td>Geographic Feasibility Calculator</td>
<td>A utility program used to compute a reasonable amount of time required to travel the shortest water route between two given latitude, longitude points.</td>
<td>Provides a rough estimate on the amount of time it would take for a vessel to transit between two points.</td>
</tr>
<tr>
<td>Geographic Proximity</td>
<td>Produces agents capable of detecting when a vessel reports a position within a specified geographic area of interest.</td>
<td>Used to detect all vessel movements by all vessels in a geographic area.</td>
</tr>
<tr>
<td>Next Waypoint Feasibility Monitor</td>
<td>Produces agents capable of determining the practicability of a vessel to transit from its latest reported position to a specified position in a specified amount of time.</td>
<td>Typically used to detect when a vessel deviates from its planned schedule.</td>
</tr>
<tr>
<td>Vessel In or Out of Area</td>
<td>Produces agents capable of detecting when vessels have crossed the boundaries of a given geographic area.</td>
<td>Typically used to detect when vessels enter or leave a port area or any area designated to be a “hot zone.”</td>
</tr>
<tr>
<td>Vessel Proximity</td>
<td>Produces agents capable of detecting when any vessel reports a position within a specified proximity to a target vessel’s most recently reported position.</td>
<td>Typically used to put a monitoring “bubble” around a vessel of interest to detect encroachment by other vessels.</td>
</tr>
<tr>
<td>Template Name</td>
<td>Description</td>
<td>Operational Value</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Vessel Rendezvous</td>
<td>Produces agents capable of detecting potential rendezvous between vessels by identifying pairs of vessels whose reported positions indicate they have violated a specified minimum distance of separation within a specified geographic region.</td>
<td>Typically used to monitor areas where it would be uncharacteristic for two vessels to be in close proximity, i.e., open water.</td>
</tr>
<tr>
<td>Flexible Templates</td>
<td>Produces agents capable of detecting vessel characteristic change, vessel position change, geographic proximity, and vessel proximity.</td>
<td>The most sophisticated of all template types, can be used in a variety of ways to build agents with complex search and monitoring criteria.</td>
</tr>
</tbody>
</table>

**Configurations**

A configuration is a customized instance of a template. Configuration construction consists of a user specifying required values for a template’s input parameters, selecting appropriate operational settings, and saving the completed product as a named configuration. Named configurations allow a user to come back to favorite customizations without having to reconfigure the template each time. Named configurations are shareable among users so a common configuration can be constructed one time by any user and used repeatedly by all other users. Over time, users at a site will build up a library of configurations that are pertinent to the operational functions for that particular site. Figure 2 shows the

![Figure 2. FACT Web Portal—Agent Configuration Screenshot](image-url)
FACT Web Portal page that is used by the operational user to configure FACT templates. Within Figure 2, the available templates are shown on the left hand side of the display; note that in this case the Geographic Proximity Template is highlighted for selection. The middle of the display in Figure 2 shows all of the existing Configurations for the selected Geographic Proximity Template; note that in this case the Hawk Island GeoProx configuration is highlighted for selection. The right hand side of the display shows the current parameterized settings for the selected Hawk Island GeoProx configuration.

Agents

In the context of the FACT system, an agent is a software program that can perform a sequence of tasks, defined as a workflow, on behalf of a user. The FACT agents, which utilize ATL’s Extensible Mobile Agent Architecture (EMAA) framework, employ artificial intelligence mechanisms to operate as autonomous entities. Within the FACT application, an agent is created when a FACT user “launches” a configuration. Launching is the process of selecting an existing configuration, making any final adjustments to configured values, and then executing the configuration. Once an agent is launched, it is an independent instantiation of the configuration from which it was rendered. Therefore, any operations performed on the agent do not impact the configuration and vice versa.

Search Modes

When an agent is launched, it can be executed in one of two search modes: historical and future/persistent.

Historical

Agents launched in historical search mode examine archived data to locate information meeting the user specified search conditions. An agent launched in historical search mode inspects the underlying FACT data repository to locate and extract archived data that meets the agent’s search criterion, reports the data to the user, and then terminates execution. Agents that execute in historical search mode are also known as Single Query Agents because they report results and then cease execution.

Future/Persistent

Agents launched in persistent/future search mode monitor and examine incoming data to detect new data meeting the user specified search conditions. Agents launched in persistent search mode have cyclical periods of activation and hibernation. The agent will ‘hibernate’ for a period of time (as determined by the configurable persistence period) and will activate when the period expires. Upon activation, the agent will: examine any new data in the FACT data repository to determine if the agent’s prescribed watch condition has occurred, return matching results (if any), and proceed to ‘hibernate’ again. This cycle is repeated indefinitely or until either the agent is stopped by an operator or a specified expiration date/time is reached.

Agent Monitoring

When an agent is launched, an entry will appear in a section of the FACT Web Portal known as the Agent Monitor Pane. The Agent Monitor Pane allows the FACT user to control the agents (e.g., stop, remove) and to access each agent’s result set. Each agent entry within the pane has a hyperlink that the user can “click” to obtain the agent’s current result set. The results hyperlink is rendered as the current number of results in the set and is automatically updated as the agent collects new data that meets its assigned search condition.
Figure 3 shows the Agent Monitor Pane of the FACT Web Portal. Within Figure 3, the left hand side shows the agents that are actively executing. The top portion of the left hand side shows those agents that have been configured to execute in personal mode (visible only to the user that launched the agent). The bottom portion of the left hand side shows those agents that have been configured to execute in public mode (visible to all users). The right hand side of the display shows the results of the currently selected agent, in this case the Philadelphia GeoProx agent.

Figure 3. FACT Web Portal—Agent Monitor Screenshot

System Architecture

The FACT agent system architecture, as illustrated in Figure 4, is based on a multi-tier, client-server architectural model. The tier breakdown includes: end-user and service interaction interfaces (Client Tier), web processing components (Web Tier), agent system and algorithmic elements (Agent Tier), and data processing and storage elements (Data Tier). This section describes each tier in detail and explains the role of each within the FACT system.

Client Tier

The client tier represents the set of client-side programs that can be used to interact with the FACT system. Currently, there are three means by which a user can access the FACT system; (1) FACT Web Portal, (2) FACT Agent Web Service interface, and (3) Google Earth™ FACT Client. The FACT Web
Portal is a zero-install client interface operating in a standard web browser that allows operators to configure, launch, and control agents; examine agent results data; and view vessel characteristics data. The FACT Agent Web Service interface is a programmatic interface that exposes FACT agent capabilities within a SOA environment and can be used by application developers to create any number of custom client applications that leverage the FACT agent capabilities. The Google Earth™ FACT Client is a customized instance of a Google Earth™ client that displays the current vessel position picture, vessel characteristics data, position history traces, and agent search areas rendered in geographically accurate locations.

**Web Tier**

The web tier consists of components that control and enable web accessibility to the FACT system. Web applications are deployed within this tier to provide web interfaces that expose FACT capabilities such as agent control and data access. The web applications are hosted in a Tomcat Application Server and operate in the same security realm within the Application Server to support seamless interoperability between client types. In terms of operation, components within the agent tier can be used entirely through the Agent Web Service, entirely through the Web Browser, or any combination thereof.

**Agent Tier**

The agent tier consists of three components that transform agent configurations into executable agents, facilitate the execution of agents, and provide high-level management and control of agents.
Within the agent tier, the Workflow Manager component is responsible for creating executable agents. The process for instantiating new agents is triggered when the Workflow Manager receives an agent creation request from any of the applications residing in the web tier. The first part of the creation process is the extraction of agent definition and configuration parameter values from the user generated configuration template. After parameter extraction is complete, agent construction begins by loading the specific agent definition. Agent definition contains the agent’s execution workflow—a description of the tasks to be executed and the established route between tasks that the agent has been directed to follow.

The Runtime Manager component within the agent tier handles the high-level agent monitoring and control activities for all agents. When a newly instantiated agent is received from the Workflow Manager component, metadata on the agent is generated and stored in a persistent registry. The Runtime Manager also generates and stores a unique identifier for the agent that is used to communicate with the agent once execution begins. The agent identifier is provided back to the invoking client applications for reference in subsequent control requests, i.e., stop, restart, and remove. Once the registry entry has been generated, the agent is then passed to the Execution Sandbox for execution.

The Execution Sandbox component provides a safe place for the agents to execute, effectively and efficiently using local and external services. The sandbox includes mechanisms that can limit how the agent executes as well as the total number of agents executing at a particular time. While certain best practices have been implemented (e.g., denying requests for excessive amounts of data), the drain that a particular configured agent might have on the system can not be determined until execution time. That being said, it is important to protect the system as well as external resources from potential thrashing and other overburdening that could arise from a relaxed configuration; the Execution Sandbox provides such safeguards.

**Data Tier**

The data tier is responsible for data collection, management, and access functions. The Vessel Warehouse component gathers data from external maritime data sources, formats and organizes the data for use by the FACT agents, and stores the data in a local data repository. Accessor methods provided by the Vessel Warehouse allow the agents to examine and extract the data stored within the repository. The modular nature of the Vessel Warehouse allows for rapid adaptation to new or alternate data sources. It is important to note that the Vessel Warehouse component was introduced in Release Four of the FACT software. Previous versions of FACT, including the version presently installed at CNE-C6F and CCMAR, use a proprietary data repository developed by the Naval Undersea Warfare Center (NUWC) in Newport, Rhode Island. While Figure 4 shows the FACT Release Four Vessel Warehouse extracting data from an AIS Live data feed, the NUWC product federates data from U.S. Navy data sources.

The Result Manager component within the Data Tier stores and organizes the results accumulated by existing FACT agents. As agents execute and gather information that meets their corresponding prescribed search conditions, the Result Manager stores the “matching data” and associates the data back to the proper agent. Accessor methods provided by the Result Manager allow clients to obtain the results associated with an agent.

The Alert Manager component within the Data Tier stores and organizes alerts produced by the FACT agents. Alerts are notifications delivered to a client by a persistent agent when the agent detects its prescribed condition. Alert production by an agent is an optional feature that can be specified by the operator during the agent configuration process. The Alert Manager associates alert information to the producing agent and provides accessor methods to allow clients to obtain alert information.
Benefits

Maritime intelligence and data centers have access to a multitude of data sources and vast amounts of data. Often, there is such an influx of data that it is difficult for analysts and watchstanders to process and evaluate the data in a timely fashion. Further, with so much data from so many sources, it is even more difficult for personnel to extract the meaningful context from the data. However, the automation and intelligence delivered by the configurable FACT agents provide analysts and watchstanders with the ability to convert the overwhelming volume of data into meaningful, actionable knowledge. The agents are able to persistently, dependably, and accurately examine incoming data to find the precise information that the analysts and watchstanders have indicated is vital to maintaining situation awareness.

Data vigilance is an important benefit gained from the use of the FACT software. When executed in persistent search mode, a FACT agent will continuously monitor available data sources for the arrival of new data and will determine if the data meet prescribed search conditions. When an agent detects new data that meet its search criterion, it alerts the operator of the event. For instance, if a Geographic Proximity agent is configured to detect the movement of vessels in a specified area, it will vigilantly monitor the underlying FACT data sources for the arrival of new position reports, and will examine the new reports to determine if any positions are relative to the specified geographic area, and if so, report the incident to the operator. Without this automation, an operator would need to be assigned to continually observe a vessel track data source to detect the condition. The agent’s data vigilance capability frees the operator from the time consuming monitoring task and the agent is able to alert the operator as soon as a condition occurs.

Additionally, the FACT agents deliver the benefit of rapid discovery of data conditions that are buried within volumes of archived data. The FACT agent templates allow operators to quickly configure agents to search for information that precisely meets their needs. For example, the Vessel Proximity agent can be configured to search for and retrieve information for any vessels that had been within n miles of a specified target vessel during a specified time period. Depending upon the time window and density of the data, the agent should be able to deliver the information in several seconds. If the analyst had to perform this task manually, it could take a considerable amount of time because he/she would first have to gather the position information from the appropriate data source(s) and then manually analyze the data to compare each position of the target vessel with every position of every other vessel in the area. This considerable time savings allows an operator/analyst to monitor or investigate far more vessels during a given shift than otherwise possible and with the agents acting on the human’s behalf, the human is able to devote the “freed” time to other tasks that are vital to maintaining situation awareness.

Agents are also an ideal solution for contending with information overload because of their ability to deliver highly accurate results tirelessly. The FACT agents meticulously examine the underlying maritime data and are able to reliably and repeatedly detect and discern prescribed conditions within the data. Human execution of similar operations would be prone to calculation errors, data omissions, distraction, and fatigue. Additionally, since agents are able to operate in parallel, at any given time an operator may have hundreds of agents executing on his/her behalf thereby increasing the information processing throughput beyond human capacity.

In a more specific sense, the FACT software has proven to be very effective in detecting certain known patterns of abnormal behavior exhibited by maritime surface vessels. FACT users have learned to configure FACT templates to detect and alert on types of abnormal behavior that might be indicative of illicit activities or emerging threats. For example, the Abnormal Vessel Speed Template has been useful in detecting the loitering of vessels. A vessel may well be loitering for legitimate logistical or business reasons, but loitering under certain circumstances or within certain boundaries may be a telltale sign of criminal activity. The configurability of the FACT templates allows the user to create agents that help to
discern normal vessel behavior from abnormal behavior. In this way, the operator is only alerted to the specific behavior that he/she has designated as suspicious. With timely notification of suspicious behavior, the operator is positioned to conduct a more extensive investigation and/or to notify a higher authority to determine whether or not an interdiction should be conducted.

Findings and Results

The first release of the FACT software (Release One) was delivered to CNE-C6F in December 2005. Release One was a proof of concept prototype intended to gauge how effectively web-based agent technologies could be applied to an operational environment. The objective of the first release was to examine the viability of using agents in the CNE-C6F operational environment both in terms of accuracy of the agents and usability of software. Upon delivery and installation of the software at CNE-C6F, an evaluation was conducted to determine the usefulness and utility of the software. The evaluation determined the software to be very effective in helping CNE-C6F personnel monitor vessels and tracks of interest in the Command’s area of responsibility. The web-based configurability of the agents also seemed to hit the mark in terms of the proper level configurability; it provided the flexibility to set configurable parameters but did not require the users to know or understand the underlying implementation details.

With the basic utility of the software proven by Release One, the subsequent release (Release Two) sought to significantly expand the capabilities of the system. LM ATL created new agent templates that automated additional operational tasks within the CNE-C6F operational domain. The development of the new templates was based on feedback from the users of the Release One software and from additional requirement definition sessions held with CNE-C6F personnel. Enhancements were also made to the user interface to improve the configurability of the agents. Release Two was installed at CNE-C6F in July 2006. After the CNE-C6F operators employed the system for thirty days, an assessment was conducted to determine how the system was impacting the CNE-C6F Command. The assessment determined the software was saving approximately 30-60 minutes of investigation time per vessel for each vessel of interest being tracked by each Watchfloor shift. Additionally, the assessment revealed that the FACT software significantly enhanced situation awareness by alerting operators to conditions that otherwise might have gone unnoticed.

Release Three of the FACT software introduced even more functional templates, and “hardened” the software to production grade. In October 2007 the Release Three software was installed at CNE-C6F and also at the NATO CCMAR facility. Upon installation at CCMAR, the software was evaluated during a two week NATO exercise. An out brief of the exercise revealed that the FACT software contributed to a two orders of magnitude improvement in the number of tracks processed during the exercise operations. At one point during the exercise, the operators were able to effectively share and manage over 170 agents.

LM ATL has prepared a fourth release of the FACT software that has not yet been integrated into the FastC^2AP software baseline. Release Four of the software introduces the Google Earth™ FACT Client and the Web Service interface to the FACT system. LM ATL is conducting internal tests and evaluations on the software and is awaiting direction from U.S. Navy organizations as to potential incorporation into the FastC^2AP software suite.