Dynamic Agent Systems in the CoAX Binni 2002 Experiment

Steven Wark, Andrew Zschorn, Don Perugini
Command & Control Division
Defence Science & Technology Organisation
Edinburgh SA, Australia.
{firstname.lastname}@dsto.defence.gov.au

Patrick Beautement
QinetiQ
Malvern, UK.
pbeautement@qinetiq.com

Austin Tate
Artificial Intelligence Applications Institute
The University of Edinburgh
Edinburgh, UK.
a.tate@ed.ac.uk

Jeffrey M. Bradshaw, Niranjan Suri
Institute for Human and Machine Cognition,
University of West Florida,
Pensacola FL, USA.
{jbradshaw, nsuri}@ai.uwf.edu

Abstract - *The goal of the international CoAX (Coalition Agents eXperiment) program was to demonstrate how agent systems could be used to provide agile and flexible command and control systems for coalition operations, and facilitate rapid integration of national C2 systems. The CoAX experiments modelled a coalition C4ISR system as a distributed, heterogeneous agent network using the DARPA CoABS (Control of Agent Based Systems) Grid infrastructure based on Java JINI technology. This paper outlines the CoAX Binni experiment which was held in October 2002 at the US Naval Warfare College, Newport RI. It describes the technology used in this experiment and the role of the ATTITUDE multi-agent architecture in the Australian component of the experiment. This involved logistics planning (and dynamic replanning) for a casualty evacuation from an Australian ship using BDI agents developed in the ATTITUDE architecture, and included interactions with Coalition medical and planning agents. Distributed agents were used to represent the various organisational entities involved in a simplified logistics model, and agent interactions with the Coalition C4ISR system were mediated by human operators using I-X Process Panels. This provided a semi-autonomous system, where human approval initiated further autonomous interactions between Coalition and Australian agents.*

Keywords: Agent systems, data fusion, tracking, command and control, resource management.

1 Introduction

The goal of the international CoAX (Coalition Agents eXperiment) program [1] was to demonstrate how agent systems could be used to provide agile and flexible command and control (C2) systems for coalition operations, and facilitate rapid integration of national C2 systems. To achieve this the CoAX program sought to demonstrate: how flexible, timely interaction between different systems and information sources can be effectively mediated by agents; how policies and domain management helps facilitate selective sharing of information and control of agent behaviour; how ease of composition and dynamic reconfiguration of coalition entities leads to adaptive responses to unexpected events; and how loosely coupled agent architectures can be more efficient and effective than monolithic programs. CoAX was a 3-year experimental program, with demonstrations staged after 1, 6, 9, 21, and 33 months.

The CoAX Binni 2002 demonstration was the culminating demonstration of the program. It was held in October 2002 at the US Naval Warfare College, Newport RI, with some 20 organisations and over 60 agents modelling coalition operations in the fictitious country of Binni [3]. These agents performed diverse roles such as: domain management; situation reporting; medical support; logistics planning; plan deconfliction and optimisation; information source evaluation; dynamic coalition reorganization; sensor tasking; and data fusion.

This paper outlines the CoAX Binni 2002 experiment and the technologies used, and the role of the ATTITUDE multi-agent architecture [5]. ATTITUDE agents perform logistics planning and dynamic replanning for a medical evacuation (medevac) of casualties from an Australian ship. Agents represented the organisational entities in a simplified logistics model, and agent interactions with the coalition C2 system were mediated by human operators using I-X Process Panels [6]. This provided a semi-autonomous system, where a human operator mediated autonomous interactions between coalition and Australian agents.

2 CoAX Binni Scenario

The CoAX Binni 2002 Technology Integration Experiment (TIE) was based on the Binni scenario [3], which was developed as a rich environment to test new concepts in coalition operations and command and control systems, with some changes made to suit the CoAX experiments.

2.1 Background

The scenario is based on fictitious countries in the Red Sea area, and is set in the year 2012. Global climate change has turned deserts into fertile agricultural land, and the region has become a major grain exporter to Europe. The resulting economic and social upheaval has exacerbated historical tribal rivalries, destablising the region.

The major regional powers in the scenario are the (fictitious) countries of Gao, Agadez, and Arabello. A long-standing territorial dispute between Gao and Agadez has resulted in the annexation of Agadez territory by Gao, and the creation of the new state of Binni, with strong cultural, economic, and military ties to Gao. Clashes between military/paramilitary forces continue in Binni, with the potential to disrupt trade – and Europe's food supply. The UN has recognised the democratic government of Binni, and has passed a Security Council Resolution to deploy a UN War Avoidance Force for Binni (UNWAFB), composed of coalition forces from the US, UK, Australia, and Gao.

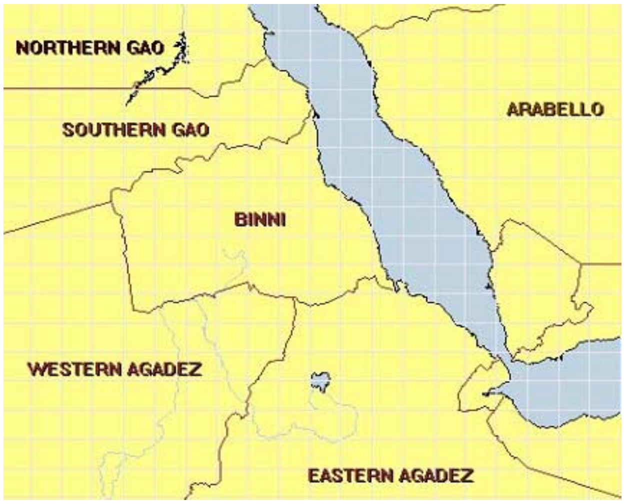


Figure 1: Political boundaries in CoAX Binni scenario.

2.2 Current Crisis

In the vignette used for the CoAX demonstrations, Agadez has recently launched a military offensive into Binni, in the form of major raids causing widespread destruction. These forces are now apparently in rout from a counterattack by Gao forces, but are actually seeking to draw the Gao forces towards a much larger Agadez force.

In the face of this crisis, the UNWAFB is mandated to: stop the Gao forces in the rout of the Agadez forces; stop the Agadez forces from reaching their main force; and negotiate both a settlement between the warring factions, and the dispersal of armed forces and supplies.

2.3 Coalition Operations

In the CoAX demonstrations an agent based C2 system was used to represent the various organisational and functional entities in the UNWAFB coalition C2 structure shown in Figure 2. Agents were used to facilitate intelligence gathering, visualisation, planning and execution from the strategic to the tactical levels of command.

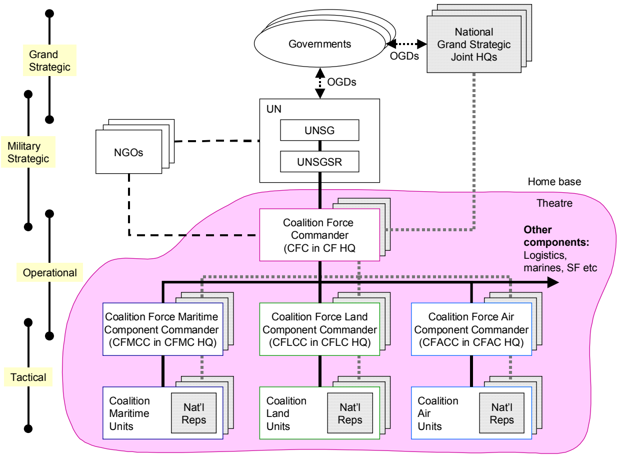

Grand Strategic
Military Strategic
Operational
Tactical
Governments
OGDs
National Grand Strategic Joint HQs
OGDs
UN
UNSG
UNSGSR
NGOs
Home base
Theatre
Coalition Force Commander (CFC in CF HQ
Other components: Logistics, marines, SF etc
Coalition Force Maritime Component Commander (CFMCC in CFMC HQ)
Coalition Force Land Component Commander (CFLCC in CFLC HQ)
Coalition Force Air Component Commander (CFACC in CFAC HQ)
Coalition Maritime Units
Nat'l Reps
Coalition Land Units
Nat'l Reps
Coalition Air Units
Nat'l Reps

Figure 2: UNWAFB C2 structure in CoAX Binni 2002.

In the previous CoAX demonstrations [1], the primary goal of the UNWAFB was to reconnoitre the area of conflict and establish a Total Exclusion Zone (TEZ) between the Gao and Agadez forces, without compromising other UNWAFB objectives. To achieve this the UNWAFB commander decided to perform a 'Firestorm' operation, using an air strike to clear a 5km x 100km corridor between the opposing forces as a TEZ (see Figure 3).

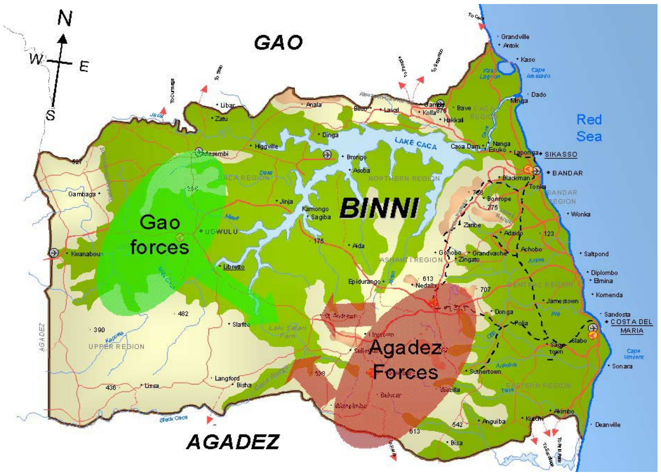


Figure 3: Location of main Gao and Agadez forces

The previous CoAX demonstrations showed how agent systems could be used to successfully plan a complex military operation such as this, and dynamically detect, respond to, and thwart emergent threats to successfully execute the Firestorm operation.

2.4 CoAX Binni 2002

The CoAX Binni 2002 demonstration is based on events unfolding in the Red Sea immediately after the successful Firestorm operation, as the UNWAFB aircraft are returning to base. UNWAFB maritime forces have been active in the area, monitoring the Firestorm operation and tracking the activities of the Agadez Navy – in particular two Agadez submarines (see Figure 4). Throughout the crisis, Arabello, a major trading partner with Australia, has remained neutral. It has a sophisticated anti-submarine warfare (ASW) capability deployed in the Red Sea, and is currently monitoring the activity of Agadez and UNWAFB forces.

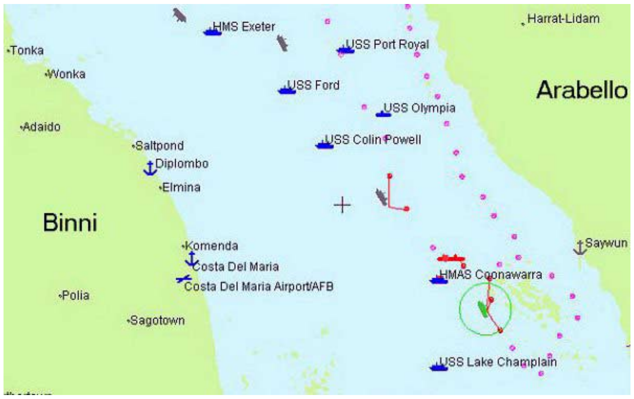


Figure 4: Situation display for CoAX 2002 TIE

An Australian ship, the HMAS Coonawarra, is deployed on ASW patrol in the Red Sea, using conventional sensors and a newly developed high resolution Magnetic Anomaly Detector (MAD) sensor. With the UNWAFB intervention in Agadez's military plans against Gao, Agadez takes a hostile stance, and orders its submarines to attack UNWAFB forces. Taking advantage of favourable environmental conditions, the submarines attack the HMAS Coonawarra, which suffers damage and casualties.

In response, the attack is reported through the UNWAFB C4ISR network, medical support is provided to the ship via telemedicine technology, and a casualty evacuation is dynamically planned and executed. The loss of the sensors onboard the Australian ship creates a gap in UNWAFB ASW capability, and Arabello is identified as a suitable replacement feed. Alarmed by Agadez aggression, Arabello willingly joins the UNWAFB coalition 'on-the-fly' to provide an underwater sensor feed that is shared and fused with UNWAFB ASW resources to enable countermeasures to be deployed. The Agadez submarines are quickly located and neutralised. Realising it is now in an untenable position, Agadez returns to the negotiating table.

3 CoAX Technologies

The CoAX Binni 2002 TIE was aimed at extending the previous CoAX demonstrations to show that the technology could be applied to complex, heterogenous agent systems, and that it allows dynamic formation and integration of agent organisations. To achieve this goal a range of agent and middleware technologies were used in the experiment.

3.1 SimWorld

The CoAX Binni 2002 scenario was modelled in the SimWorld event-driven simulation environment from BBN technologies (see http://www.bbn.com). Agents subscribe to the SimWorld service to receive updates of the scenario world state when they occur.

3.2 CoABS Grid

The DARPA CoABS Grid [3] is middleware that integrates heterogeneous agent-based, object-based, and legacy systems. It includes a method-based API that allows agents and services to register, advertise their capabilities, discover other Grid agents based on their capabilities, and send messages between agents. The Grid also provides: a software foundation to assist in making legacy systems Grid-aware; message encryption; service authentication; administration; and logging and visualisation tools.

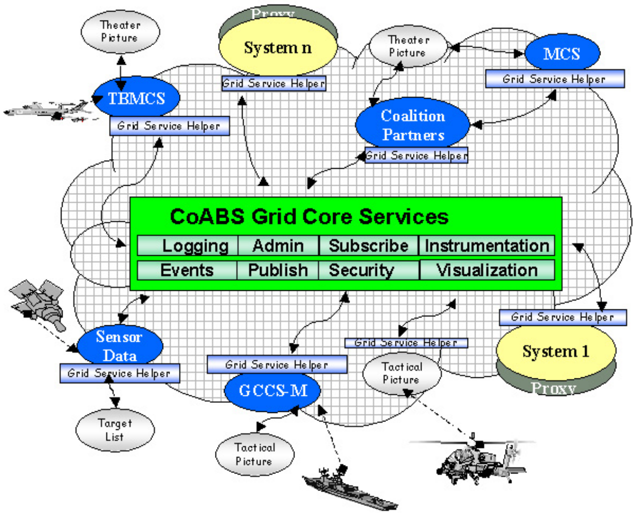


Figure 5: CoABS Grid as enabling infrastructure.

The CoABS Grid was designed to be scalable in terms of the number of agents on it, the number of types of agents, and the number of different actions each agent can perform. All messages between agents are sent point-to-point, and lookup services (LUS) are used for agent registration and discovery. Experiments on the scalability of the Grid [7] have verified that up to 10000 agents can exist simultaneously on the Grid without significantly degrading lookup performance.

The Grid extends Java Jini classes. Agents on the Grid are represented by a Jini *service* object. To register and de-register from the Grid, agents communicate with a Jini LUS, which grants leases and globally unique identifiers to *services*. It propagates this information to all other LUS, so that multiple, equivalent LUS can provide scalability and robustness. By using multi-cast techniques, services require no prior knowledge of LUS host machines.

The Grid provides helper classes that simplify the registration process. Grid agents implement a proxy interface for message delivery. The Grid does not mandate how messages are delivered, but Java RMI is the default. Messages can contain any serialisable data.

3.3 KAoS

The Knowledgeable Agent-oriented System (KAoS) [3] provides management services to ensure that agent systems from diverse sources can be used safely in operational environments. Bounds on agent behaviour are defined by policies expressed in DAML (DARPA Agent Mark-up Language). These are declarative constraints on one or more agents that can regulate registration, access, encryption, resource use, agent mobility, agent obligations, and agent conversations. With the appropriate semantics, agent conversation policies can control the types of, and even the content of, messages exchanged between agents.

KAoS services and tools allow human control of the specification, management, conflict resolution, and enforcement of policies for agents – eliminating the need for the operator to deal directly with DAML. The KAoS domain management services allow agents to be grouped into logical domains that can represent functional, organisational, or administrative teams. Policies can be applied to individual agents, agent domains, or to all agents on the Grid. Agents can belong to more than one domain.

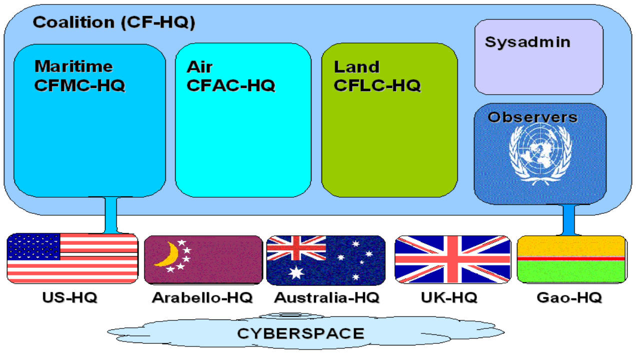


Figure 6: Agent domains in CoAX 2002 TIE

In the CoAX Binni 2002 TIE, agents were grouped into domains to represent national, coalition, and administrative groupings, as shown in Figure 6.

3.4 I-X Process Panels (I-P^2)

I-X Process Panels (I-P^2) [6] were used to support task, process, and event management for the human coalition participants. They provided an effective interface to allow human mediation of the information flow between agents within the C4ISR system, as well as an end-user interface for human decision makers.

I-P^2 acts as a workflow and messaging interface for the user, and can act in conjunction with other panels and users. Messages or tasks can be actioned, forwarded up the C2 chain for action by a higher authority, or delegated. The status of any task can be monitored at all times. Frequently used messages and standard operating procedures can be embedded in an I-P^2 and invoked when appropriate. When a task is actioned, the status is automatically updated in all relevant I-P^{2}s in the action history.

3.5 Agent Systems

The CoAX Binni 2002 TIE involved 20 organisations with over 60 agents, and utilised a range of agent systems. A summary of these systems are given below:

Adaptive Agent Organisations (UT–Austin) attempts to form the best organisation of agents for each specified goal to efficiently respond to environmental dynamics (see http://www.lips.utexas.edu/).

Anaconda (CMU) agent visualiser shows agents registered in the various domains and inter-agent message traffic, allowing the user to monitor the agent environment. (see http://www.cs.cmu.edu/~softagents/).

ATTITUDE (DSTO) multi-agent system allows reasoning with uncertainty and with situations using an extended BDI architecture [5].

D'Agents (Dartmouth College) is a mobile agent system used to support mobile medical monitoring agents (see http://agent.cs.dartmouth.edu/).

Decision Desktop (QinetiQ) uses software agents to acquire and translate data (using Semantic Web technology), and reason with the information. It allows the decision maker to acquire, visualise, and manipulate dynamic information (see http://www.qinetiq.com/).

IMPACT Asset Location Predictor Agent (U. Maryland) can reason with probabilistic temporal information to predict target location (see http://www.cs.umd.edu/projects/impact/).

Information Broker Agent (BBN) maintains a dynamic index of information services provided to the agent community (see http://www.bbn.com/).

Information Trust Evaluator Agent (UT-Austin) assesses the perceived trustworthiness of multiple information sources (based on reputation, uncertainty, and age) and dynamically selects the best information source (see http://www.lips.utexas.edu/).

Interoperable Intelligent Agent Toolkit (LMATL) allows agents to be created graphically and dynamically to rapidly implement modified or new agent behaviours (see http://www.atl.external/lmco/com).

Matchmaker Agents (CMU & BBN) provide dynamic indexing services that contain information (as DAML ontologies) on how to use the data from other agents (see http://www.bbn.com/).

Medical Monitoring Agents (Dartmouth College) are mobile agents that can monitor and interpret remote medical sensors, and generate alerts when appropriate (see http://agent.cs.dartmouth.edu).

Mixed Initiative Agents (BBN) allow human assisted planning and coordination of asset tasking (see http://www.bbn.com/).

Multilevel Coordination Agent (U. Michigan) works top-down with independently developed plans and predicts unintended interactions, identifies candidate resolutions, and presents the ranked candidates for selection. (see http://ai.eecs.umich.edu/people/durfee/COABS/).

NOMADS (UWF/IHMC) is a mobile agent technology that provides the ability to capture the execution state of agents and control the resources they consume. (see http://www.coginst.uwf.edu/nomads/).

NRL DB agent (NRL) receives contact reports and populates a Track Management Database (TDBM) server.

NRL **Viewer (NRL)** displays track information from the TDBM server.

4 ATTITUDE

The agents in the Australian national domain in UNWAFB C2 system were developed in the ATTITUDE multi-agent architecture [5]. ATTITUDE is an extended BDI architecture developed by DSTO that has been designed specifically to support the programming of reactive systems, and for information fusion.

ATTITUDE is so named because it utilises *propositional attitude* expressions as programming instructions, with the form:

<subject> <attitude> <proposition>

For example, when a software agent encounters the instruction **Fred believe (sky is blue)**, it will issue a message to software agent Fred instructing him to believe that (sky is blue). Similarly, when encountering the instruction **I believe (sky is blue)**, it will attempt to believe that (sky is blue). ATTITUDE supports a number of *attitudes* other than believe – such as desire, also desire, expect, anticipate, ask if believe, ask if expect, ask if anticipate.

An important characteristic of attitude programming is that *each propositional attitude instruction either succeeds or fails*, possibly with side effects, depending upon whether the recipient agent is able to satisfy the instructional request. Computational routines for a software agent arise by linking together instructions. The execution path selected through a network of instructions is determined by the successes and failures of the instructions attempted along the way. The control structure is therefore governed by a *semantics of success.*

ATTITUDE beliefs can include Horn clause rules, and it has an inference engine that allows reasoning with these beliefs in an agent's knowledge base. This has also been extended to include a Bayesian inference engine [8]. An agent's knowledge base in ATTITUDE can be partitioned into *events*, which represent a collection of beliefs about the world over a bounded region of time and space. Events can be 'clipped together' using Boolean operators to represent a *situation* in the world, and the inference engine can then be applied to this situation. This allows ATTITUDE to apply contextual and 'what if' reasoning with its beliefs.

ATTITUDE can dynamically form groups of agents and use the group as the *subject* for its proposition attitude instructions. This allows team behaviour to be captured, and also allows ATTITUDE to perform inferences across the beliefs of all members a group, so that as an ensemble they can reach conclusions that no agent individually could reach.

4.1 Architecture

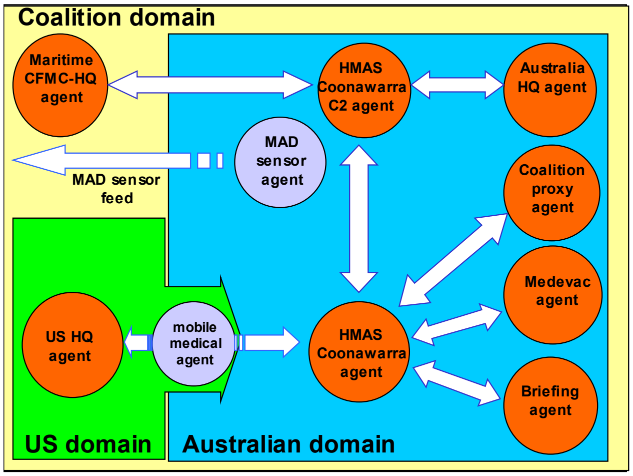


Figure 7: Australian agents in CoAX 2002 TIE.

The Australian national domain of the CoAX Binni 2002 demonstration was modelled around a central agent, the *HMAS Coonawarra agent* (see Figure 7), to facilitate development and testing away from the CoAX demonstration venue. In this model of the C2 system, messages between the Coalition and Australian domain are routed through the HMAS Coonawarra C2 agent, which is an I-P^2 agent that allows the commander of the HMAS Coonawarra to accept, reject, or modify the messages from the agents in the Australian domain. These messages may be forwarded to the Australia HQ agent, which is another I-P^2 agent that represents the Australian national HQ, or the Maritime CFMC-HQ agent (see Figure 2 for organisational role of CFMC-HQ), which is an I-P^2 agent representing the UNWAFB Maritime HQ.

4.1.1 HMAS Coonawarra Agent

The *HMAS Coonawarra agent* models an executive subsystem for the ship's onboard systems. It is the central component to the agent infrastructure in the Australian domain and communicates with other agents and the commander. Information on the ship status and responses to events are modelled in this agent.

The HMAS Coonawarra agent also manages other agents in the domain. For example, it invokes a mobile agent infrastructure when it requests medical support, and when the MAD sensor (which uses a mobile proxy agent to provide a sensor feed to the coalition) is operational.

4.1.2 Coalition Proxy Agent

The *coalition proxy agent* was used for convenience (and to facilitate testing) to represent an off-board logistics agent with up-to-date information on the status of coalition medevac resources.

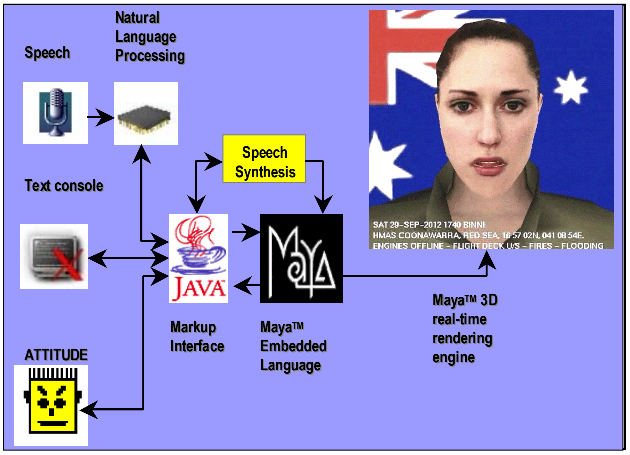
4.1.3 Medevac Agent

The *medevac agent* was used to represent a logistics subsystem onboard the HMAS Coonawarra. It forms a logistics plan to evacuate injured personnel from the HMAS Coonawarra to a nearby medical facility, using available onboard and coalition resources. To form the plan, the medevac agent cooperates with coalition agents in the CoAX environment that can provide transport (helicopter) resources. When the HMAS Coonawarra agent detects injured personnel, it invokes a goal for the medevac agent to determine if a medical evacuation of injured personnel is required. If so, the medevac agent will query the HMAS Coonawarra agent and the coalition proxy agent to discover available medical facilities and transport resources. The proxy agent will respond with the (suitable) available medical facilities and helicopters, providing distances to facilities, start location, earliest start time, and types of the helicopters available. The medevac agent uses prior knowledge of the carrying capacity and speed of the types of helicopters. Helicopter availability depends on the medical priority of the injured.

A simple algorithm is used by the medevac agent to form a logistics plan. The helicopter that can transport the injured to the closest suitable medical facility at the earliest time is selected to perform the transportation task, and higher priority injured personnel are transported first. If the selected helicopter cannot transport all the injured, the selection process is continued with the remaining injured. The plan considers the fact that the HMAS Coonawarra's flight deck may not be operational, or may be operational after an estimated time for repair. Once formed, the logistics plan is sent to the HMAS Coonawarra agent, which forwards it to the HMAS Coonawarra C2 agent (and I-X panel) for human action.

If the situation onboard changes, the HMAS Coonawarra agent sends updates to the medevac agent, which responds accordingly by reforming the plan if necessary. Depending on the changes that have occurred, this may involve re-querying the medical facilities and transport resource proxies.

4.1.4 Briefing Agent



Speech
Natural Language Processing
Speech Synthesis
Text console
JAVA
Markup Interface
Maya™ Embedded Language
Maya™ 3D real-time rendering engine
ATTITUDE
SAT 29-SEP-2012 1740 BINNI
HMAS COONAWARRA, RED SEA, 16 57 02N, 041 08 54E.
ENGINES OFFLINE – FLIGHT DECK U/S – FIRES – FLOODING

Figure 8: Architecture of animated briefing agent.

The *briefing agent* represents an animated agent that is invoked by the HMAS Coonawarra agent to alert (and brief) the ship's crew on the ship's status, as detected by internal systems. For the purposes of the CoAX Binni 2002 TIE, the briefing agent simply displayed a pre-rendered movie of an animated agent, although the animation itself was generated in real time from a situation report in the Future Operations Centre Analysis Laboratory (FOCAL) at DSTO Edinburgh (South Australia). In principle, the HMAS Coonawarra agent could send a situation report to the briefing agent, and the briefing agent would then generate, in real time, the animated character. This complexity was not deemed necessary for the CoAX 2002 demonstration.

The architecture of the system [9] used to generate the animated character in FOCAL is shown in Figure 8. The system is designed to allow automatic input from an agent, human input from a console, or conversational input from a speech recognition system. Marked up text with emotional and gesture tags is sent to an interpreter application that invokes the text-to-speech (TTS) system to generate speech, and synchronises this with real-time facial animation software rendering photo-realistic models.

5 CoAX TIE

The CoAX Binni 2002 TIE was performed at the US Naval Warfare College, Newport RI, in October 2002, using 12 laptop computers connected over an isolated LAN. The demonstration had 6 main parts, as described below.

5.1 Part 1: Attack on HMAS Coonawarra

The submarine attack on the HMAS Coonawarra triggers a simple damage model in the HMAS Coonawarra agent, changing the ship status – engines are offline and the flight deck is damaged. This in turn triggers a series of automated alert messages sent to the HMAS Coonawarra C2 panel, the Australia HQ panel, and the Maritime CFMC-HQ panel. A more detailed status message is sent to the briefing agent, which briefs the crew of the HMAS Coonawarra via an embodied agent.

Status updates are sent by the HMAS Coonawarra agent to the HMAS Coonawarra C2 panel for human action. The commander assesses the situation and uses the C2 panel to report to the Maritime CFMC-HQ panel and to send a request for additional medical support to monitor the casualties in sick bay. The report arrives at the Maritime CFMC-HQ panel and the request is delegated to the nearest ship with suitable medical facilities – the USS Colin Powell. Confirmation of the various tasking actions is automatically propagated back through the C2 chain.

5.2 Part 2: Medevac planning

Having received the request for medical aid, medics decide to deploy a mobile medical monitoring agent to the HMAS Coonawarra. The UNWAFB HQ and HMAS Coonawarra are using different mobile agent infrastructures – NOMADS and D'Agents. KAoS policy management tools provide interoperability by allowing the mobile agent to be dynamically reconfigured and moved between infrastructures. Once in place the mobile agent continually monitors the status of casualties in sick bay and forwards the results to the HMAS Coonawarra agent and to medics onboard the USS Colin Powell.

Once the situation onboard has stablised, the HMAS Coonawarra agent invokes contingency planning for a medevac operation via the medevac agent, and forwards the resulting plan to the HMAS Coonawarra C2 plan for action. The casualties are stable and can be treated onboard, so a low priority operation is planned that waits for the damaged flight deck to be repaired as this is more resource efficient. However, the mobile medical monitoring agent detects the sudden worsening of a patient to a critical condition. It alerts the medics onboard the USS Colin Powell so that they can provide immediate aid, and notifies the HMAS Coonawarra agent of the change in status. This triggers the medevac agent to replan, updating it's contingency plan to produce a high priority medevac plan. The new plan is forwarded through the HMAS Coonawarra C2 panel to the UNWAFB's agent-based deconfliction/optimisation service (the Multi-level Coordination Agent), which takes into account current UNWAFB activities and returns a simplified plan. This plan is executed and the critical patient is safely airlifted to the USS Colin Powell.

5.3 Part 3: Arabello joins the coalition

The crisis on the HMAS Coonawarra has disabled the MAD sensor onboard. The UNWAFB task an Adaptive Agent Organisations tool to assess and prioritise potential alternative sensors to track the Agadez submarines. This identifies the Arabello ASW network as the most suitable replacement.

Arabello willingly join the coalition, and are provided with a 'Coalition Starter Pack', which is a set of tools and documentation to allow Arabello to rapidly (notionally overnight) achieve interoperability with the UNWAFB. The Arabello HQ domain is added to the Grid, and the Arabello system administrators use the KAoS Policy Administration Tool (KPAT) to set up agent policies that provide a secure environment to share selected information with the UNWAFB (and vice versa).

The Arabello commander tasks an agent to provide the selected sensor data to the UNWAFB. The agent registers with an Information Broker agent so that other agents know the data is available. It also registers its subscription capability with the coalition Matchmaker agent so that other agents know how to request data dynamically.

5.4 Part 4: Fusion of Arabello ASW data

Now that select Arabello sensor data is available to the UNWAFB, the US HQ is tasked with fusing it with existing coalition information. The coalition system administrators use the Interoperable Intelligent Agent Toolkit (I2AT) to create a mediator agent for the Arabello data using the information provided by the Matchmaker and Information Broker agents. The I2AT allows non-programmers to create and integrate a new agent to access the Arabello data in minutes rather than months.

The Information Trust Evaluator agent fuses the data from the Arabello mediator agent with other sensor feeds and publishes the fused data. The IMPACT Asset Location Predictor agent uses this information to predict the probable future location(s) of the Agadez submarines, and this is made available for display by the Decision Desktop, and to other agents such as the NRL DB agent and NRL Viewer.

5.5 Part 5: Countermeasures are deployed

Once the Agadez submarines have been located, Arabello is tasked to create a picket-line to box them in. The Arabello commander uses Mixed Initiative agents to generate a tasking plan for his ships.

The damage on the HMAS Coonawarra has now been repaired, and its MAD sensor is back online. To assist the Arabello ships, a sensor feed from the HMAS Coonawarra is made available to them. The Australians do not wish to disclose the exact capabilities of the MAD sensor, and so KAoS policies are defined by the UNWAFB system administrators that dynamically reduce the resolution of the feed published to Arabello by a proxy agent.

5.6 Part 6: Back to the peace table

The UNWAFB and Arabello forces successfully box in the Agadez submarines. Realising its tactical goals are compromised, and the tide of international opinion is against it, Agadez aborts its military offensive and returns to the peace talks.

6 Conclusions

The CoAX Binni 2002 demonstration has shown how an agent-enabled infrastructure significantly aids the construction and use of a coalition command support system capable of dynamic command and control. It showed how: interoperability between different systems is achieved by agent-mediated interactions; policies and domain management can facilitate selective sharing of information; ease of composition and dynamic configurability leads to effective responses to change and unexpected events; and how loosely coupled agent systems that expose behaviours to the community can be efficient, effective, and adaptable.

Agents deployed in the demonstration included 'embodied' interface agents interacting with humans, information agents manipulating, transforming, and fusing information, and 'housekeeping' agents administrating the agent interactions using goals set by human operators.

In particular, the Australian component of the demonstration showed how agents can automatically: monitor, report on, and react to changes in system status; gather information to form contingency plans in response; and dynamically update plans as a situation evolves. It illustrated how embodied agents can be employed to interface with human operators. Finally, it demonstrated how a 'non-Grid-aware' agent system such as ATTITUDE can be easily deployed on the CoABS Grid, and integrated with the KAoS domain management system to provide policy-controlled interoperability with a heterogenous mix of systems.

7 Acknowledgements

The authors wish to acknowledge the other participants, and participating organisations, in the CoAX 2002 demonstration, including the AFRL, ARL, BBN, Boeing, CMU, DARPA, Dartmouth, DRDC, DSTL, GITI, ISX, LMATL, The Mitre Corporation, MIT Sloan, NRL, NWDC, OBJS, The Potomac Institute, Schafer, Stanford, TTCP, UMD, U.Michigan, UT-Austin, USC/ISI, US PACOM.

Steven Wark, Andrew Zschorn, and Don Perugini also wish to thank Dr Dale Lambert for supporting this work under the FOCAL task, and providing valuable advice and feedback throughout its implementation.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of other parties.

References

[1] D.N. Allsopp et al., *Coalition Agents Experiment: Multiagent Cooperation In International Coalitions*. IEEE Intelligent Systems, May/June 2002 pp.26-35. (See also http://www.aiai.ed.ac.uk/project/coax/)

[2] B.P. Kettler (ed.), *The CoABS Grid: Technical Vision*, http://coabs.globalinfotek.com/

[3] J.M. Bradshaw et al., *Making agents acceptable to people*. In N. Zhong & J. Liu (Ed.), *Handbook of Intelligent Information Technology*. Amsterdam, Holland: IOS Press, 2003.

[4] R.A. Rathmell, *A Coalition Force Scenario: 'Binni – Gateway to the Golden Bowl of Africa'*. Proc. Int'l Workshop Knowledge Based Planning for Coalition Forces. Artificial Intelligence Applications Institute, University of Edinburgh, Scotland. 1999 pp.115-225. (See also http://binni.org/).

[5] D.A. Lambert, *Advisers With Attitude For Situation Awareness*, Proc. 1999 Workshop on Defence Applications of Signal Processing, 1999, La Salle, Illinois.

[6] A. Tate, J. Dalton, and J. Stader, *I-P^2 – Intelligent Process Panels to Support Coalition Operations*, Proc. 2nd Int'l Conf. Knowledge Systems for Coalition Operations (KSCO-2002), pp. 84-190, Toulouse, France, April 2002. (See also http://www.aiai.ed.ac.uk/project/ix/)

[7] M.L. Kahn and C. Della Torre Cicalese, *CoABS Grid Scalability Experiments*, Autonomous Agents 2001 Conference, 2nd Int'l Workshop on Infrastructure for Agents, MAS, and Scalable MAS, May 29, 2001.

[8] I. Fabian and D.A. Lambert, *First-Order Bayesian Reasoning*, Proc. Artificial Intelligence (AI'98), Brisbane Australia, 1998.

[9] P. Taplin et al, *Situation Awareness Using a Virtual Adviser*, Talking Head Workshop, OZCHI 2001, Fremantle, WA.