SOFTWARE AGENTS AS FACILITATORS OF COHERENT COALITION OPERATIONS

Dr David Allsopp (DERA, UK).
(dallsopp@signal.dera.gov.uk E211, DERA Malvern, Worcestershire, WR14 3PS, UK).
Squadron Leader Patrick Beautement (DERA, UK) *.
(PBeautement@dera.gov.uk E109, DERA Malvern, Worcestershire, WR14 3PS, UK).
Dr Jeffrey M Bradshaw (IHMC / UWF, USA)
(jbradshaw@ai.uwf.edu IHMC / UWF, Pensacola, FLORIDA FL 32501, USA).
Dr John Carson (DERA, UK).
(jcarson@signal.dera.gov.uk E203, DERA Malvern, Worcestershire, WR14 3PS, UK).
Dr Michael Kirton (DERA, UK).
(kirton@signal.dera.gov.uk E206, DERA Malvern, Worcestershire, WR14 3PS, UK).
Dr Niranjan Suri (IHMC / UWF, USA)
(nsuri@ai.uwf.edu IHMC / UWF, Pensacola, FLORIDA FL 32501, USA).
Prof Austin Tate (AIAI, Edinburgh, UK).
(a.tate@ed.ac.uk AIAI, The University of Edinburgh, EH1 IHN, UK).

Abstract

Software agents can be viewed as semi-autonomous entities which help people cope with the complexities of working collaboratively in a distributed information environment. This paper describes the research that DERA is carrying out into Software Agents for use in Command Systems and the collaborative work with the 16 partners of an international Coalition Agents Experiment. Specifically, the paper aims to show that using software agent-based C2 frameworks is a useful way of dealing with the complexity of real-world problems such as supporting agile and robust Coalition operations and enabling interoperability between legacy or previously incompatible systems. In addition, Agent-enabled 'grids' can be used to rapidly integrate a wide variety of agents and infrastructures, with domain management services structuring agent relationships, limiting their behaviours and enforcing Coalition policies.

Section 1- Introduction and Background

1. Software agents are currently receiving much attention in the research community. This interest is being driven by the phenomenal growth of the Internet and the World-Wide-Web. Agents can be viewed as semi-autonomous software designed to help people cope with the complexities of working collaboratively in a distributed information environment. This involves the agents communicating between the users and between themselves. The agents are used to find, format, filter and share information, and work with users to make the information available wherever and whenever users need it. This paper describes the research that DERA is carrying out into Software Agents for use in Command Systems (SACIS) and the progress which is being made in the collaborative work with the 16 international partners of the Coalition Agents Experiment (CoAX).

Military Context

2. Success in military operations involves carrying out high-tempo, coherent, decisive actions (faster than an opponent can react) resulting in decision dominance through the use of command agility. Command agility is about being flexible and adaptable so that fleeting opportunities can be grasped. This is done by the Commander issuing clear intent and then delegating the control authority to subordinates - allowing them the scope to exercise initiative. It also means being innovative, creative and unpredictable in a manner that (even if low-tempo) increases confusion in the mind of an opponent. This process is command led, which means that human decision-making is primary and that the role of technology is secondary. Shared understanding and Information Superiority are key enablers in this process and are fundamental to initiatives such as the UK's Joint Battlespace Digitisation programme[1]. Concepts such as Network-centric Warfare (see http://www.dodccrp.org/ncw.htm) also demand a more decentralised approach such as that provided by software agents.

3. In addition to the problems of integrating single-service and Joint capabilities into a coherent force, the nature of Coalition (now often just called multi-national) operations implies some need to configure incompatible 'come-as-you-are', or foreign systems, into a cohesive whole rapidly. Many problems in this environment can only be solved by organisational changes and by 'aligning' doctrine, concepts of operations and procedures. Coalition scenarios trigger the need for a rapid on-the-fly response and cannot be predicated on using pre-existing co-ordinated systems - hence the need for a flexible approach which allows capabilities to be assembled at 'run-time'. However, in addressing this requirement for interoperability, it is also crucial to address issues of security of data, control over semi-trusted software from other Coalition partners, and robustness of the resulting system (e.g. the ability to withstand denial-of-service attacks).

4. The current reality of Coalition Operations is often a picture of data overload and information starvation, labour intensive collection and co-ordination, individual stove-pipe 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. Indeed, for 'Cyberspace Superiority' to be obtained (as part of the Battlespace) then it is essential that the Coalition Forces are able to act decisively *inside* Cyberspace and that the *only way* that this can be achieved is though a variety of software agents acting on behalf of, or mediating the actions of, human users within Cyberspace - as well as at its margins *and beyond*.

Aims of the "Software Agents in Command Information Systems" and CoAX Projects

5. In this paper, we report on early progress in DERA's Software Agents in Command Information Systems (SACIS) project. Included in this is a related international collaborative effort whose overall goals are to demonstrate that the agent-based computing paradigm offers a

[1] Which aims to integrate the use of information across the Joint arena by exploiting appropriate doctrine, organisations and procedures, personnel and technology.

promising new approach to dealing with the technical issues of establishing coherent command and control (C2) in a Coalition organisation. This collaborative effort is being carried out under the auspices of DARPA's Control of Agent Based System (CoABS) programme, which provides a software agent-enabled "Grid" (see http://dtsn.darpa.mil/iso/ or http://coabs.globalinfotek.com/). The collaborative effort is called CoAX (Coalition Agents eXperiment) and it is a CoABS technology integration experiment (TIE) - see http://www.aiai.ed.ac.uk/project/coax/. The overall objectives of all this research are to determine and demonstrate the potential effectiveness of software agent technology to assist with issues of interoperability, etc in the context of military command systems. Specifically, the aims are to show that:

- agents are a useful metaphor for the complexity of real-world systems such as military operations;
- an agent-based C2 framework can support agile and robust Coalition operations;
- software agents can be used to enable interoperability between legacy or previously incompatible systems;
- the CoABS Grid can be used to rapidly integrate a wide variety of agents and systems - i.e., rapid creation of virtual organisation;
- domain policies can structure agent relationships and enforce Coalition policies;
- intelligent task and process management can improve agent collaboration;
- and that semantic interoperability can improve agent collaboration and interoperability between disparate Coalition command systems.

The SACIS and CoAX research has built a software agent testbed based on the DARPA CoABS Grid and this paper will describe the work done, the demonstrations carried out so far, the scenario and storyboard used and some of the initial insights which have been gained.

Structure of the Paper

6. The paper begins by providing a brief description of the key ideas and technologies underpinning software agents in Section 2. Section 3 describes the Coalition scenario and military command structure used in our demonstration experiments. Section 4 describes the corresponding agent architecture that was developed to reflect the military organisational structure. The events occurring in the storyboard used for the demonstrations so far are given in Section 5. A preliminary assessment of software agent capabilities and a discussion of future research are provided in Section 6. Concluding remarks are given in Section 7.

Section 2 - Software Agent Technology

7. There are a number of very good sources describing the state-of-art in software agent research and applications: see, for example, the Web site hosted by the University of Maryland [1]; the review article by Jennings et al [2]; the collection of research papers edited by Huhns and Singh [3]; the book edited by Bradshaw [4]; and the papers in the special issue of IEEE

Intelligent Systems [5]. Although there is no comprehensive and widely accepted definition of the notion of software agent [6], Jennings et al provide a useful working description:

> "An agent is a computer system situated in some environment that is capable of flexible autonomous action."

8. Jennings et al point out that "being situated" means that an agent is part of an environment from which it receives sensory input and that the agent can change in some way. Autonomy means that an agent should be able to act without the direct intervention of users. Flexibility implies that the system is responsive, pro-active and social - i.e., agents should, in principle, be able to interact with other agents and humans.

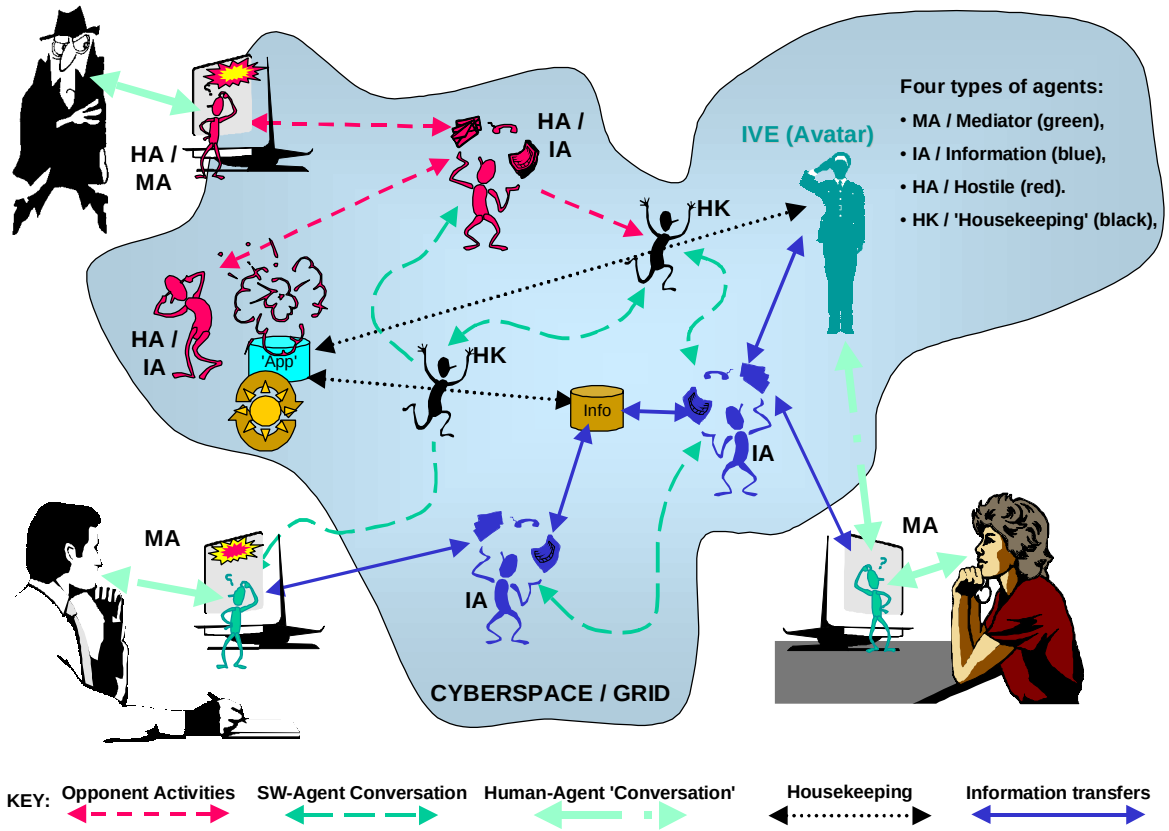

Figure 1 - Types of Software Agent

Figure 1 shows four main types of software agents. The 'Housekeeping Agent' (HK) is an entity which is responsible for assisting with the maintenance of Cyberspace, for example, by adjusting resource loading, monitoring for adverse performance, network routing, etc. The 'Information Agent' (IA) facilitates the movement, analysis and formatting of information in and through Cyberspace. A third entity is the 'Mediator Agent' (MA) which is the focal point of interaction between the human user and the underlying information, 'applications' and Cyberspace itself. In addition, there may be hostile versions of these agents. In SACIS and CoAX we use the following general definition for all types of software agents:

"Agents can be viewed as (software) entities acting on behalf of, or mediating the actions of, a human user and having the ability to carry out tasks autonomously to achieve goals or support the activities of the user."

9. In order for agents to communicate with one another and to share information, agent communication languages (ACLs) have been developed. There are two main ACLs: the Knowledge Query and Manipulation Language (KQML) and the Foundation for Intelligent Physical Agents (FIPA) ACL [7]. These languages handle propositions, rules and actions and are therefore at a higher level of abstraction than middleware such as CORBA (common object request broker architecture) and RMI (remote method invocation). In addition, there has been much work on the provision of a machine-understandable semantic-web of information such as eXtensible Markup Language (XML) and the Resource Description framework (RDF). Software agents can use RDF to advertise and describe their capabilities and can parse RDF descriptions to ascertain the relevance and utility of information provided by other agents. DARPA has recognised the potential contribution that technologies such as RDF could make to achieve semantic interoperability among software agents and has a new programme underway called DAML (DARPA Agent Mark-up Language [8]). The overall goal of the DAML programme is to develop a language aimed at representing semantic relations in machine-readable ways compatible with current and future Internet technologies. We are currently working on a specification and implementation of a DAML-based policy language (KAoS Policy Language or KPL), which will be used in CoAX to represent both simple atomic policies and complex constructions.

10. One way of viewing a community of agents is as a set of distributed, asynchronous processes communicating and sharing information by message passing in some infrastructure. In this regard, an important output from DARPA's CoABS programme is the CoABS Grid - a middleware layer based on Java / Jini technology that provides the computing infrastructure to integrate heterogeneous agent communities and systems rapidly. In a recent article, Jennings [9], argues that the agent paradigm is a good way of building complex software systems in general. Hence the potential benefit of using software agents in the Coalition setting, for example, where legacy command systems could be provided with software agent wrappers that allow them to inter-operate and share information with other systems and agent applications in a loosely connected, heterogeneous architecture that is underpinned by the CoABS Grid. The scenario, used as the basis of the experiments to test this hypothesis, is described in the next section.

Section 3 - A Representative Scenario and Coalition Command Structure

The SACIS and CoAX work needed a suitably realistic scenario for its experiments and so the Team expanded the fictional "Binni" scenario developed by Dr. A.R. Rathmell [10] for The Technology Co-operation Programme[2] (TTCP). In this scenario the year is 2012 and global warming has altered the political balance of the world. The action is set in an area that is currently the Sudanese Plain [see Figure 2 below]. Previously uninhabited land in the Plain is now arable and the area has received large amounts of foreign investment. A conflict has

[2] The C3I Group, Technical Panel 9.

developed between two countries who are fighting for control of Binni. To the north is Gao - which has expansionist aspirations but which is only moderately developed, with old equipment and with a mostly agrarian society. To the south is Agadez a relatively well developed and fundamentalist country. Gao has managed to annex an area of land, called it Binni and has put in its own puppet government. This action has come under fierce attack from Agadez. Gao has played the 'threat of weapons of mass destruction from Agadez' card and has enlisted support from the UN who have deployed a force, the UN War Avoidance Force for Binni (UNWAFB), to stabilise the region.

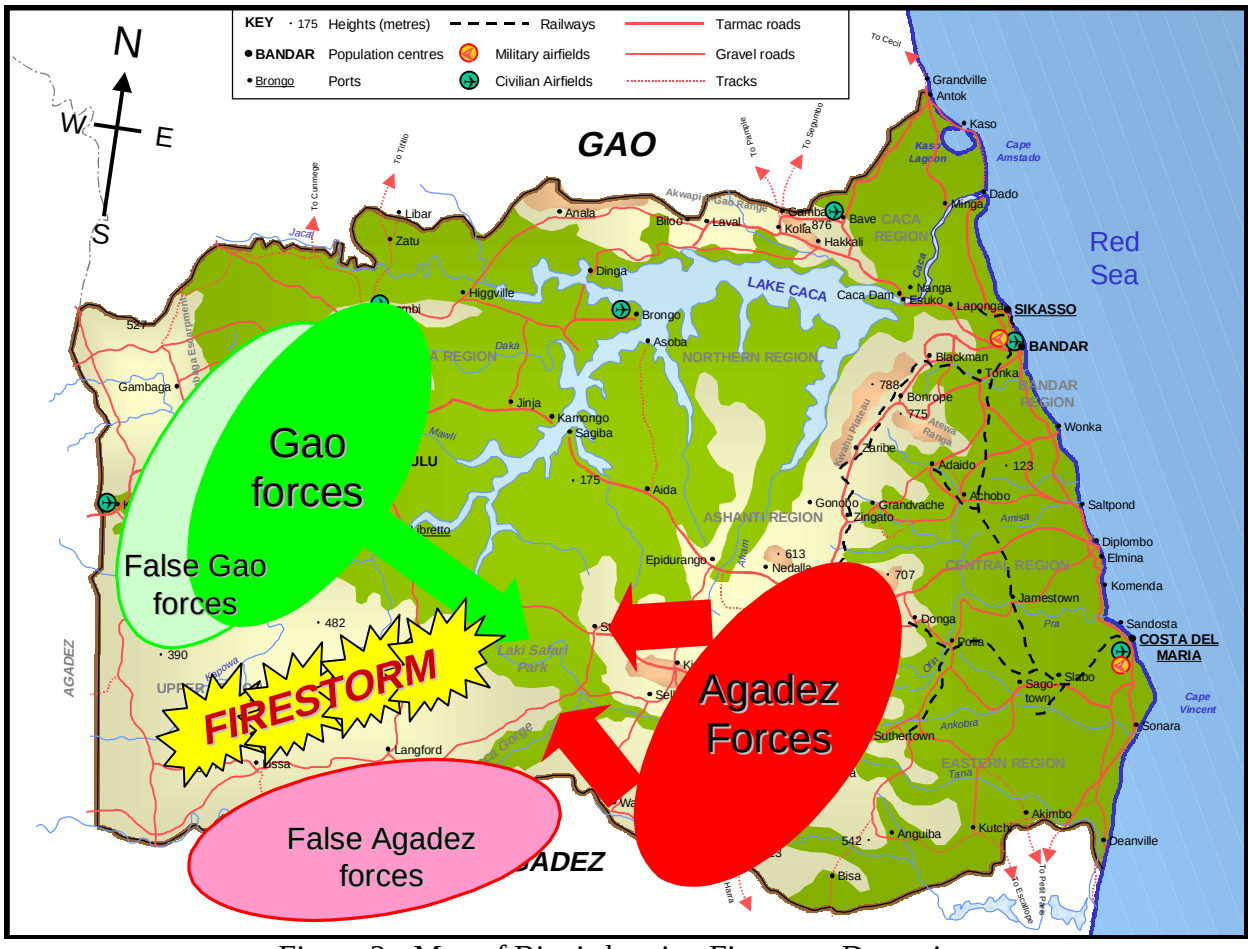

Figure 2 - Map of Binni showing Firestorm Deception

11. The UNWAFB has arrived in theatre and is not being opposed by Agadez. Gao is providing 'host nation' support in Binni at the ports and airports in the east and the Coalition Forces are working through an initial planning phase. One of the options under consideration is to lay down a 'firestorm' between the Gao and Agadez forces in this region. This will prove to be contentious as Gao will try to provide false information to displace the Firestorm. Also, the international media will hear of the operation and will object to the bombing taking place near a wildlife refuge area (the Laki Safari Park).

12. This Binni Coalition (multi-national) operation needs to rapidly configure various incompatible, 'come-as-you-are' or foreign systems into a cohesive whole within an open,

heterogeneous diverse and dispersed environment. This scenario provides a suitable test for the software agent experiments, where the run-time composability of software agents is a very close metaphor for the dynamic uncertainty of Coalition Operations. The complexity of the situation must not be underestimated and is best illustrated by looking at the Binni Coalition Command Structure shown in Figure 3 below.

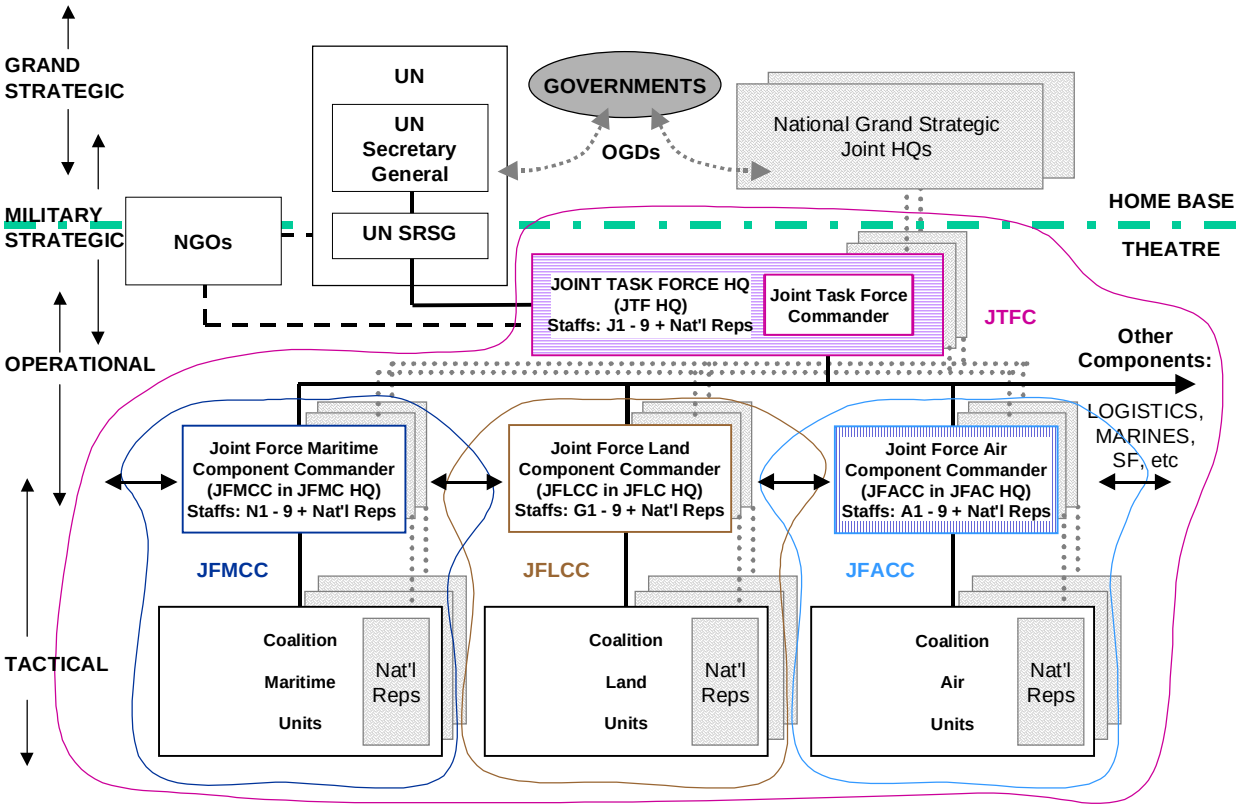

Figure 3 - Representative Coalition Command Structure

13. This is a representative and realistic Coalition command structure involving the UN, Governments, Other Government Departments (OGDs - such as the Foreign Office), Non-Government Organisations (NGOs - such as Oxfam), representatives of all the Coalition Countries (with their own 'ghosted' Command Structures) and the Coalition HQs and subordinate fighting forces. The solid black lines on the diagram show the legal lines of command authority (the 'command chain') and accountability. Dashed lines show an advisory / negotiating role. This is the kind of Coalition structure which would be agreed by the participants and no part of it is 'owned' by any specific country. Other possible Components (Logistics, Special Forces (SF) etc) are not shown. Note that the 'Levels of Command' overlap and their boundaries are not rigidly defined - as a general rule though, the JTFHQ and Component HQs span the critical operational / tactical boundary, which can *roughly* be equated to the planning / execution boundary.

Section 4 - Corresponding Software Agent Architecture

Human 'Domains'

14. Integrating the use of information across a Coalition is not just a matter of employing technology - it involves the creation of a coherent 'interoperability of the mind' at the human level as well as exploiting appropriate doctrine, organisations, personnel and procedures. In a Coalition many social and cultural factors, therefore, come into play. The SACIS and CoAX Teams realised that the mapping between the human and technical worlds was not straightforward and that, from the human perspective, four kinds of 'domains' could be identified as follows:

- Organisational Domains: These relate to a span of control or legal authority of a command entity where information would be shared across the domain at a common security level. The SACIS and CoAX work has focussed on the Joint level of command - in particular on two organisation domains: the Joint Task Force HQ (JTF HQ) and the Joint Force Air Component HQ (JFAC HQ). Technically, the JTF HQ and the JFAC HQ are sub-domains of the Coalition Organisational structure shown in Figure 3.
- Country Domains. Each of the National command chains would be a separate, self-contained 'domain' with its own processes, information, security regime etc. The interface with the Coalition is often through the National Representatives (liaison officers) who carry out any necessary 'translation' and act as a 'safety gap' for security reasons. Figure 3 shows each Nation 'ghosting' all of the Command HQs - in practice each Nation will provide different degrees of command 'presence' throughout the Battlespace.
- Functional Domains. These relate to a set of entities collaborating on a common task. Such domains may be virtual (ie exist only in Cyberspace), are often informal and may come and go as the military imperative changes. A more formal functional domain would be the Intelligence community which spans various levels of command.
- Individual Human Domains of Responsibility. In simple terms Commanders have responsibility for the effective running of their own HQ *and* all the subordinate ones (in practice they delegate this authority). Hence the individual human domains of influence may overlap - shown with the shapes with irregular boundaries on Figure 3 above.

These types of domains are not entirely exclusive and there are many different levels of overlap and interaction depending on the viewpoint taken. It is this complexity at the human level that create difficulties for technical systems.

Software Agent Domains

15. A software agent domain could be more tightly defined in technical terms as follows:

> "Software agent domains are bounded objects (which can interoperate via intermediate structures) with clear identities and ethos. Each domain contains entities and structures working collaboratively and sharing information, processes and 'control' procedures such as security regimes and policies."

Hence, the SACIS and CoAX Teams maintain that software agent domains provide a better mapping to the human domains described above than some other technical approaches. The next section explains how the software-agent domains were defined and how they operate.

16. In a software-agent-enabled infrastructure (such as the CoABS Grid) agent domain management services are defined. These services will evolve from and enhance existing services available within the software agent framework. An agent domain consists of a unique instance of a domain manager (DM) along with any agents that are registered to it. The function of a domain manager is to: 1) manage agent registration, 2) serve as a single point of administration for policy management. That is, the domain manager could configure, re-configure, store, publish and enforce polices that exist for that domain. Domains assure those who deploy agents systems that there is policy uniformity across multiple platforms and hosts, as long as semantically equivalent monitoring and enforcement mechanisms are available across those platforms and hosts. Under these conditions, it would follow that a given domain could extend across host boundaries and, conversely, multiple domains could exist concurrently on the same host. With respect to platform independence, it should be possible for agents running on the same platform to be in different domains (for example, a resident and a visiting mobile agent running on the same platform may belong to different domains having more or less restrictive security privileges). In addition, domains contain match-makers (MM) which work together to provide information about local and remote agent services that are available.

17. A policy is a machine-readable set of statements in which some element (such as an agent) of an agent system declares a specification intended to describe or govern its interaction with other elements of the agent system. For example, an agent may declare a policy that all messages it exchanges with other agents must be encrypted, or that certain timing and message sequencing constraints must be observed when requesting a particular kind of service from that agent. The latter is an example of a conversation policy. A domain manager has policies such as:

- no agents registered to its domain may communicate with agents outside the domain;
- no agent can consume more than a given fraction of some system resource;
- agents must respond to messages from the domain manager within a given time frame;
- agents with higher priority tasks pre-empt lower priority ones.

The policy is expressed in a persistent machine-readable format, which could be interpreted by a platform-specific policy enforcement mechanism. At the current time, the representation is very simple, but a more powerful representation in DAML is currently being defined. Policy and policy-enforcement mechanisms could be defined in multiple locations in a given implementation. The separation of policy specification from policy-enforcement mechanisms allows policies to be dynamically reconfigurable, and relatively more flexible, fine-grained, and extensible. Agent developers can build applications whose policies can change without necessarily requiring changes in source code. The rationale for using declarative policies to describe and govern behaviour in agent systems includes the following claims: easier recognition of non-normative behaviour, policy reuse, operational efficiency, ability to respond to changing conditions, and the possibility of off-line verification.

Software Agent Domains in SACIS and CoAX

18. The SACIS and CoAX Teams have set up a demonstration containing 23 separate software agents grouped into 6 agent domains (including one sub-domain). The interoperability of the agents and systems was made possible using the inter-agent communications services provided by the DARPA CoABS Grid, with the policies enforced by KAoS[3] domain management services. Figure 4 below shows that within each agent domain there is a matchmaker (MM) and domain manager (DM), and then a set of software agents and agent-enabled tools / applications, which are described further below.

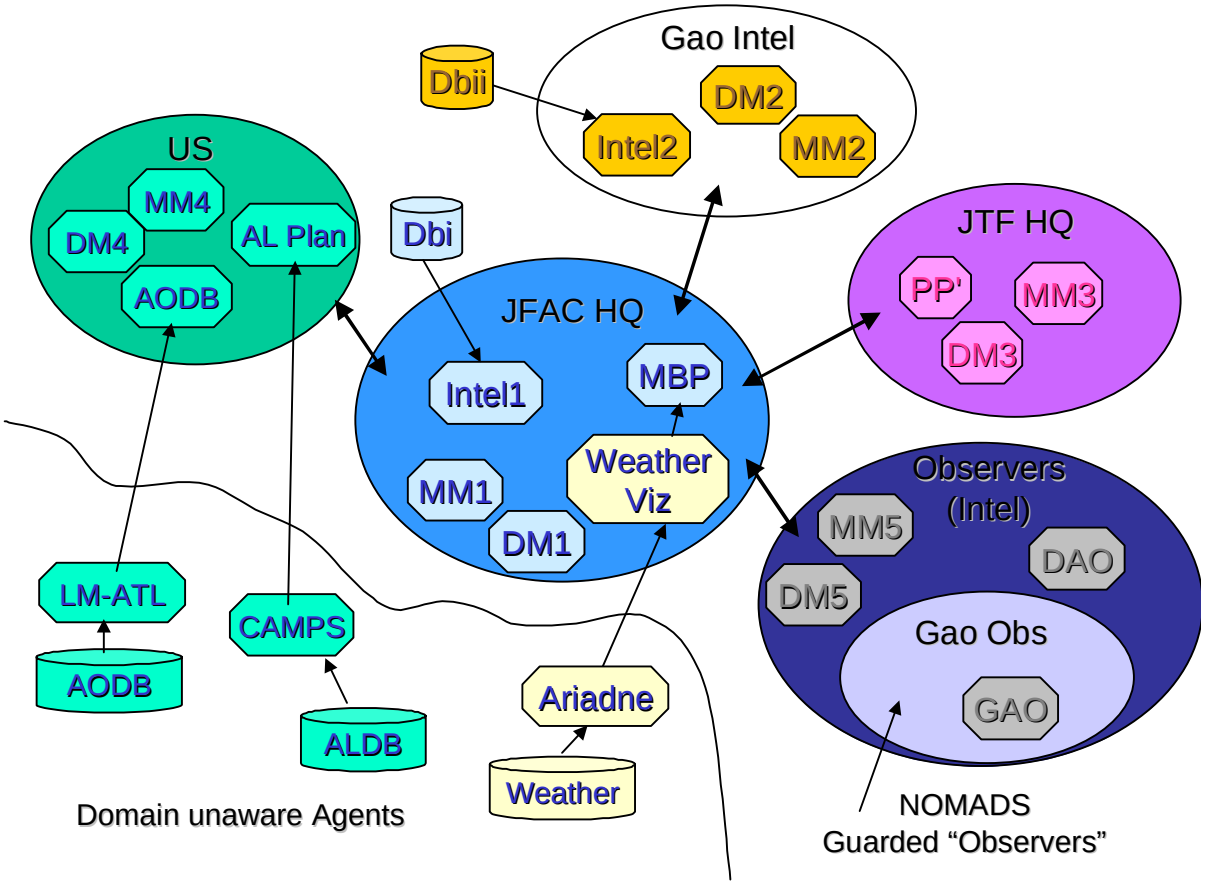

Figure 4 - Software Agent Domain Mappings

19. The SACIS and CoAX teams have carried out a number of experiments. In the one described here two legacy systems from different nations were brought together and agent-enabled, along with an agent process management tool. In addition, a number of agent-wrapped databases, agent-wrapped public domain Internet information and agent domain and policy administration and malicious agent management tools were provided. The supporting infrastructure was built on the DARPA Grid.

[3] Knowledgeable Agent-oriented System - from IHMC / UWF and Boeing.

20. In the JFAC HQ the Joint Force Air Component Commander's (JFACC) staff use a UK tool called the Master Battle Planner (MBP). The MBP assists the planners by providing them with a visualisation - which relates directly to their way of thinking because it is an Expressive System [11] - on which they can manipulate the air intelligence information, assets, targets and missions etc to build up an Air Battle Plan. The MBP has been given a software agent wrapper to enable it to bring together information from the various Coalition partners in near real-time - and this provides a significant improvement in overall capability.

21. There is also a weather cell in the JFAC HQ which can acquire information from the Internet via the Ariadne (see http://www.isi.edu/info-agents/ariadne/) software agent. Ariadne runs off-site and gathers information from a number of weather web-sites. WeatherViz shows a world map, a region-of-interest map and the weather reports received. This information is put onto the agent information Grid where it is picked up by MBP and other agents and can be used by the operators to inform planning - again, a significant improvement in capability.

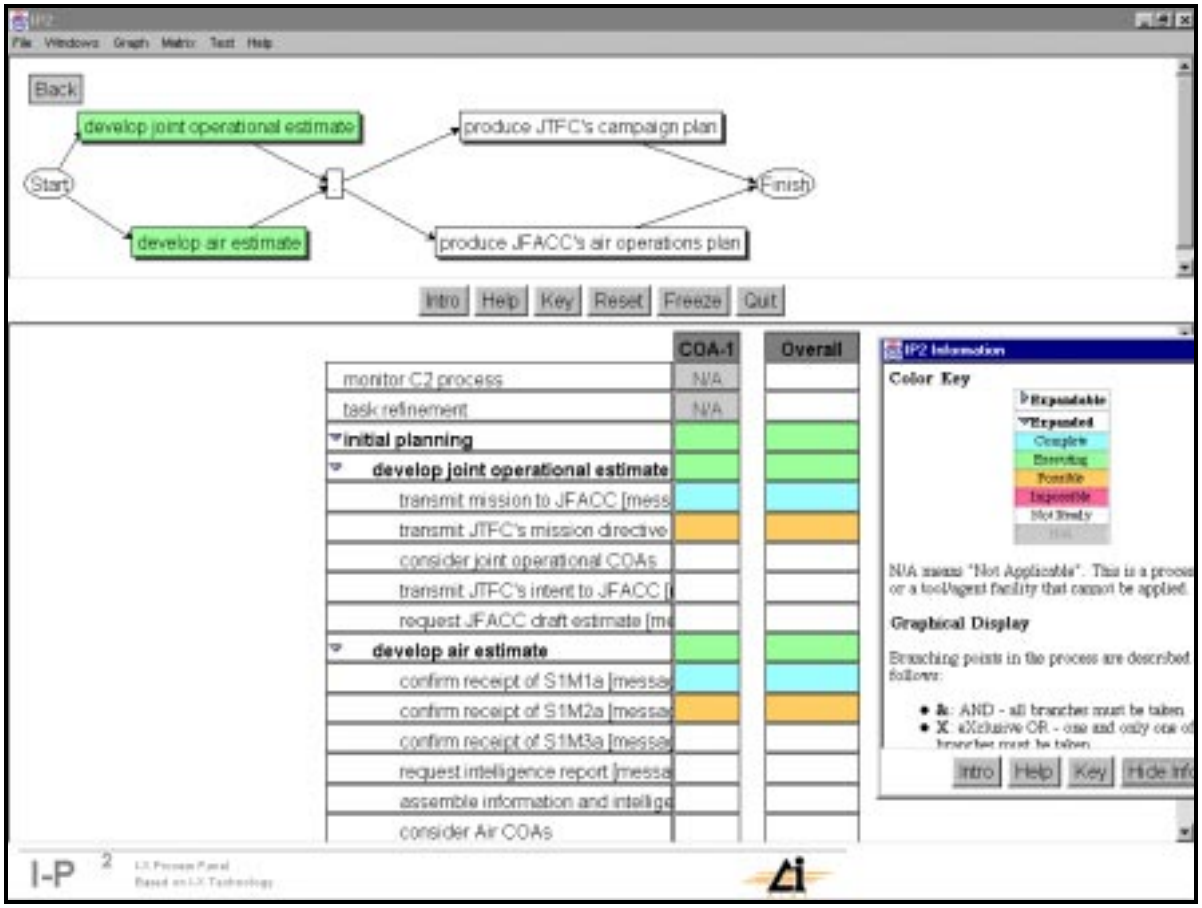

Figure 5 - The Process Panel

22. The Process Panel (PP), see Figure 5 above, in the JTF HQ domain is an agent-based tool for providing the Joint Task Force Commander (JTFC) with a campaign 'process' visualisation. One window shows the main military events in a schematic view, whilst the others show other views such as task breakdown and process products. Colours are used to indicate the status of the events. This tool can be used to plan, evaluate, schedule, review and monitor courses of action

and, through the use of software agents, the PP can 'sense' remotely the status of activities which increases Coalition situational awareness and provides a warning of possible critical path events.

23. The US country domain has access to an agent-enabled tool called the Consolidated Air Mobility Planning System (CAMPS). This consists of a map display and a number of tools for managing airlift assets, their home bases and potential destinations, information about loads and for creating airlift missions. The domain also contains an agent-wrapped air operations data base (AODB) which has US Intelligence and Operations information in it. Some of this information is made available to the Coalition via the software agent wrapper - improving Coalition situational awareness and reducing the chance of conflicts.

24. In addition, there is a functional domain dealing with feeds from observers in the field. These observers are from several countries, from different parts of the organisation and involve different flavours of agent (eg the Dartmouth Agadez Observer (DAO) is a surrogate agent for D'agents from Dartmouth College, USA). There are also Gao observer agents but, as there is some distrust of these agents, they are put in a more secure, guarded domain which exploits the security and resource control aspects of NOMADS. In particular, the resource control mechanisms can protect hosts from denial-of-service attacks from malicious agents. This use of software agents *has* allowed greater interoperability, but a risk assessment shows that appropriate measures have also been introduced to protect security.

Section 5 - Experiment Story Board

25. Several demonstrations have been given of the results of the SACIS / CoAX work and screen movies and briefings are available on the CoAX web site mentioned in Section 1. In addition, the demonstration was successfully given to military staffs in Miami in February 2001 leading to a request for the Team to bid for participation in Millennium Challenge 2002 and JEFX 2002. The demonstrations have used a militarily plausible story board which showed Coalition, Country and Observers information being located, fused and employed to meet military imperatives during initial planning in the Binni scenario.

26. The demonstration begins in the JFAC HQ where the Air Component Commander's staff bring together information from the various Coalition partners. They use the MBP, whose wrapper agent detects the human activity and which then sends out requests for updates to the various Coalition agent information resources. Once information has been received, the MBP (see Figure 6 below) displays icons for enemy potential targets such as airbases, ground forces, and SAM sites; and for friendly air units, ships and airspace regions.

27. The JFACC Staffs have also received the JTFC's Guidance documents and they acknowledge their receipt on the MBP. This triggers an agent which informs the JTFC's Staffs of this via the Process Panel, which now shows that the documents have been received by the JFAC HQ, and changes the status of related tasks to show that they can now proceed. This means that software agents are enabling the JTFC to monitor the progression of the overall planning task throughout the Coalition and be alerted quickly if critical-path events are occurring.

28. Back in the JFAC HQ the staff gather more information about the Theatre of Operations. They need to know about the weather in the Firestorm area, the disposition of Gao and Agadez ground forces, and whether there are any other air missions in the area. An operator uses WeatherViz to select the appropriate weather station and requests an update from Ariadne (which extracts the relevant data from publicly available information on the Internet). Once this information arrives it is sent to the MBP and appears as an icon on the map display. This shows how the software-agent enabled infrastructure can allow operators to access suitable information as and when they need it, regardless of where it is located.

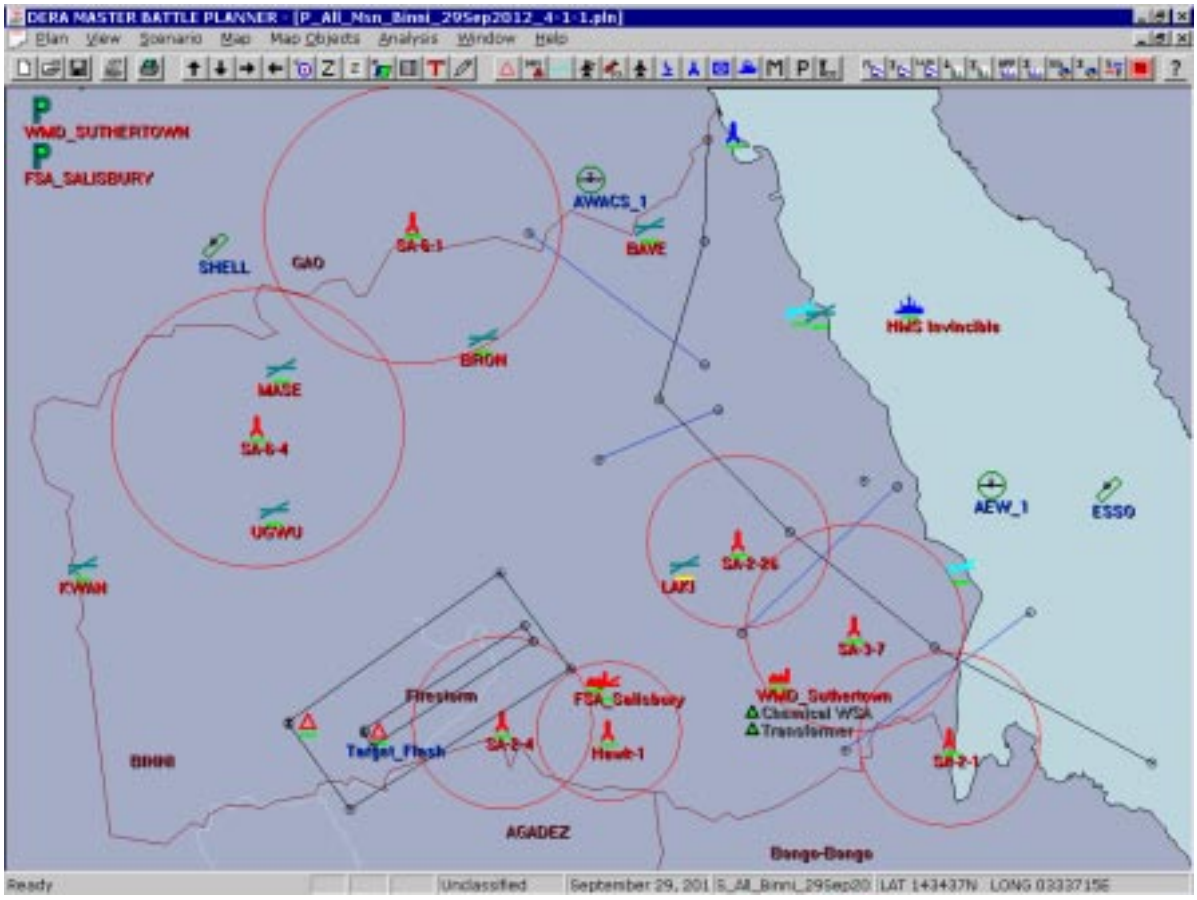

Figure 6 - Master Battle Planner

29. While this has been going on, in the USA Country Domain, work has started on planning the movement of critical weapons into the Theatre of Operations. The Firestorm requires the use of a new weapon and the USA intends to fly the weapons to Cyprus and then into Theatre. An operator uses the software agent GUI (which could be done remotely with the software agents mediating the interaction) on CAMPS to edit the load and number of aircraft to be sent. The plan is validated by CAMPS and a route generated from Cyprus into Binni. The operator accepts the plan and tells CAMPS to "Achieve it" which means that an order is generated containing all the relevant information. The missions generated are then sent through the agent infrastructure and displayed automatically in the MBP. In the JFAC HQ, when the planners subsequently generate

the necessary offensive, defensive and support missions for the Firestorm, as they now know about the airlift missions and won't create other missions which would conflict with them.

30. Back in the JFAC HQ, the planners are examining the Firestorm options in detail. Based on initial Intel, the RECCE area and possible Firestorm target areas have been defined. However, unknown to the Coalition staffs, Gao doesn't like the Firestorm option and is trying to subvert it by feeding in misinformation about the location (see Figure 2) of Gao and Agadez forces. The Coalition has observers in the field who have been feeding in their observations, which are different to those from Gao. To firm-up the Firestorm options the planners check the currently known positions of Gao and Agadez forces. They do this by double-clicking on several of the assets on MBP's map display. This triggers software agents to request other agents to return their up-to-date information on the assets - for some of the assets two dialog boxes pop-up containing different information - one from the Gao agents and one from the JFAC HQ's Intel agents. The planners realise the deception and decide to cut off the communication with the Gao agents.

31. The Coalition System Administrators have access to the KAoS Policy Admin Tool (KPAT) which is used to block access to the Gao domain. The purpose of KPAT is to provide an easy-to-use domain management interface for Coalition agent system administrators (SysAds) to control the behaviour of groups of agents dynamically. Coalition agent policies can be developed and verified in advance and stored in a policy library, or they can be created and enforced on-the-fly. By providing domain management through policies, rather than through requiring agent developers to write special purpose code, three benefits accrue:

- the burden on agent developers is reduced;
- changes in policy can be effected dynamically on software agents to change their behaviour at run-time without changes to the agent code;
- policies can be enforcement on malicious agents without them knowing anything about this in advance or even at all.

A policy is defined to block communication from the Gao agents and it is selected and committed - which changes the domain policies such that the Gao information is excluded. Once access to the Gao domain has been turned off, the planners refresh the MBP's display by requesting an update of information through the Coalition agent network. It is found that many of the ground unit positions alter. This is because they are now only using the accurate information from observers in the field and not on the misinformation from the Gao observer agent. This has demonstrated another feature of the agent infrastructure - robustness and the ability of the system to reconfigure automatically as levels of service change and as entities come and go.

32. The final part of the demonstration is driven by the fact that Gao is unhappy at this turn of events and covertly begins a denial-of-service attack. At the start of the UNWAFB deployment an 'observer' domain was created containing Dartmouth Gao Observer (DGO) and Dartmouth Agadez Observer (DAO) agents. Gao requested that one of its agents (Gao Agadez Observer, GAO) is put on the sensor platform that is hosting DAO, so that it can observe Agadez movements independently. Because there is some mistrust of Gao, permission is granted, but GAO is required to run under the NOMADS environment, which allows the SysAds to change and monitor the computing resources given to an agent. After the Gao Intel domain is cut off,

Gao intends to avenge itself by launching a denial of service attack on the DAO host machine (by writing continuously to hard disk and using up CPU and network resources) by its GAO agent within the observer domain.

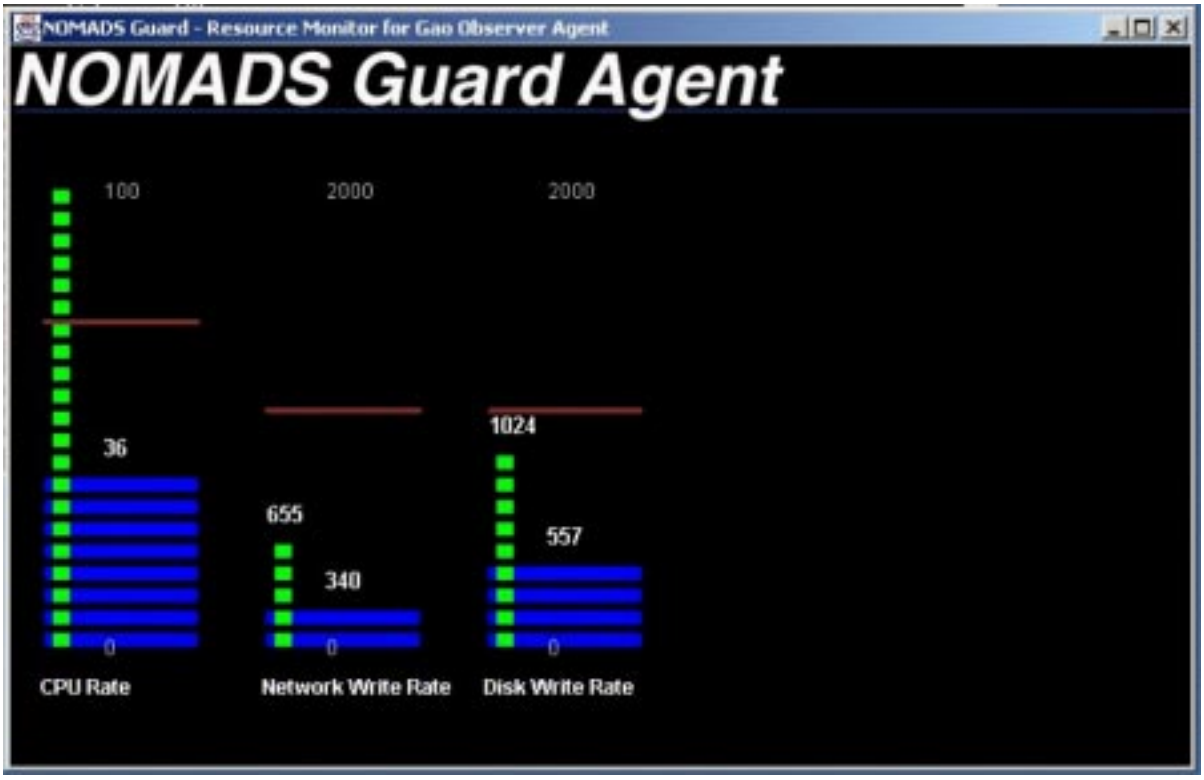

Figure 7 - The NOMADS Guard Display

33. A sentinel function of the NOMADS guard (see Figure 7 above) monitors the resources used by the GAO agent. At first, the average computing resource consumption of the GAO agent is reasonable, but when the denial-of-service attack is launched, the GAO agent consumes resources excessively. In consequence of the attack, the (friendly) DAO agent's ability to perform its tasks is slowed to a halt. The KAoS domain administration services must not only be able to respond to administrator-requested policy changes, but also automatic event-driven changes. A sentinel function of the NOMADS guard detects the denial-of-service attack. It automatically responds to the attack by reducing the resource limits set (the original limits before the attack are shown as the horizontal red bars in Figure 7) of the GAO agent and in effect neutralises the attack.

34. In summary, the demonstration has shown part of a realistic UN supported Coalition operation involving several command HQs, organisational, national and functional entities working together in a software agent enabled infrastructure with various levels of access and functionality. The demonstration has shown how legacy systems can be agent-wrapped and interoperability achieved with other agents tools and services - all supported by the CoABS Grid.

Section 6 - Assessment of Software Agents

Technical Progress to Date

35. To date, the key technical achievements of SACIS and CoAX are as follows:

- we have demonstrated a proof-of-concept, agent-based Coalition C2 architecture within a representative Coalition scenario;
- for the first time, we have shown legacy and previously stand-alone US and UK military systems inter-operating via agent technology;
- we have shown how agent organisation, behaviour, security and resources can be managed by explicit Coalition policy control;
- specifically, the demonstrations have shown certain features and benefits of an agent-enabled environment such as:
    - operators working collaboratively with agents;
    - agents working semi-autonomously 'in the background';
    - the robustness of an agent-enabled infrastructure (the DARPA CoABS Grid) - reconfiguring automatically;
    - access being provided (on demand) to remote, previously stand-alone and dispersed information and remote operation of applications;
    - agents' use of communication and computing resources being controlled;
    - the ability to create shared understanding and improved visualisation.

Assessment and Future Research Programme

36. The SACIS research started in April 1999 and the CoAX project officially began in February 2000 and we believe that we have made good early progress towards demonstrating the utility of agent technology in Coalition operations. We have put together a prototype Coalition C2 architecture that supports and embraces heterogeneity and have exercised this in an agent-based C2 demonstration that enacts Coalition activities within the Binni scenario. The CoABS Grid and KAoS domain management capabilities have allowed us to interoperate US and UK military systems as well as a variety of agent-based information resources. In should be noted that the CoABS Grid, in particular, has played a vital role in rapid integration of systems.

37. Assessment work funded by DARPA CoABS programme has reported favourably on the performance issues of agent-enabled infrastructures and the experiences of the SACIS and CoAX Teams have shown that the agent-wrapping of legacy systems and the integration of different agent systems at short notice is relatively straightforward. Indications are that agent wrapping is simpler where systems 'expose' more of their internal information and methods and work is also in hand in this area. In addition, a heterogeneous set of agents can be made to interoperate as long as implementers adhere to some minimum set of message and other standards. Also, heterogeneity should be accepted and embraced as it is seen as being inevitable and can actually be beneficial in a number of cases - especially in security terms.

38. The next phases of the research work are designed to further enhance the software agents and provide a more exacting demonstration capability. These phases are described below:

a. CoAX Binni 2001: This work is due in July 2001 and will:

- move on to supporting the execution phase of conflict (which is characterised by being more uncertain and which has more demanding performance requirements) with software agents - see Figure 8 below;
- take on and address C2 process tracking;
- improve domain management by allowing nested and overlapping domains;
- address both the planning and execution phases of a realistic military scenario;
- involve up to 45 agents in 7 domains that are part of a realistic and extended Coalition C2 architecture.

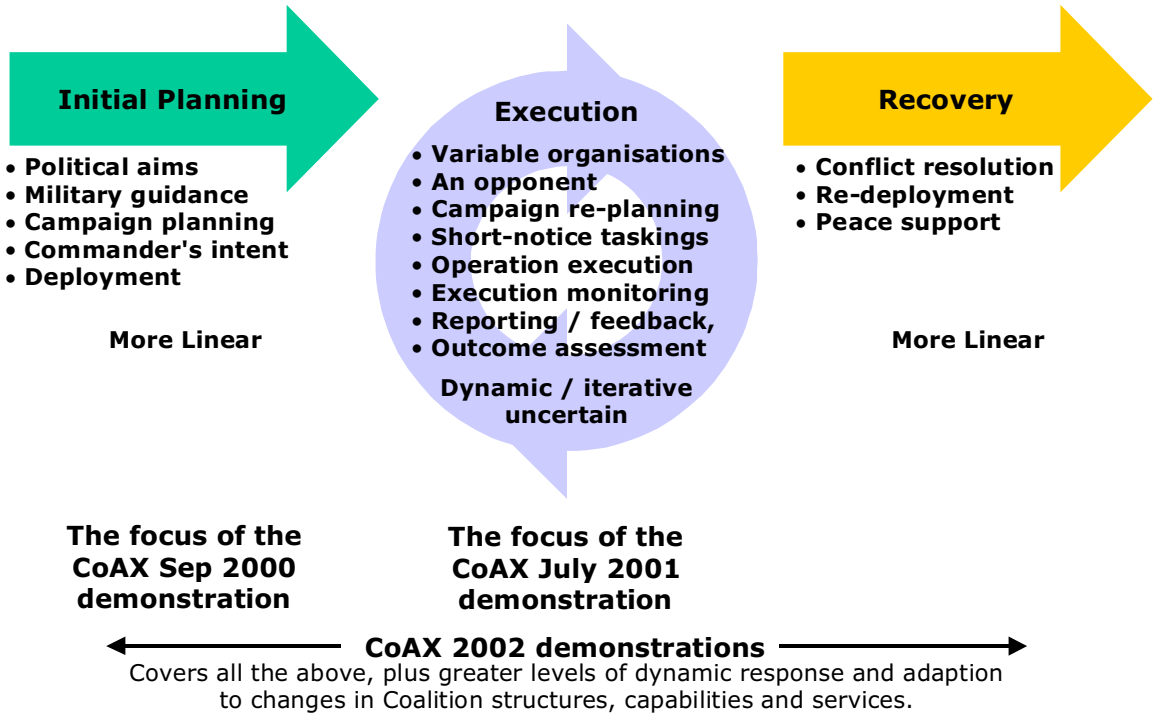

Figure 8 - Phases of an Operation and the CoAX demonstrations

b. CoAX Binni 2002: This work is due in April 2002 and will:

- add the ability to monitor, plan and control the Coalition C2 processes and deal with events arising from execution;

- include obligation management, eg: ensure that agents are meeting their commitments;
- involve the dynamic creation of ‘virtual Coalition organisation’;
- see agents and domains added to Coalition structure ‘on-the-fly’;
- deal with dynamic Coalition tasks and processes;
- use a dynamic Coalition scenario, with partners joining / leaving unpredictably;
- cope with a dynamic, uncertain and event-driven execution environment.

This work has been submitted for inclusion in Millennium Challenge / JEFX 2002.

c. In the research so far, a certain amount of hardwiring and pre-agreed formats have been employed and in the next phase of the work semantic web technology will be used to investigating further how run-time interoperability can be achieved 'on-the-fly' as it is felt that semantic interoperability can significantly improve agent collaboration.

Military Implications of the Results

39. Software Agents can be viewed as entities acting on behalf of, mediating or supporting the actions of human users and having the ability to carry out tasks autonomously to achieve goals or assist the activities of the users. This research project has shown how software agents can carry out tasks which enable interoperability between information systems and infrastructure services brought together in a ‘come-as-you-are’ Coalition.

40. In the experiments so far, the software agents operated in a number of roles. They have worked ‘in the background’ – through matchmaking, domain management, process management and other agent services – to find, establish and maintain the infrastructure, information and procedural links necessary to achieve and support interoperability in a dynamically changing environment. In addition, they have worked collaboratively with human operators, mediating requests for information and formatting and displaying the results almost transparently.

41. Dealing effectively with unpredictable changes – owing, for example, to the destructive activities of opponents or because of systems failing and services being withdrawn – is a typical Coalition problem where software agents could make a significant contribution. So far, we have shown that a software agent infrastructure is robust and, to some extent, is 'self-healing'. Our aim is to investigate this further to show that software agents can provide agility, robustness, flexibility and additional functionality beyond that provided by the individual Coalition partners.

42. Our conclusion is that software agents, together with agent-based infrastructures and services provided by the CoABS Grid and KAoS, could play a key role in supporting Coalition operations. We think that this technology will provide the ability to bring together and integrate systems quickly to aid in all aspects of Coalition operations, without sacrificing security and control. Our long-term goal is to use this technology in the creation, support and dynamic reconfiguration of virtual organisations - with Coalitions being an archetypal and timely example of an area where this technology is vitally needed.

Section 7 - Concluding Remarks

43. The central hypothesis that is being investigated in SACIS and CoAX is that the agent-based computing paradigm is a good fit to the kind of computational support needed in Coalition operations. Thus far, we have shown that agents can usefully share and manage access to information across a stylised Coalition architecture in support of planning. This has required our gaining knowledge and expertise in, for example, agent communication languages, agent infrastructures, agent policy management [12], and agent legacy-system integration.

44. Over the next year, SACIS and CoAX have a series of technical demonstrations planned of increasing complexity. Building on the baseline capabilities already established, in the next phases of CoAX our priorities will be on demonstrating how agent technology can deal with the dynamic aspects of Coalition formation and the uncertainties of the execution phase where we will face an opponent. We envisage showing partners joining or leaving, services becoming unavailable, as well as agents aiding the human problem-solving and decision-making process in response to external events.

45. One early lesson has been that the Cyberspace inhabited by the software agents should not be seen just as an information pipe between humans - it is a Battlespace in its own right. This indicates that 'Cyberspace Superiority' should be obtained (as for any other part of the Battlespace) and that to achieve this it is essential that the Coalition Forces are able to *act decisively inside Cyberspace*. As humans cannot physically enter Cyberspace, it may be that the only way that Cyberspace Superiority can be achieved is though the use of a variety of software agents acting on behalf of or mediating the actions of human users.

References

1. University of Maryland Baltimore County AgentWeb: http://agents.umbc.edu/

2. Jennings, N R, Sycara, K, and Wooldridge, M. 1998. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:275-306.

3. Huhns, M N, and Singh, M P (editors) 1998. *Readings in Agents*. Morgan Kauffman: San Francisco, California.

4. Bradshaw, M (editor) 1997. *Software Agents*. AAAI Press: Menlo Park, California.

5. *IEEE Intelligent Systems* 1999. 14(2).

6. Shoham, Y. 1999. What we talk about when we talk about software agents. *IEEE Intelligent Systems*, 14(2): 28-31.

7. Labrou, Y, Finin, T, and Peng, Y. 1999. Agent communication languages: the current landscape. *IEEE Intelligent Systems*, 14(2): 45-52.

8. DARPA Agent Mark-Up Language: http://www.darpa.mil/iso/ABC/BAA0007PIP.htm and http://dtsn.darpa.mil/iso/programtemp.asp?mode=347 and http://www.daml.org/

9. Jennings, N R. An Agent-based Approach for Building Complex Software Systems. *Communications of the ACM*. Vol 44, No: 4. pp. 35 to 41. April 2001.

10. Rathmell, R.A. (1999) 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*, (ed. Tate, A.) pp. 115-125, Edinburgh, Scotland, 10th-11th May 1999.

11. Pawson, R. Expressive Systems, 2000. http://expressive-systems.org/

12. Bradshaw, J., Suri, N., Kahn, M., Sage, P., Weishar, D. & Jeffers, R. Terraforming Cyberspace: Toward a policy-based grid infrastructure for secure, scalable, and robust execution of Java-based multi-agent systems. Proceedings of the Workshop on Agent-based Cluster and Grid Computing, *IEEE International Symposium on Cluster Computing and the Grid* (CCGrid2001), Brisbane, Australia, 14-18 May, 2001. (Enlarged version to appear in IEEE Computer, in press).

Acknowledgements

DERA work was carried out as part of the Technology group 10 of the UK MOD Corporate Research programme. We also acknowledge the contributions of the following CoAX partners:

- Ariadne Agents - Public Domain Weather provision (USC / ISI).
- Consolidated Air Mobility Planner (CAMPS) - legacy airlift tool (GITI / BBN / AFRL).
- Domain Management - Knowledgeable Agent-oriented System (KAoS) (UWF / IHMC and Boeing).
- EMAA / CAST agents - US Air Intelligence Provision (Pete Gerken, LM-ATL).
- Malicious Agent control - KAoS Policy Administration Tool (Boeing, UWF / IHMC).
- Master Battle Planner (MBP) agent-wrapped, legacy planning tools for Air Battle Planning (Chris Walker and Don Brealey, DERA).
- NOMADS Mobile Agent System (UWF / IHMC and Boeing).
- Observer D'agents (Dartmouth).
- Process and Task Management tools (AIAI).
- The DARPA "Grid" - an agent-enabled infrastructure (GITI, ISX).

Edinburgh work on the CoAX project is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Command and Control Directorate under grant numbers F30602-99-1-0024. UWF / IHMC work on KAoS and NOMADS is supported by a contract from DARPA's Control of Agent-based Systems (CoABS) program (Contract F30602-98-C-0170), the DARPA Ultra*Log program, the NASA Cross-Enterprise and Intelligent Systems programs, and the National Technology Alliance.