The coalition agents experiment: network-enabled coalition operations

David Allsopp, Patrick Beautement and Michael Kirton
Distributed Technology Group, QinetiQ Ltd

Jeffrey M Bradshaw and Niranjan Suri
Institute for Human and Machine Cognition, University of West Florida

Austin Tate
Artificial Intelligence Applications Institute, University of Edinburgh

Mark Burstein
BBN Technologies

Abstract
Multinational coalitions are increasingly important in military operations. But coalitions today suffer from heterogeneous command systems, labour-intensive information collection and coordination, and different and incompatible ways of representing information. The purpose of Network Enabled Capability (NEC) is to enhance military capability by exploiting information better. The Coalition Agents Experiment (CoAX) was an international collaborative research effort to examine how the emerging technologies of software agents and the semantic web could help to construct coherent command support systems for coalition operations. Technology demonstrations based on a realistic coalition scenario showed how agents and associated technologies facilitated run-time interoperability across the coalition, responded well to unexpected battlespace events, and aided the selective sharing of information between coalition partners. We describe the CoAX experiments, the approaches and technologies used, and highlight how they support the NEC concept. CoAX produced a prototype "Coalition agents starter pack" that could be developed further to support coalition warfare.

1 INTRODUCTION

Military context

Success in military operations calls for high-tempo, coherent, decisive actions (faster than an opponent can react) resulting in decision dominance through the use of command agility – the flexibility and adaptability to grasp fleeting opportunities. To achieve this, the commander must issue clear intent and then delegate the control authority to subordinates, allowing them the scope to exercise initiative. It also means being innovative, creative and unpredictable, to increase confusion in the mind of an opponent. This process is command led, which means that human decision-making is primary and the role of technology secondary. Shared understanding and information superiority are key enablers in this process and are fundamental to initiatives such as NEC. Indeed, the aim of NEC is to enhance military capability through the better exploitation of information.

The current reality of coalition operations is often a picture of data overload and information starvation, labour-intensive collection and coordination, individual stovepipe systems, incompatible formats, scattered snapshots of the battlespace and a horrendous technical integration task. This paper aims to show that the agent-based computing paradigm offers a promising new approach to dealing with such issues by embracing the open, heterogeneous, diverse and dispersed nature of the coalition environment.

Technical approach

Agents are software components that are goal-oriented, active and social [1]. They operate in the digital world and can work on behalf of people to provide the information and services users need [2-4]. The premise of our research is that software agents and associated technologies (discussed further in section 3) provide a powerful conceptual basis for developing large-scale, open, distributed systems for the battlespace in which warfighters and computer systems must work and share information together in a seamless and flexible manner. This will enable warfighters to acquire, visualize and manipulate diverse and dynamic information – however they wish and whenever they need it – putting them in control.

The focus of our research was on creating and demonstrating an agent-enabled infrastructure that would support multinational coalition operations. In addition to the problems of integrating single-service and joint capabilities into a coherent force, the nature of coalition operations implies some need to configure incompatible or foreign systems rapidly into a cohesive whole. Many such problems can only be solved by organizational changes and by aligning doctrine, concepts of operations and procedures. Coalition operations trigger the need for a rapid on-the-fly response and cannot be predicated on using pre-existing coordinated systems – hence the need for a flexible approach that allows capabilities to be assembled at 'run time'. However, in addressing this requirement for interoperability, it is also crucial to tackle issues of security of data, control over semi-trusted software from other coalition partners, and robustness of the resulting system. These were all addressed in our work. Furthermore, throughout this paper, we shall highlight where our research directly supports the following NEC core themes [5]:

- full information availability – enabling a user to search, manipulate and exchange information of different classifications captured by, or available in, all sources internal and external to the battlespace

- shared awareness – providing a shared understanding and interpretation of a situation, the intentions of friendly forces, and potential courses of action amongst all elements in the battlespace

- flexible working – enabling assets to reconfigure rapidly to meet changing mission needs, allowing them to work together with minimum disruption and confusion

- agile mission groups – enabling the dynamic creation and configuration of mission groups that share awareness and that coordinate and employ a wide range of systems for a specific mission

- synchronized effects – achieving overwhelming effects within and between mission groups by coordinating the most appropriate assets available in the battlespace through dynamic distributed planning and execution

- effects based planning – taking an approach to planning that focuses on the use of military and non-military effects against an enemy, and which is integrated with other planning processes in the battlespace

- resilient information infrastructure – ensuring information resources can be managed and that secure access is provided with the flexibility to meet the needs of agile mission groups

- fully networked support – allowing the ready use of non-frontline government bodies, industry, academia and public service capabilities to support operations.

The Coalition Agents Experiment (CoAX)

This international collaborative research programme ran from February 2000 to October 2002 [6]. It involved twenty-six formal partners from the UK, the US and Australia, with support from, among others, TTCP [7] and Defence Research and Development Canada. The CoAX web site maintains an up-to-date listing of participants [8]. QinetiQ researchers were members of the project 'Software Agents in Command Information Systems', which ran from April 1999 to December 2002 and was funded from MoD's Beacon initiative and the CISP (Communications Information and Signal Processing) technology domain of MoD's research programme. The US Defense Advanced Research Projects Agency (DARPA) supported the participants from the US, the University of Edinburgh and QinetiQ through the Control of Agent-Based Systems Programme (CoABS), a multi-million dollar effort that ran from 1997 to 2002 [9]. Australian researchers came from the Defence Science and Technology Organisation in Edinburgh, South Australia.

CoAX was a CoABS technology integration experiment led by a small team of principal investigators from QinetiQ, the Artificial Intelligence Applications Institute (AIAI) at the University of Edinburgh, the Institute for Human and Machine Cognition (IHMC) at the University of West Florida, and BBN Technologies. A series of CoAX demonstrations that showed increasing functionality was carried out between 2000 and 2002 – referred to as CoAX Binni 2000, CoAX Binni 2001 and CoAX Binni 2002. The final demonstration was held over two days in October 2002 at the US Naval Warfare Development Command, Rhode Island, before an invited audience of over a hundred senior officials from the US DoD, US military, US government agencies and UK MoD. This paper focuses on the CoAX Binni 2002 demonstration, though we briefly describe the 2000 and 2001 demonstrations to provide context.

CoAX Aims

The overall goal of CoAX was to show that an agent-enabled infrastructure could significantly aid the construction of a coalition 'command support system' and improve its effectiveness. More specifically, the operational and technical objectives of CoAX were to show how:

a) flexible, timely interaction between different types of potentially incompatible systems and information 'objects' could be effectively mediated by agents, leading to agile command and control, and improved interoperability

b) ease of composition, dynamic reconfiguration and proactive coordination of coalition entities lead to adaptive responses to unexpected events at 'run-time', providing robustness in the face of uncertainty

c) loosely-coupled agent architectures, where behaviours and information are 'exposed' to the community, are more efficient and effective than monolithic programs

d) agent policies and domain management help to facilitate:

  - selective sharing of information between coalition partners, leading to coherent operations

  - control of appropriate agent behaviour, leading to an assured and secure agent computing environment.

2 COALITION SCENARIO AND COMMAND STRUCTURE

Scenario

To create a suitably realistic scenario for the demonstrations, the CoAX team adapted and expanded the fictional Binni scenario [10-11] developed for TTCP [7]. It is set in 2012 on what is currently the Sudanese Plain (figure 1). Global warming has affected agriculture and altered the world's political balance; a previously uninhabited land has become arable and has received considerable foreign investment. It is now called the "Golden Bowl of Africa."

Fig 1. Binni and adjacent countries

A conflict has developed between two countries in the area: Gao to the north and Agadez to the south. Gao has expansionist aspirations but is only moderately developed, possessing old equipment and a mostly agrarian society. Agadez is a relatively well-developed fundamentalist country. Gao has managed to annex an area of land, name it Binni, and establish its own puppet government, which has then come under fierce attack from Agadez. Gao, voicing concerns about weapons of mass destruction, has enlisted UN support to stabilize the region. Arabello is a country on the eastern edge of the Red Sea that becomes involved and eventually provides anti-submarine warfare (ASW) capabilities to the coalition.

Coalition command structure

As the coalition forms, it needs to configure a variety of incompatible stovepiped systems rapidly into a cohesive whole within an open, heterogeneous, dispersed environment. The complexity of this environment is exemplified through the Binni coalition command structure shown in figure 2.

This representative and realistic coalition command structure involves the UN, governments, other government departments (such as the Foreign Office), non-government organizations (such as Oxfam), representatives of all the coalition countries (with their own 'ghosted' command structures, shown as dotted lines), and the coalition headquarters and subordinate fighting forces. The participants would normally agree to the coalition structure when it is formed; no specific country owns any part of the formal command chain, and levels of command overlap, with no rigidly defined boundaries. Dashed lines show an advisory or negotiating role.

From the human perspective, we identified four types of domains (which overlap and are not mutually exclusive) in the Binni coalition:

- organizational, such as the coalition force headquarters
- country, with each national command chain a separate, self-contained domain
- functional, where entities collaborate on common tasks such as meteorology or intelligence
- individual human domains of responsibility, where commanders have responsibility for their own headquarters and all subordinate ones.

3 ENABLING TECHNOLOGIES

We researched and developed a number of emerging technologies, centred around the agent computing model, to facilitate the rapid and seamless sharing of data and information in distributed enterprises. Figure 3 shows how the technologies are linked. Their descriptions follow.

Software agents

Agents can be viewed as semi-autonomous entities that help people to cope with the complexities of working collaboratively in a dispersed information environment [2]. A community of agents works as a set of distributed, asynchronous processes,

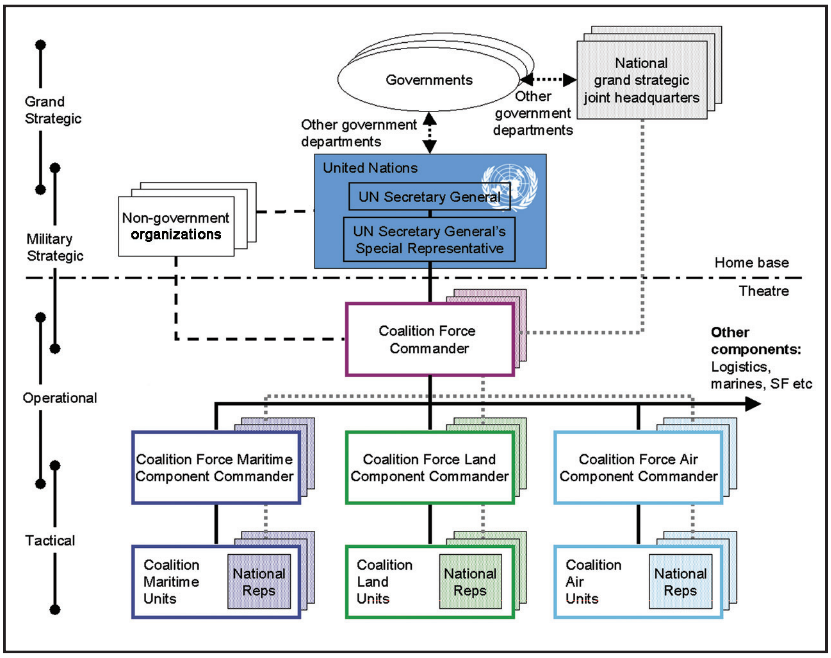

Fig 2. This representative coalition structure shows the chain of command down from the United Nations. The solid black lines show the legal lines of authority (the command chain) and accountability. Dashed lines show an advisory or negotiating role and dotted lines the 'ghosted' command chains of the participating nations. The approximate command levels at which the various entities operate are on the left.

Human-cyberspace interface: interface agents, augmented cognition, ubiquitous computing

Semantic web: facilitates common understanding of terms, relations and services across communities

Agent-based computing: provides glue to link and interoperate disparate systems and applications;
provides services that facilitate interoperability

Grid infrastructure: generic middleware for building distributed applications and creating virtual organizations;
security, dependability, data exchange

Fig 3. Emerging technologies for information sharing in distributed enterprises. The name of each layer is followed by a brief description of its properties.

communicating and sharing information by passing messages in a digital infrastructure. Essentially, agents communicate with users and among themselves to find, format, filter and share information. They work with users to make this information available whenever and wherever they need it, and can be organized to support individuals, military commands and virtual function teams [4]. Agents can also suggest courses of action proactively, monitor mission progress, and recommend plan adjustments as circumstances unfold. Moreover, the agent paradigm provides the modularity and abstraction required for building large, distributed and complex software systems [12].

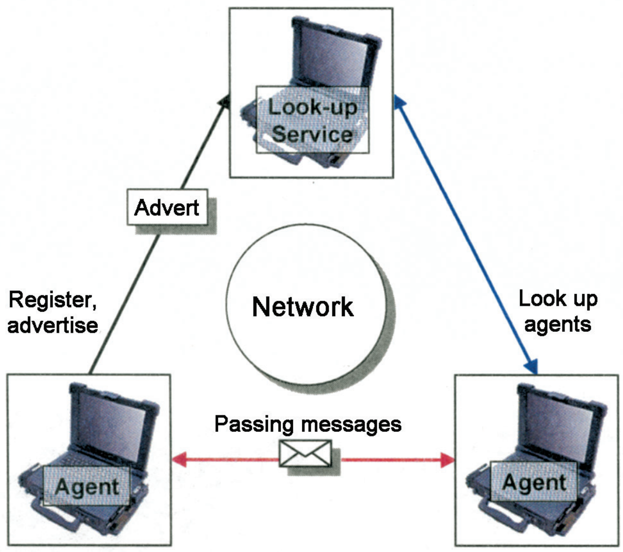

Fig 4. Representation of a network of computers showing multiple inter-operating software agents. Grid software provides look-up services that are used to register and advertise agents and communication services for passing messages between agents. Agents and look-up services can be distributed flexibly across the network, with multiple agents per machine if required.

The CoABS grid

Agents and systems that are to be integrated in a network-enabled environment require an infrastructure for discovering other agents and passing messages between agents. The CoABS grid [9] provided this capability in the series of CoAX experiments (figure 4). The CoABS grid middleware included an interface to register agents, advertise their capabilities, discover agents based on their capabilities, and send messages between agents. It also provided a logging service for both message traffic and other information, a security service to provide authentication, encryption and secure communication, and event notification when agents register, de-register, or change their advertised attributes.

The CoABS grid is based on the Java language and Jini networking technology from Sun Microsystems, making use of two important components of Jini:

- *look-up services*, which are used to register and discover agents and other services. Multiple look-up services can be run for robustness and scalability

- *entries*, which are placed in the look-up services by agents to advertise their capabilities.

Operators or even agents themselves can add or remove agents on the CoABS grid or update their advertisements without network reconfiguration. Agents that fail are automatically purged from the look-up services.

Agent domains and policies

The increased intelligence that software agents provide is both a boon and a danger. Because they operate independently without constant human supervision, agents can perform tasks that would be impracticable or impossible using traditional software applications. However, this autonomy, if unchecked, could also severely impair military operations if defective or malicious agents were to arise.

In CoAX, the Knowledgeable Agent-Oriented System (KAoS) provided services to assure that agents from different developers and running on diverse platforms always operated within the bounds of established policies, and were continually responsive to human control to permit safe deployment in operational settings [13-15]. KAoS services and tools permitted policy management within the specific contexts established by complex military organizational structures.

KAoS policy and domain management services organized agents into logical groups corresponding to organizational structures, administrative groups and task-oriented teams. Within CoAX, these domains mirrored the human domains described in section 2, allowing for complex hierarchical and overlapping structures. An agent domain consisted of a domain manager component and any agents registered to it. The domain manager managed agent registration and served as a point of administration for the specification, analysis and conflict resolution, distribution and enforcement of policies, represented in DARPA Agent Mark-up Language ontologies (see below). Figure 5 shows a typical domain configuration built on the CoABS grid and domain management services of KAoS.

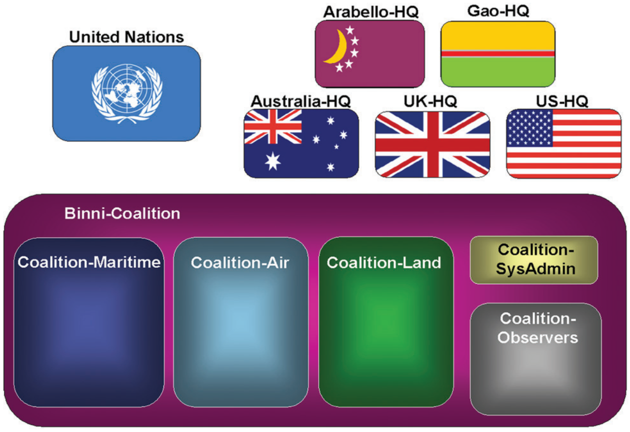

Fig 5. Domain structure used in the CoAX Binni 2002 demonstration. Rounded rectangles indicate domains; each domain would contain a variety of agents whose activities would be governed by a domain manager and matchmaker agent (omitted for clarity). Domain nesting indicates a hierarchy of responsibility and control.

Nomads, which consists of Aroma, an enhanced Java-compatible virtual machine, with its Oasis agent-execution environment, was used in conjunction with KAoS to enforce fine-grained resource control, and information filtering and transformation policies.

Semantic web

Currently, web pages are geared towards visual presentation of information for humans with no support for machine understanding, severely limiting the automated processing of the huge volumes of information on the web. In this context, the semantic web is a vision: the idea is to have data on the web defined and linked such that it can be used by machines not just for display purposes but for automation, integration, inference and reuse of data across various applications [16,17]. Clearly, to turn these ambitions into reality requires the development of new technologies, tools and methodologies. The semantic web model uses Uniform Resource Identifiers (URIs) to identify resources (electronic images, documents, services; web page addresses such as http://www.QinetiQ.com are a type of URI). The Extensible Mark-up Language (XML) is a meta-language that provides a flexible, extensible common text format for data exchange. Schemas and ontologies provide a means of describing the meaning of terms in a domain. In the semantic web, these are based on, for example, the Resource Description Framework and the DARPA Agent Mark-up Language (DAML) [18].

In the CoAX demonstrations, XML was one of the languages used for inter-agent messaging, and DAML was used to encode and reason about domain entities, domain policies and agent message contents. Semantic web ontology-based tools, such as the Decision Desktop (section 4, figure 10), were used for coalition-wide information gathering and visualization.

4 DEMONSTRATION STORY-BOARDS AND TECHNOLOGIES

The CoAX demonstrations were built around story-boards that described a set of events that were realistic in military terms. These are described next.

CoAX Binni 2000 demonstration: information gathering phase

The events of the CoAX Binni 2000 demonstration focused on the initial planning phase of conflict [6]. A number of options to separate the opposing forces and restore peace in the region, including the deployment of a large ground-observation and peace-enforcement force, had been rejected and a 'Firestorm' mission was chosen. The aim was to clear land and keep belligerent forces apart to facilitate simpler remote and ground observations with less risk to the coalition peacekeepers. The demonstration started by showing how the coalition used agents to gather initial information from among the partners. This provided coalition-wide shared awareness. During the course of events, it became clear that Gao was feeding misinformation, and special system administration steps were taken to monitor the information passed to and from Gao within the coalition (figure 6). Later, Gao became belligerent and launched a denial-of-service attack against the coalition's C4I infrastructure. This was automatically detected and thwarted using the advanced KAoS policy administration capabilities available to the coalition, coupled with fine-grained resource control available in Nomads [14,15].

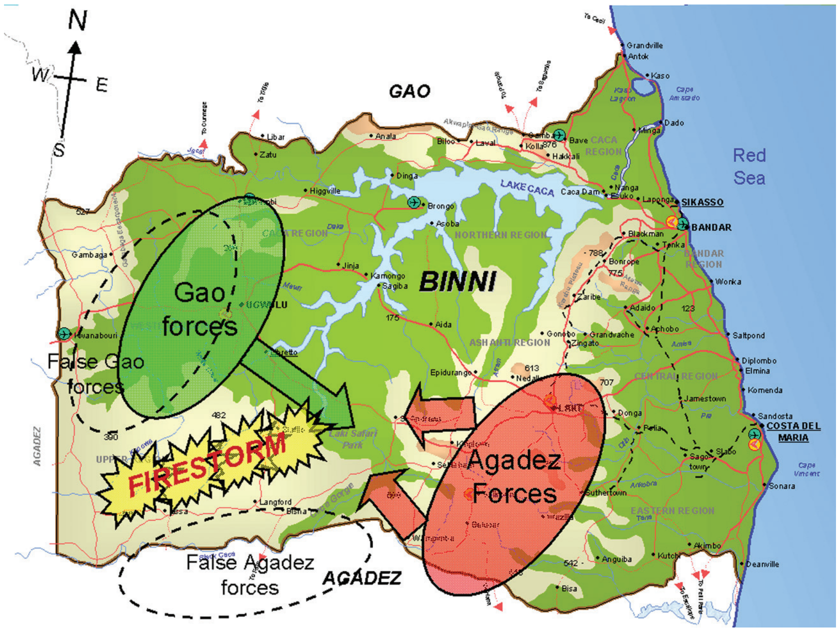

Fig 6. Map of Binni showing firestorm deception. Misinformation from Gao is intended to displace the Firestorm to the west, allowing Gao and Agadez forces to clash in the region of the Laki Safari Park.

Overall, the demonstration showed

- the grouping of agents into policy-governed domains
- the linking of agents and 'agent wrapping' of legacy military systems such as the UK's Master Battle Planner, enabling it to receive dynamic updates (figure 7)
- the extraction and import of publicly available data on the web
- the detection and control of hostile agents
- visualization of the current state of operations via I-X (Intelligent Technology Project) Process and Event Panels [19]
- support for coalition shared awareness.

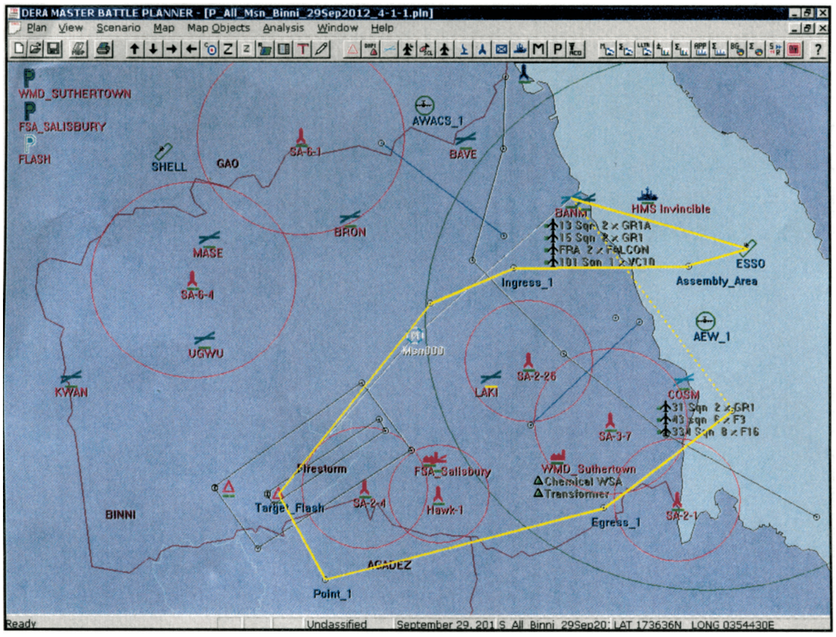

Fig 7. Master Battle Planner map display of Binni, Gao and Agadez showing information gathered from an agent-enabled coalition infrastructure. A selected mission is highlighted, proceeding from an airbase (BANM) to refuelling tanker (ESSO) to the target via waypoints and airspaces, and back to base by a different route.

CoAX Binni 2001 demonstration: dynamic execution phase

The events of the CoAX Binni 2001 demonstration moved on from the initial planning and information gathering phase to a specific day and time in the execution phase, involving the monitoring, battle management and short-notice re-planning associated with coalition operations [6].

The Firestorm mission was planned in detail and aircraft were prepared for their missions. However, the news media broke a story that wildlife in an important safari park in Binni might be in danger as the park overlapped the Firestorm area. With only an hour to go, the UN Secretary General's Special Representative to Binni asked the Coalition Force Commander (CFC) to guarantee that wildlife would not be at risk from the Firestorm operation. Dynamic information gathering and information feeds using agent technology were used in real time to communicate the positions of some of the large mammals at risk. After consideration, it was decided to continue with the Firestorm operation but to re-plan as necessary to avoid risk to wildlife. Firestorm targets were adjusted in time, or secondary targets selected as necessary, for the first wave of firestorm bombing. The impact of these changes on the coalition's medical and humanitarian operations was automatically detected, and unintended conflicts between disjoint coalition operations were avoided. Lastly, Agadez fighters launched high-value asset attacks against the coalition forces; these were detected and important monitoring agents were moved to other computational platforms as the monitoring aircraft regressed.

This demonstration showed newly-arrived agents integrated into domains at short notice, introduced additional time-critical agent functionality such as de-confliction of air task messages and updates exported from master battle planner, run-time re-configuration, and integration of remote, near real-time sensor feeds and unclassified information from the Internet.

CoAX Binni 2002 demonstration: dynamic coalition reconfiguration

The events of Binni 2002 followed those of 2000 and 2001, and began with an attack on an Australian monitoring ship in the Red Sea by two Agadez submarines. The neighbouring country of Arabello (figure 1) was prompted by the attack to offer its ASW capabilities. This offer was quickly accepted and Arabello's sensors were rapidly linked into the coalition's C4I agent framework. Subsequent coalition ASW activities forced Agadez to back down and return to peace talks with Gao at the UN. This scenario is described in more detail next to highlight the key role played by agent technologies.

Submarine attack

Following its unsuccessful fighter attack, Agadez ordered two submarines in the Red Sea to attack an Australian monitoring ship (HMAS Coonawarra). The status of the Coonawarra was monitored by onboard agents, which detected flooding and electrical fire in the engine room and damage to the helideck. They generated an alert, which was sent up the chain of command, in accordance with the agents' standard operating procedures (SOPs) to the Australian and Coalition Force Maritime Component (CFMC) HQs.

On the Coonawarra, the Captain's agent-enabled C4I included a Process and Event-handling Panel (figure 8). I-X Process Panels understand the coalition's organizational structure and can support inter-human and inter-agent messaging in a structured form concerning issues, activities, constraints and reports [19]. They can offer SOPs for

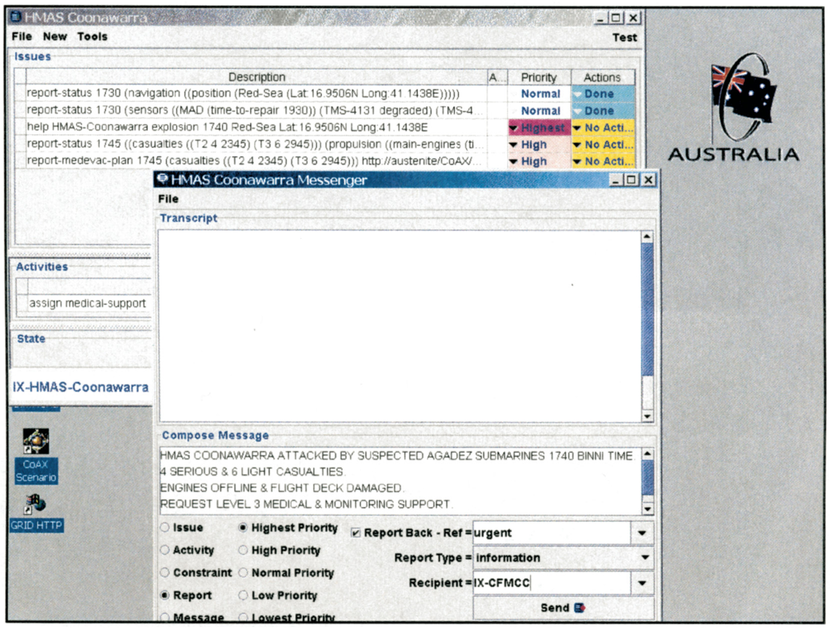

Fig 8. The foreground I-X Process Panel shows a message generated automatically by agents on board the Coonawarra in response to the submarine attack. The panel is aware of the coalition structure and suggests a recipient accordingly. The panel in the background shows ongoing issues being reported to the Captain, for example status reports from agents on the ship.

responding to events or making requests. In this case, the Captain used the panel to report on the attack, the ship's status and the resulting ten casualties. This report was sent automatically to the relevant HQs. Owing to the basic facilities on the ship, the Captain requested minimum level 3 medical support and assistance with medical monitoring.

The report was passed up the command chain to the coalition force (CF) HQ where an event panel was used to delegate the immediate aid and medical assistance tasks to the nearest ship with level 4 medical facilities, the USS C Powell. The Powell acknowledged; this confirmation was sent back through the various panels.

This vignette showed how interface agents working collaboratively across the software agent network reported a submarine attack on an Australian ship to the coalition C4I infrastructure. The agents on the Coonawarra were able to respond to the attack and the damage that was caused by reconfiguring themselves to take account of the information sources that were no longer available. This supports the NEC core themes of full information availability and resilient information infrastructure. Mixed initiative (human-agent interaction) messaging was used to request medical assistance and tasking, and responsibilities were reallocated. This supports the core themes of shared awareness and flexible working.

Casualty information

Australian personnel wore medical tags that monitored their well-being and sent data to a medical database on the ship. To aid the Australians, system administrators at CF HQ were tasked to deploy medical monitoring agents to the Coonawarra. These agents interrogated the medical database on the ship and made the information available in near real-time to the medics on the USS C Powell and at the Australian, CFMC and CF HQs. This was achieved using KAoS policy administration tools, which dynamically reconfigured the agents with new mobility policies and thus permitted them, while still running, to move to where they were needed.

A monitoring agent then reported that one casualty was in crisis. Medics stabilized the critical patient and recommended immediate evacuation to a Level 3 medical facility. The coalition's de-confliction / optimization agent service determined that there was a logistics supply helicopter already *en route* that could also pick up casualties. As a result, the critical patient received attention 30 minutes earlier than would have occurred without this collaborative re-planning.

In this vignette, security permissions were set up and mobile medical monitoring agents were dispatched. Using services defined by the grid mobile agent system, they moved from one type of agent environment to another and still performed as before. The medical evacuation flight was de-conflicted as a result of agent-instigated alerting. The agents de-conflicted and optimized the plan by being able to access and exploit synergies in coalition-wide open information. This supports NEC core themes of flexible working and synchronized effects. The agent behaviour in cyberspace that was triggered by these events was monitored and visualized as part of full-spectrum dominance, which supports the core theme of shared awareness. It is this ability to adapt to the ever-changing realities of conflict at run-time that makes software agents so useful.

Arabello joins coalition

The Coonawarra had novel magnetic anomaly detection equipment, and had been releasing the resulting information to the coalition, but this capability was seriously degraded by the attack. The nearby country of Arabello was identified as a possible ally to fill this information gap. Wishing to support a trading partner under direct attack, and seeing the risk to shipping from Agadez submarine activity, Arabello asked to join the coalition and offered its ASW capability, an underwater sensor grid. The coalition used its agent performance evaluation tool to examine this capability and verify it as suitable.

Coalition system administration staff provided a "Coalition Agents Starter Pack" [20] to Arabello to bring them up to speed on coalition operational and technical matters, and to set up secure, selective information interoperability between the coalition and Arabello. This pack contained scenario information, agent wrappers, process and event panels, policy and domain management capabilities, and set-up and configuration instructions.

To avoid sharing more intelligence than necessary, Arabello created a policy restricting its agents only to provide reports on Agadez submarines, and only to coalition agents. This was an example of restricting communications by message content, rather than just by the domain of the sender and receiver [14,21]. Policy information was represented using DAML, which, combined with KAoS components, provided powerful policy reasoning, de-confliction and enforcement capabilities. Any conflicts between policies and possible resolutions were displayed graphically for system administration staff.

Once interoperability with Arabello was established, a formal tasking was sent from the CFC via the Process Panels, requesting the sensor data. The Arabello agents did not need to know what other agents would require data; they made its availability known via a matchmaker agent so that other agents could find data dynamically. This approach enabled services and capabilities to be advertised and withdrawn as circumstances changed.

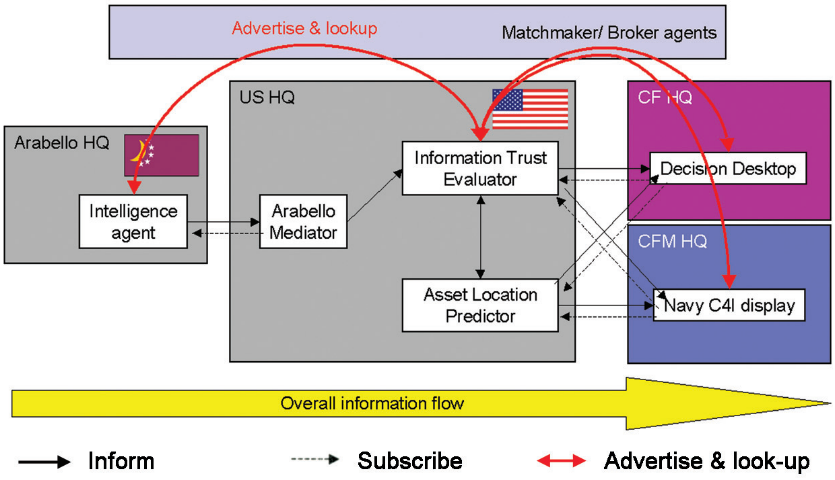

Fig 9. Secure and selective integration of sensor data from a new coalition partner. Data are fused and used to predict future submarine locations, and delivered to C4I display tools. The mediator agent is rapidly generated using an agent toolkit. Links between agents are created dynamically through look-up and advertisement – they are not hardwired.

In this vignette, Arabello joined the coalition after its alternative ASW feeds were validated as suitable. It used the starter pack to make selected parts of its agent-based underwater sensor grid capabilities visible to coalition members as an intelligence service. The service was advertised and used by coalition HQs as required. This supports the core themes of agile mission groups, resilient information infrastructure and fully networked support. This part of the demonstration showed how a completely unexpected, unprepared, partner was integrated into the coalition command structure at short notice.

Agents enable sensor fusion

Next, the CFMC Commander tasked the US HQ to acquire sensor data from Arabello, to translate it, and fuse and collate it with existing coalition information. This was delegated to the system administration staff in the various HQs. They used an agent creation toolkit (Interoperable Intelligent Agent Toolkit [22]) that lets non-programmers compose agent behaviours graphically and dynamically. Without these capabilities, the interface negotiation, code development and system integration could have taken months to achieve the same level of interoperability. An agent was created to act between the Arabello sensor agent and the US fusion agent. This 'mediator' agent could translate between the different forms of XML used to represent sensor data by members of the coalition, and thus pass data from Arabello to the US domain (figure 9).

Agent tools in the US domain provided the fusion service for the coalition. Before the submarine attack, the fusion service had been collating information from satellites (available twice daily), sundry radar returns (frequent but often unreliable), unmanned autonomous image feeds (asynchronous and often unreliable) and from the Australian magnetic anomaly detection (reliable and continuous). After the attack, Arabello provided historical data and an ongoing and moderately reliable feed. An 'information trust evaluator' agent fused sensor reports from the Arabello mediator agent with existing sources, taking into account sensor reliability and trust. The fused sensor data were made available to all agent-enabled C4I tools, such as the CFC's Decision Desktop (figure 10).

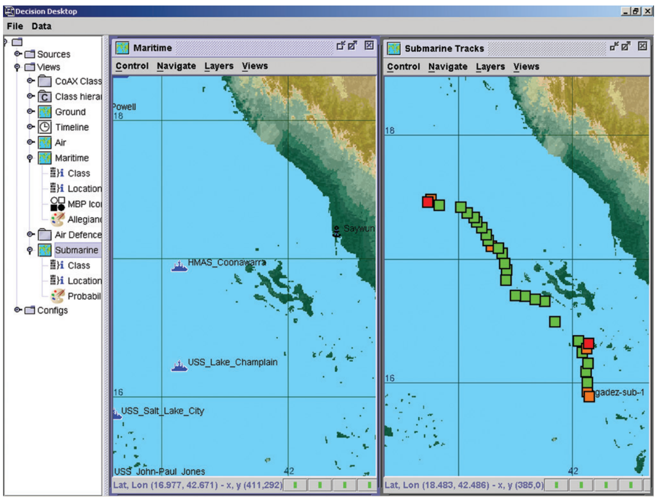

Fig 10. Submarine contacts from Arabello are delivered to the Decision Desktop C4I visualization tool. The panel to the right shows that the commander has chosen to display them according to their confidence levels. These could also be added to the standard maritime display.

Next, the warfighters needed to predict the likely positions of the Agadez submarines to determine their responses. The US possessed several 'asset movement' agents, which could access the coalition-wide fused sensor data, calculate likely predicted locations of the Agadez submarines, and provide output to C4I agent systems such as the CFMC HQ displays.

In this vignette, Arabello's sensor grid information was made available to the coalition by creating a 'go-between' agent that enabled Arabello's intelligence agents to talk to the US fusion service. Arabello's feed was collated with the others available to the coalition, and a trust evaluator agent dynamically selected the best information and forwarded it, re-assessing the value of the feeds, and switching sources appropriately as time passed. This supports the core theme of full information availability.

Next, the Agadez submarines' locations were predicted and then displayed on the various coalition C4I systems. This was achieved by interaction among the heterogeneous agents. Using these techniques, information was published and made available to be picked up by any decision makers, as they demanded it, and displayed on their C4I systems in the form they required. This supports the NEC core theme of shared awareness.

Dissemination of information and countermeasures

Once the positions of the Agadez submarines were predicted, coalition ASW forces had to locate them exactly and box them in with patrol boats and sonobuoys. To help Arabello with this task, the coalition provided a feed from the magnetic anomaly sensor on the Coonawarra, now operational again. However, the Australians did not want to reveal the full capabilities of the sensor to Arabello, so they provided degraded images by setting appropriate agent policies using the KAoS policy administration tool, and Nomads filtering and transformation policy enforcement mechanisms.

The policy dynamically lowered the resolution of the sensor data before sending it to Arabello (figure 11). Other forms of transformation were possible, including introducing a time lag (non real time) or reducing the update rate. Filtering of sensitive data (eg, the location of a US submarine) could also be implemented. Suri *et al* provide more detail on how these capabilities worked in the CoAX context [21].

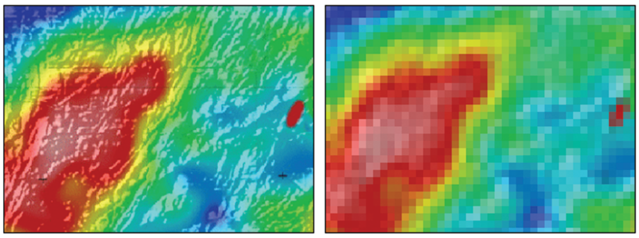

Fig 11. On the left are shown the original sensor data from the Coonawarra's magnetic anomaly detection equipment. On the right are the same data after being dynamically transformed and downgraded for release to the coalition.

The Arabello Maritime Commander used an agent-enabled planning system to help him to gather information, select targets, identify resources to use, communicate with his subordinates and issue orders. This mixed-initiative tool allowed the commander to manipulate the military objects (ships, targets, etc) directly on the screen and arrange assets to achieve synchronized effects. The agents sensed these interactions, and fetched and updated the required information in the background, acting as part of the mission team. Subsequently, because of the coherent activity by the coalition, Agadez returned to the negotiating table.

In this part of the scenario, agents dynamically maintained the interconnection and interoperability between relevant information feeds and a service that output a stream of predicted Agadez submarine positions. Agent policies were used to control the dynamic filtering of information before passing it to Arabello. This supports the core themes of full information availability and resilient information infrastructures. Interface agents supported warfighters as they assembled information to make decisions and deploy countermeasures against the Agadez submarine. This supports the core themes of agile mission teams, effects-based planning and synchronized effects.

5 CONCLUSIONS

Assessment of CoAX

The overall goal of CoAX was to show that an agent-enabled infrastructure significantly aided the construction of a coalition command support system and improved its effectiveness. Referring to the specific operational and technical objectives highlighted in section 1, we deduce that good progress was made towards achieving the aims of CoAX.

Objective a)
There were many examples in the CoAX demonstrations of agents facilitating information sharing between disparate systems. For example, the CoAX 2000 demonstration included the agent-enabled inter-operation of real military systems, namely the Master Battle Planner and the Consolidated Air Mobility Planning System.

Objective b)
Specific instances of adaptive responses at run-time include the re-planning of air missions in CoAX Binni 2001, because of the need to avoid large mammals in the Laki Safari Park, and Arabello joining the coalition in CoAX Binni 2002.

Objective c)
In CoAX Binni 2002, the loosely-coupled agent architecture allowed Arabello's ASW information to be advertised to members of the coalition, who were then able to access it when required.

Objective d)
The provision of 'downgraded' Australian sensor data in CoAX Binni 2002 provided an example of selective information sharing. The KAoS policies and domain structures controlled agent behaviour throughout all the demonstrations, facilitating coalition agent interaction and preventing defective, malicious or poorly-designed agents from impeding coalition objectives.

In the CoAX experiments, running software agents and the CoABS grid software was not found to impose a noticeable overhead on processing or communications. In general, the flexibility of agent-based systems means that they are likely to be able to adapt to varying levels of resource availability, and policies can control their utilization of resources.

Relevance to NEC

There are several reasons why agents can support NEC, making them potentially useful during conflict. First and foremost, because their behaviour is not fixed at 'design-time', they enable military commanders to behave unpredictably – to 'wrong-foot' an opponent. Software agents can be dynamically reconfigured, supporting the NEC core themes of flexible working and agile mission groups.

Secondly, an opponent will impose unpredictable events and outcomes on the battlespace, so it is impossible to plan all requirements in advance. Hence, commanders must be able to adapt to the military imperative at run-time – the command systems must not constrain users' scope of actions. Hence, as software agents can be tasked according to the circumstances, they support the core themes of synchronized effects and effects-based planning.

Next, unlike traditional software that is reactive, agents are capable of being proactive and predictive. Although their autonomy and intelligence give them much more freedom, the actions of the agents and the flow of information among them are kept under strict control by human administrators through policies that are enforced in the domains to which agents belong. Agents can adapt to changing circumstances at run-time, and can use messages and events to act as triggers – hence they are not tightly constrained at design time to what they do. Indeed, their ability to 'self-heal' at run time makes them robust in the face of a real battlespace that is event-driven, high-tempo, short timescale, uncertain, diverse and dynamically varying. Consequently, software agents support the core themes of full information availability and resilient information structure.

Lastly, software agents can work with humans in a so-called mixed initiative manner such that, as the humans click, type and speak, they are triggering agent actions. Agents can sense certain real-world events and report back to the humans. The humans and agents work as a collaborating distributed team [15]. The activities are not determined at 'design-time' (over-engineered and brittle) but are free flowing and natural. In this case, software agents are supporting the core themes of shared awareness and fully networked support.

Strengths and weaknesses of agent technology

The generic main strengths of the agent paradigm are that it offers [23,24]:

- a powerful metaphor for conceptualizing complex systems; it is natural to model complex systems in terms of self-supporting agents that provide services and undertake tasks on behalf of other agents, systems and users.

- distribution of control. Agents support a distributed, heterogeneous model of computing. Agent communication languages provide the means for agents to interoperate in a seamless fashion, irrespective of where they exist in the environment. Real-world problems are overwhelmingly distributed in nature.

- a natural means of exploiting and controlling concurrency. Multi-agent systems comprise asynchronous processes that communicate by passing messages. Agent co-ordination strategies and policies control how agents interact and the actions they are allowed to perform.

- a mechanism for leveraging open systems with heterogeneous computing platforms and disparate programming languages. A key advantage of agent-based computing is the inter-operation of disparate agents and systems.

- the ability to support global services marketplaces. The current trend towards viewing organizations as service providers is likely to become ubiquitous. With this view, software agents will represent the individuals, departments and organizations that provide services.

- a natural computational model for pervasive computing. As the IT world creates environments that are saturated with computing and wireless communications, it is increasingly likely that agents will be seen carrying out functions and providing services on embedded devices and systems.

Despite this promise, however, there are some current weaknesses in implementation [23,24]:

- tool support. Developing distributed computer applications is a highly skilled activity, as is developing multi-agent systems. Currently, there is a marked lack of tools that assist in the development, testing, performance monitoring and debugging of agent applications. Achieving good performance relies on careful design and implementation.

- agent component libraries. One of the distinct advantages of object-oriented development is the availability of high-quality third-party libraries of reusable components. There is now a clear requirement for agent libraries and frameworks that support, for example, intra- and inter-agent communication, planning, knowledge representation, reasoning, negotiation, etc.

- environments. The uptake of agents will require the development of robust, secure and inter-operable run-time environments that provide agents with the 'life support' they require [25]. For example, 'end to end' interoperability requires some form of shared semantics, the ability to deal with multiple agent communication languages and operations across firewalls.

Future directions

Tools such as the prototype Coalition Agents Starter Pack [20], developed for the CoAX 2002 demonstration, could form the basis for future coalition warfare programmes and could be evaluated within the UK's experimental NEC programmes such as the Experimental Network Integration Facility (now re-named NITEworks). Areas requiring further research and evaluation include richer descriptions of agent services and capabilities (based, for example, on semantic web technologies) leading to dynamic, automatic and trusted composition of services across the battlespace, and agent-based architectures that are secure and resilient in the face of physical and information-based attacks.

Acknowledgements
We are greatly indebted to all those who worked on CoAX – the CoAX web site [8] contains a complete listing of all participants. We would also like to thank Dan Daskiewich (Air Force Research Laboratory), Jim Hendler (DARPA and University of Maryland), RAdm Lee Kollmorgen (Naval Research Laboratories), Capt David Marsh (Air Force Research Laboratory), Rick Metzger (Air Force Research Laboratory) and Lt Cdr Dylan Schmorrow (DARPA) who played a key role in instigating and supporting CoAX. QinetiQ work was carried out as part of MoD's research programme and also part funded by DARPA. Apart from DSTO Australia, the other projects that participated in CoAX were sponsored by DARPA and managed by the US Air Force Research Laboratory. DARPA, the US Army, the Office of Naval Research and NASA sponsored Jeffrey Bradshaw's work on KAoS policy-based agent domain services. The US Government, the University of West Florida, the University of Edinburgh and BBN Technologies are authorized to reproduce and distribute reprints of the published article for their purposes notwithstanding any copyright annotation hereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the UK MoD, DARPA, the US Government, QinetiQ, University of Edinburgh, University of West Florida or BBN Technologies.

References

1 Jennings N R, Sycara K and Wooldridge M. *A roadmap of agent research and development.* Autonomous Agents and Multi-Agent Systems **1** 275 (1998).

2 Bradshaw J (editor). *Software Agents*. AAAI Press / MIT Press, Cambridge, Massachusetts (1997).

3 Bradshaw J (editor). *Handbook of Agent Technology*. AAAI/MIT Press, to be published (2003).

4 Bradshaw J M *et al* (editors). *Software Agents for the Warfighter.* ITAC Consortium Report. Cambridge, Massachusetts, AAAI Press/The MIT Press. (2002).

5 Networked Enabled Capability flyer, version 5.0, 18 October 2002.

6 Allsopp D N *et al*. *Coalition Agents Experiment: Multiagent Cooperation in International Coalitions,* IEEE Intelligent Systems May/June **17:3** 26 (2002).

7 TTCP web site: http://www.dtic.mil/ttcp.

8 Coalition Agents Experiment web site: http://www.aiai.ed.ac.uk/project/coax.

9 Control of Agent-Based Systems web site: http://coabs.globalinfotek.com.

10 Rathmell R A. *A Coalition force scenario 'Binni – gateway to the Golden Bowl of Africa'*. In Proceedings of the International Workshop on Knowledge-Based Planning for Coalition Forces (editor A Tate) pages 115-125, Edinburgh, Scotland, 10th –11th May 1999. Available at: http://binni.org.

11 Binni web site: http://binni.org.

12 Jennings N R. *An agent-based approach for building complex software systems.* Communications of the ACM **44:4** 35 (2001).

13 Bradshaw J M *et al*. *KAoS: towards an industrial-strength open agent architecture.* In Software Agents, AAAI Press / MIT Press, Cambridge, Massachusetts (editor J M Bradshaw) 375-418 (1997).

14 Bradshaw J M *et al*. *Representation and reasoning for DAML-based policy and domain services in KAoS and Nomads*. Proceedings of the Autonomous Agents and Multi-Agent Systems Conference (AAMAS 2003). Melbourne, Australia, ACM Press, New York (2003).

15 Bradshaw J M *et al*. *Making agents acceptable to people.* In Handbook of Intelligent Information Technology. IOS Press, Amsterdam, Holland (editors N Zhong and J Liu) (2003).

16 Semantic web activity statement web site: http://www.w3.org/2001/sw/Activity.

17 Berners-Lee T, Hendler J and Lassila O. *The Semantic Web*. Scientific American (2001) available at: http://www.sciam.com/2001/0501issue/0501bernerslee.html.

18 The DARPA Agent Mark-up Language web site: http://www.daml.org.

19 Tate A, Dalton J and Stader J. *I-P2: Intelligent Process Panels to Support Coalition Operations*. In Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations (KSCO-2002), Toulouse, France, 23-24 April 2002. Available at:http://i-x.info/documents/2002/2002-ksco-ip2.pdf

20 Coalition Agents Starter Pack available at: http://www.aiai.ed.ac.uk/project/coax/demo/2002/coalition-starter-pack.html.

21 Suri N *et al*. *DAML-based policy enforcement for semantic data transformation and filtering in multi-agent systems.* In Proceedings of the Autonomous Agents and

Multi-Agent Systems Conference (AAMAS 2003), Melbourne, Australia. ACM Press, New York (2003).

22 Interoperable Intelligent Agent Toolkit web site: http://www.atl.external.lmco.com/overview/programs/IS/I2AT.html

23 Kirton M J and Jennings N R. *Software agents and interoperability.* QinetiQ/KIS/SEB/CR020183/1.0 (2002).

24 Millington P. *Exploitation of agent technology.* Logica report, reference EC26904:3.1 (2002).

25 Bradshaw J *et al*. *Terraforming cyberspace*. IEEE Computer **34:7** 48 (2001).

Glossary

AIAI	Artificial Intelligence Applications Institute (at the University of Edinburgh)
ASW	Anti-Submarine Warfare
CF	Coalition Force
CFC	Coalition Force Commander
CFMC	Coalition Force Maritime Component
CISP	Communications Information and Signal Processing
CoAX	Coalition Agents Experiment
CoABS	Control of Agent Based Systems
C4I	Command, Control, Communications, Computing and Intelligence
DARPA	Defense Advanced Research Projects Agency
IHMC	Institute for Human and Machine Cognition (at the University of West Florida)
I-X	Intelligent Technology Project
KAoS	Knowledgeable Agent-Oriented System
NEC	Network Enabled Capability
SOP	Standard Operating Procedure
URI	Uniform Resource Identifier
XML	Extensible Mark-up Language

QinetiQ Ltd, Malvern Technology Centre, Malvern, Worcs WR14 3PS
Kirton@signal.QinetiQ.com