KSCO 2002

Second International Conference on Knowledge Systems for Coalition Operations
23rd and 24th April 2002
Toulouse, France

Edited by Austin Tate
Artificial Intelligence Applications Institute
The University of Edinburgh

Supported by:
Australian Defence Science and Technology Organisation
Canadian Defence Research Establishment Valcartier
European Office of Aerospace Research and Development
UK Defence Science and Technology Laboratory
IEEE Intelligent Systems

Contents

Introduction

The first Knowledge Systems for Coalition Operations (KSCO) meeting was held in Edinburgh in May 1999 and focussed on Knowledge-Based Planning for Coalition Operations. An international working group of interested individuals was formed at that meeting to encourage international collaboration on KSCO. The KSCO-2002 conference is the second in a series of international meetings which aims to bring together practitioners and key decision makers in coalition operation management with researchers from areas of knowledge representation and reasoning, planning, multi-agent systems and related areas in order to exchange experience and ideas, share inspiration and suggest novel concepts. Practitioners benefit from meeting each other and from learning the possibilities of recent research achievements while researchers will get inspiration from each other and links to potential end users of their ideas.

Area of Conference

Topics for discussion include:

- Innovative theory and techniques for coalition formation and support to similar "virtual organisations"
- Applications and requirements for knowledge-based coalition planning and operations management
- Knowledge-based approaches to command and control
- Knowledge-based approaches to coalition logistics
- Knowledge-based approaches to Operations-Other-Than-War - such as peace keeping missions and other humanitarian operations
- Multi-agent systems and the concept of agency in coalitions
- Tools and techniques for knowledge-based simulation and modelling of coalition operations
- Security and maintenance of private information or knowledge in coalition operations
- Autonomous vs. centrally managed coalition operations

KSCO Working Group

Further information on the work of the Knowledge Systems for Coalition Operations working group and the conference series is available at

http://www.aiai.ed.ac.uk/project/coalition/ksco/

KSCO-2002 Organising Committee

- Jean Berger (DREV Canada)
- Jeff Bradshaw (UWF/IHMC US)
- Roberto Desimone (QinetiQ UK)
- Richard Devonshire (MoD UK)
- Nort Fowler (AFRL US)
- Dale Lambert (DREV Australia)
- Michal Pechoucek (Czech Technical University)
- Chris Reuter (EOARD Europe/US)
- Austin Tate (AIAI UK)

Special thanks to Jeff Bradshaw and Michal Pechoucek who took responsibility for the paper reviewing process.

Acknowledgements

The KSCO-2002 organisers would like to thank the sponsors of the event for their support. These are the European Office of Aerospace Research and Development (part of the Air Force Office of Aerospace and Development of the US Air Force), the UK Defence Science and Technology Laboratory, the Australian Defence Science and Technology Organisation, the Canadian Defence Research Establishment Valcartier and the IEEE Intelligent Systems.

The event would not have been possible without much valuable assistance and local arrangements made by Anne Bergez, Marie Dervillers, Malik Ghallab, Félix Ingrand and Jackie Som from LAAS (Laboratoire d'Analyse et d'Architecture des Systèmes), part of CNRS (the French Centre National de la Recherche Scientifique) in Toulouse.

The organisers also would like to thank Jennie Camps-Douglas at AIAI at the University of Edinburgh for acting as the conference secretariat and for preparation of the web site and proceedings and Sylvie Barrouquere of ADERMIP (Association pour le Développement de l'Enseignement, de l'Économie et des Recherches de Midi-Pyrénées) for arrangements in Toulouse.

Copyright

Coalition Agents Experiment: Multi-Agent Co-operation in an International Coalition Setting

David N. Allsopp[1], Patrick Beautement[1], Jeffrey M. Bradshaw[2], Edmund H. Durfee[3], Michael Kirton[1], Craig A. Knoblock[4], Niranjan Suri[2], Austin Tate[5], Craig W. Thompson[6]

1. QinetiQ Ltd,
Malvern Technology Centre,
St Andrews Road, Malvern,
Worcestershire, WR14 3PS, UK
{d.allsopp, m.kirton}@signal.qinetiq.com, pbeautement@qinetiq.com

2. Institute for Human & Machine Cognition
University of West Florida
40 South Alcaniz Street
Pensacola, FL 32501, USA
{jbradshaw, nsuri}@ai.uwf.edu

3. EECS Department
University of Michigan
Ann Arbor, MI 48109, USA
durfee@umich.edu

4. Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292, USA
knoblock@isi.edu

5. Artificial Intelligence Applications Institute
Centre for Intelligent Systems and their Applications
Division of Informatics, The University of Edinburgh
80 South Bridge, Edinburgh EH1 1HN, UK
a.tate@ed.ac.uk

6. Object Services and Consulting, Inc. (OBJS)
2725 Deep Valley Trail, Plano, TX 75023, USA
thompson@objs.com

Abstract. Military Coalitions are examples of large-scale multi-faceted virtual organizations, which sometimes need to be rapidly created and flexibly changed as circumstances alter. The Coalition Agents eXperiment (CoAX) aims to show that multi-agent systems are an effective way of dealing with the complexity of real-world problems, such as agile and robust Coalition operations and enabling interoperability between heterogeneous components including legacy and actual military systems. CoAX is an international collaboration carried out under the auspices of DARPA's Control of Agent-Based Systems (CoABS) program. Building on the CoABS Grid framework, the CoAX agent infrastructure groups agents into domains that reflect real-world organizational, functional and national boundaries, such that security and access to agents and information can be governed by policies at multiple levels. A series of staged demonstrations of increased complexity are being conducted in a realistic peace-enforcement scenario situated in 2012 in the fictitious African state of "Binni". These demonstrations show how agent technologies support the rapid, co-ordinated construction of a Coalition command system for intelligence gathering, for visualization, and for campaign, battle and mission planning and execution.

1 Introduction and Background

1.1 Military Context

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; the Commander issues clear intent and then delegates control 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; human decision-making is primary and 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 Command and Battlespace Management program, the US Joint BattleSpace Infosphere program and, more generally, Network-Centric Warfare (http://www.dodccrp.org/).

In addition to the problems of integrating single-service and Joint capabilities into a coherent force, the nature of Coalition (multi-national) operations implies some need to rapidly configure foreign or ‘come-as-you-are’ systems into a cohesive whole. Many problems in this environment can only be solved by organizational changes and by ‘aligning’ doctrine, concepts of operations and procedures. Due to the inevitable absence of pre-existing co-ordinated systems, Coalition scenarios require a rapid, flexible, on-the-fly approach that 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).

Currently, Coalition operations are often characterized by data overload, information starvation, labor intensive collection and co-ordination of information, and standalone stove-pipe command systems that use incompatible data formats. This leads to a horrendous technical integration task and gives commanders only scattered snapshots of the battlespace. 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. In this paper, we show that software agents that act on behalf of human users enable military commanders to act decisively in cyberspace and thus contribute towards the achievement of ‘Cyberspace Superiority’, a critical component of warfare in the information age (Alberts et al, 2001).

1.2 Software Agent Technology

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 they need it. The agents are also able to proactively suggest courses of action, monitor mission progress, and recommend plan adjustments as circumstances unfold.

A community of agents can be seen 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 program 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 (http://coabs.globalinfotek.com/).

A recent article (Jennings, 2001) argues that the agent paradigm is a good way of building complex software systems in general, and hence offers potential benefits in the Coalition setting. For example, 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, underpinned by the CoABS Grid. The scenario used as the basis of the experiments to test this hypothesis is described in section 2.

1.3 Aims of the CoAX Project

This paper describes the progress of an international collaborative effort whose overall goals are to demonstrate that the agent-based computing paradigm offers a promising new approach to dealing with the technical issues of establishing coherent command and control (C2) in a Coalition organization. This collaborative effort, entitled CoAX (Coalition Agents eXperiment), is a Technology Integration Experiment under the auspices of DARPA's Control of Agent Based Systems (CoABS) program (http://www.aiai.ed.ac.uk/project/coax/). Specific hypotheses of the research program are that:

- agents are a useful metaphor for dealing with 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 organizations;
- domain policies can structure agent relationships and enforce Coalition policies;
- intelligent task and process management can improve agent collaboration;
- semantic web technology can improve agent interoperability between disparate Coalition command systems.

The CoAX team has built a software agent test-bed based on the CoABS Grid (http://coabs.globalinfotek.com/). This paper describes the work done, the demonstrations carried out so far, the scenario and storyboard used and some of the insights gained.

1.4 Structure of the Paper

The paper begins by describing the Coalition scenario and military command structure used in our demonstration experiments. Section 3 describes the corresponding agent architecture that was developed to reflect the military organizational structure. The events occurring in the storyboard used for the various demonstrations so far are described in Section 4. A preliminary assessment of software agent capabilities and a discussion of future research are provided in Section 5. Concluding remarks are given in Section 6.

2 A Representative Scenario and Coalition Command Structure

The CoAX work needed a suitably realistic scenario for its experiments and so we expanded the fictional "Binni" scenario (Rathmell, 1999) developed for The Technology Co-operation Programme. 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 (Figure 1). Previously uninhabited land in the Plain is now arable and the area has received large amounts of foreign investment. It is now called "The Golden Bowl of Africa".

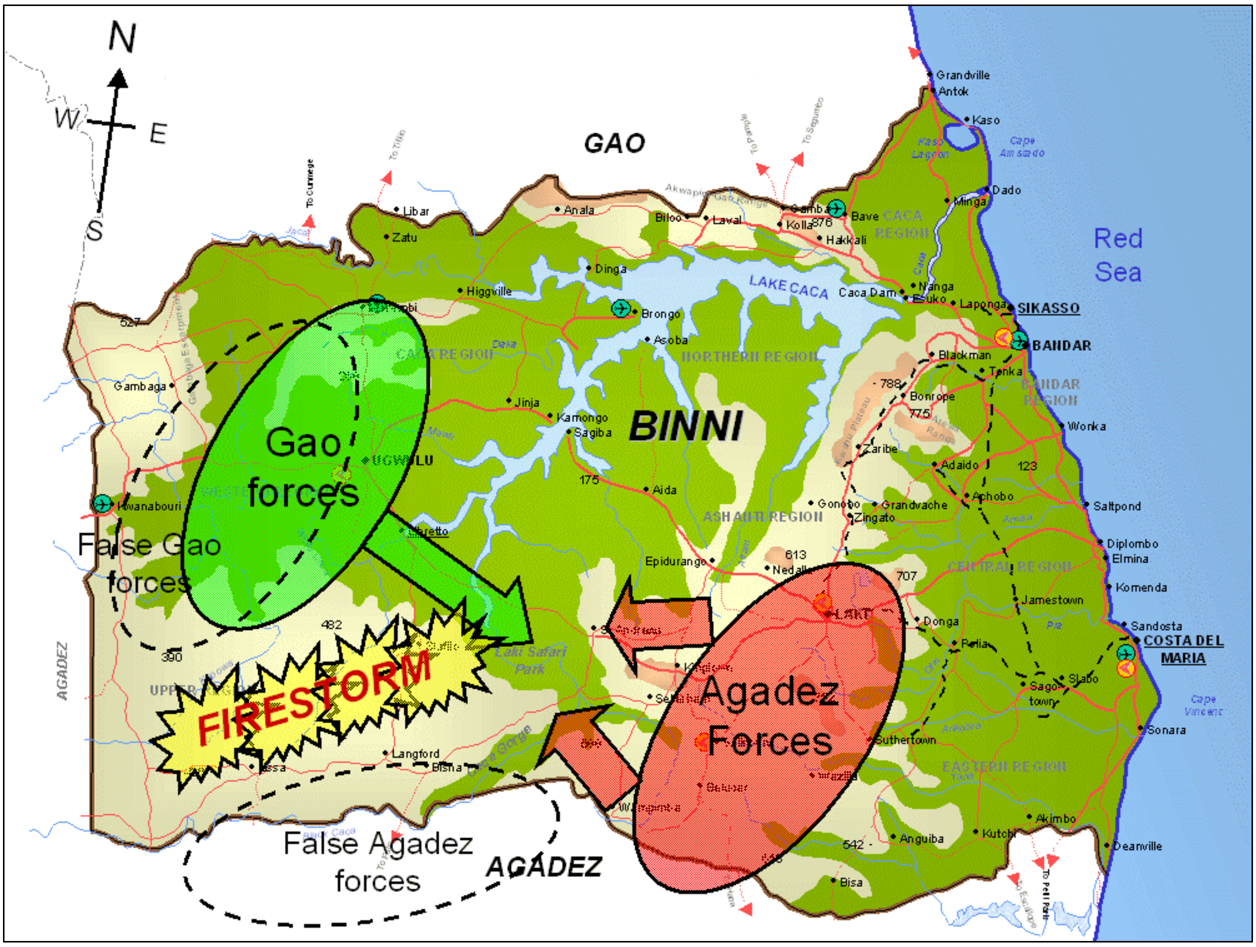


Figure 1. Map of Binni showing firestorm deception. Misinformation from Gao is intended to displace the firestorm to the west, allowing Gao and Agadez forces to clash in the region of the Laki Safari Park.

A conflict has developed between two countries in the area. 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 stabilize the region. This basic scenario was adapted for a number of CoAX demonstrations (see Section 4), beginning with the initial planning phase, then moving onto shorter timescales and more dynamic, uncertain events for the execution phase.

2.1 Coalition Command Structure

This Binni Coalition operation needs to rapidly configure various incompatible, 'come-as-you-are' or foreign systems into a cohesive whole within an open, heterogeneous and dispersed environment. This scenario provides a suitable test for the software agent experiments, where run-time composability 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 2 below.

This is a representative and realistic Coalition command structure involving the UN, Governments, Other Government Departments (OGDs, such as the Foreign Office), Non-Government Organizations (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 authority (the command chain) and accountability. This is the kind of Coalition structure that would be agreed by the participants; no part of the formal command chain is owned by any specific country. Note that the 'levels of command' overlap and their boundaries are not rigidly defined. Dashed lines show an advisory / negotiating role.

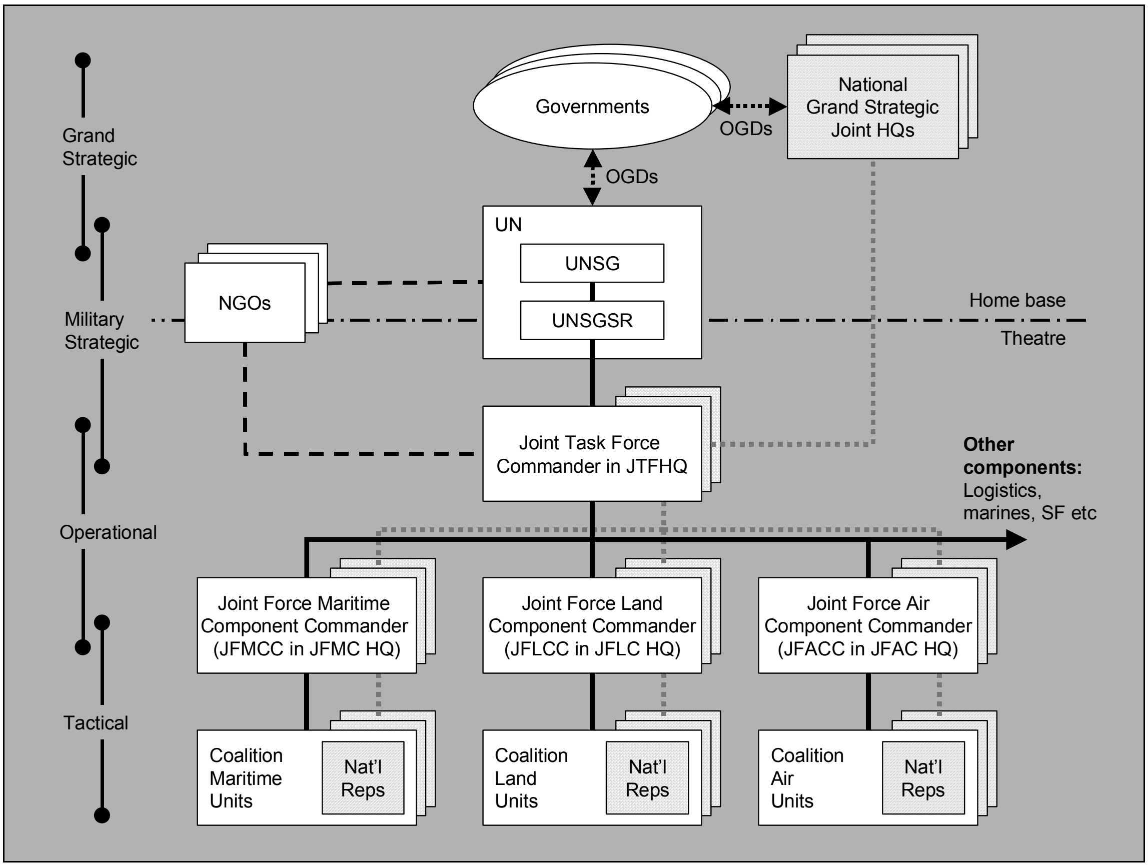


Figure 2: A representative Coalition structure, showing the chain of command down from the United Nations, including the 'ghosted' command structures of the participant nations, and Non-Government Organizations (NGOs). The approximate command level at which the various entities operate is indicated on the left.

3 Software Agent Architecture

3.1 Human Domains

Integrating 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, where many social and cultural factors come into play. The mapping between the human and technical worlds is thus not straightforward. From the human perspective, we identified four kinds of 'domains':

- **Organizational Domains**: for example the Joint Task Force HQ (JTF HQ) ;
- **Country Domains**: each of the National command chains would be a separate, self-contained domain;
- **Functional Domains**: sets of entities collaborating on common tasks, for example Meteorology or Intelligence ;
- **Individual Human Domains of Responsibility**: Commanders have responsibility for their own HQ and all subordinate ones (in practice they delegate). Hence the individual human domains of influence may overlap.

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 creates difficulties for technical systems.

3.2 Software Agent Domains

3.2.1 CoABS Grid Infrastructure

At the most basic level, the agents and systems to be integrated require infrastructure for discovery of other agents, and messaging between agents. The CoABS Grid provides this. Based on Sun's "Jini" services which are themselves based upon Java's Remote Method Invocation, the Grid allows registration and advertisement of agent capabilities, and communication by message-passing. Agents on the Grid can be added or removed, or their advertisements updated, without reconfiguration of the network. Agents are automatically purged from the registry after a short time if they fail. Multiple lookup services may be used, located by multicast or unicast protocols. In addition, the Grid provides functionality such as logging, visualization, and more recently encryption of messages and agent authentication.

3.2.2 KAoS Domain Management

The increased intelligence afforded by software agents is both a boon and a danger. By their ability to operate independently without constant human supervision, agents can perform tasks that would be impractical or impossible using traditional software applications. On the other hand, this additional autonomy, if unchecked, also has the potential of effecting severe damage to military operations in the case of buggy or malicious agents. The Knowledgeable Agent-oriented System (KAoS) provides services that help assure that agents from different developers and running on diverse platforms will always operate within the bounds of established policies and will be continually responsive to human control so that they can be safely deployed in operational settings (Bradshaw et al., 1997, 2001). KAoS services and tools are intended to allow for the specification, management, conflict resolution, and enforcement of policies within the specific contexts established by complex military organizational structures.

KAoS domain management services can be used to group agents into logical domains corresponding to organizational structures, administrative groups, and task-oriented teams. Within CoAX, these domains mirror the human domains described above, allowing for complex hierarchical, heterarchical, and overlapping structures. An agent domain consists of a unique instance of a domain manager (DM) along with any agents that are registered to it. Alternatively, an intensionally-defined domain consists of a set of agents sharing one or more common properties (e.g., the domain of all agents physically residing on some host). The function of a domain manager is to manage agent registration, and serve as a single point of administration and enforcement for domain-wide, host-wide, VM-wide, VM-container-wide, or agent-specific policies.

3.2.3 Domain policies

A policy is a declarative constraint governing the behavior of one or more agents, even when those agents may not be domain-aware or where they may be buggy or malicious. For example, a policy may be declared that all messages exchanged among agents in the JFAC HQ domain must be encrypted, or that an agent cannot simultaneous belong to the US and the UK domain. A policy does not tell the agent how to perform its task; it rather specifies the conditions under which certain actions can be performed. By way of an analogy to traffic management, it is more like a set of individually-customizable stop signs and highway patrol officers that define and enforce the rules of the road than it is like a route planner that helps agents find their way to their destinations.

Policies governing authorization, encryption, access control, and resource control are part of KAoS domain management. However, due to our focus on agent systems our scope goes beyond these typical security concerns in significant ways. For example, KAoS pioneered the concept of agent conversation policies (Bradshaw et al., 1997). Teams of agents can be formed, maintained, and disbanded through the process of agent-to-agent communication using an appropriate semantics. In addition to conversation policies, we are developing representations and enforcement mechanisms for mobility policies, domain registration policies, and various forms of obligation policies. These policies are represented in ontologies using the DARPA Agent Markup Language (DAML), and an efficient description logic-based approach is used as the basis for much of the domain manager's reasoning to discover and resolve policy conflicts and to perform other kinds of policy analysis.

The separation of policy specification from policy-enforcement mechanisms allows policies to be dynamically re-configurable, 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 behavior in agent systems includes the following claims: easier recognition of non-normative behavior, policy reuse, operational efficiency, ability to respond to changing conditions, and the possibility of off-line verification.

3.3 Software Agent Domains in CoAX

The CoAX demonstrations contain software agents grouped into agent domains using the CoABS Grid, with the policies enforced by KAoS domain management services. A typical domain configuration is shown in Figure 3.

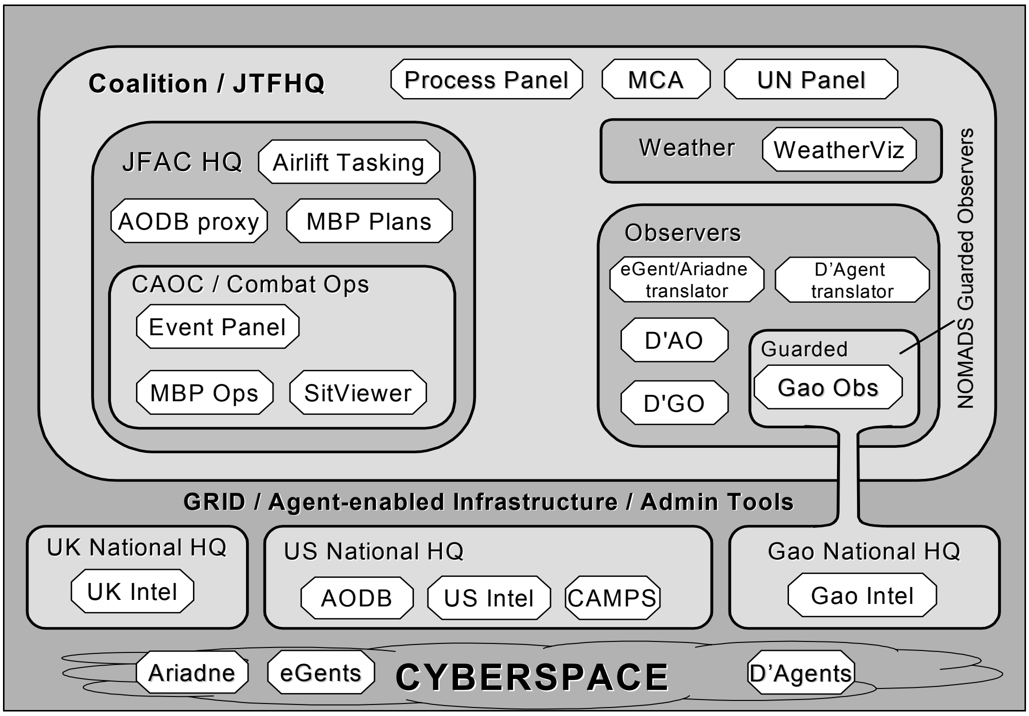


Figure 3. Typical CoAX domain structure; domains are indicated by rounded rectangles; agents by angled rectangles. Some agents are proxies for agents or legacy systems that are not themselves domain aware. Each domain would also contain a Domain Manager agent and a Matchmaker agent (omitted for clarity). Nesting of domains indicates a hierarchy of responsibility and policy control. The agent acronyms are expanded in the body text.

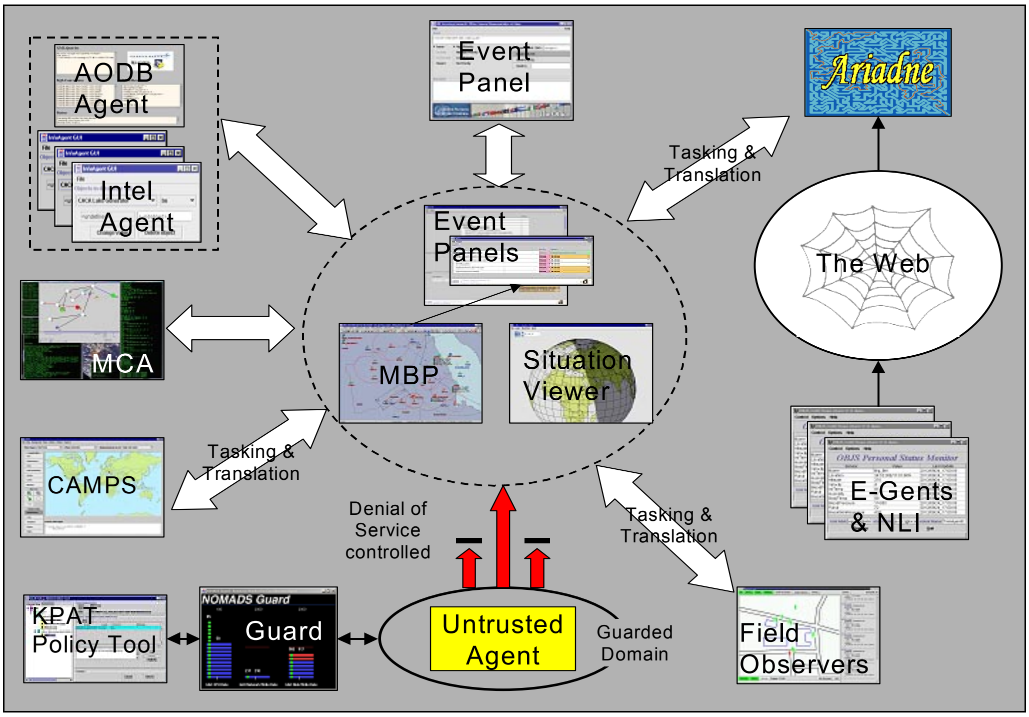


Figure 4: Overview of technologies and agents. The central visualization and planning tools find and acquire data (e.g. disposition of ground forces) and services (e.g. airlift scheduling and plan deconfliction) from the other agents and systems, in some cases via intermediate tasking and translation agents. MBP = Master Battle Planner, MCA = Multi-level Coordination Agent, KPAT = KAoS Policy Admin Tool, AODB = Air Operations Data Base, NLI = Natural Language Interface, CAMPS = Consolidated Air Mobility Planning System.

4 Demonstration Storyboard and Technologies

In this section we progress through the storyboard created for the Binni Scenario, and describe each of the agent systems and technologies brought into play for each part of the story. An overview of the interactions from the agent/system point of view is shown in Figure 4.

4.1 Population of Domains

Following the outbreak of hostilities, the UN has deployed their UN War Avoidance Force for Binni (UNWAFB), to stabilize the region. The active Coalition participants at this time are the UK, US and Gao.

In agent terms, a variety of agent domains are set up using the CoABS Grid infrastructure and the KAoS domain management services, representing the organizational structures (the JTF HQ and the JFAC HQ), the nations (UK, US, Gao) and various functional domains such as Weather and Observers. These domains are populated with a number of agents, which register with their Domain Manager and optionally advertise their services with their domain Matchmaker.

4.2 Data Gathering and Air Planning

After exploring options to separate the opposing forces and restore the peace in the region, the deployment of a large ground observation and peace enforcement force and other courses of action have been rejected, and a 'Firestorm' mission has been decided upon. This will clear land to enable simpler remote and ground observations with less risk to the Coalition peacekeepers. The Coalition undertakes initial information gathering and planning.

4.2.1 Master Battle Planner (MBP)

Air planning at the JFAC is performed using QinetiQ's MBP, a highly effective visual planning tool for air operations. MBP assists planners by providing them with an intuitive visualization on which they can manipulate the air intelligence information, assets, targets and missions, using a map-based graphical user interface (Figure 5). This enables an operator to build a battle scenario containing targets, offensive and defensive units, airspace measures and other objects using simple dialogs and point-and-click techniques on the map. Objects on the map can then be moved around, and their properties can be changed. Information such as the allegiance and status of units, and the ranges of units may also be displayed.

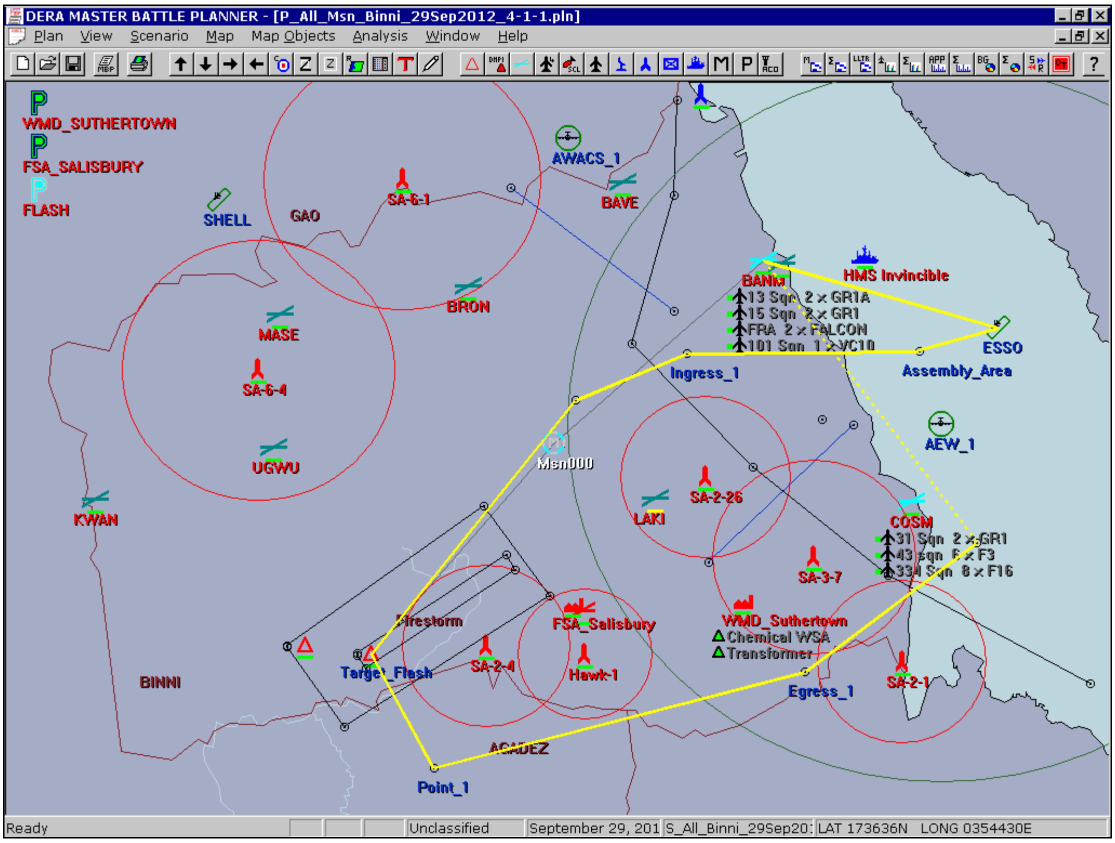


Figure 5: Master Battle Planner map display of the fictional countries of Binni, Gao, and Agadez. A selected mission is highlighted, proceeding from an airbase (BANM), to refueling tanker (ESSO), to the target via waypoints and airspaces, and back to base by a different route.

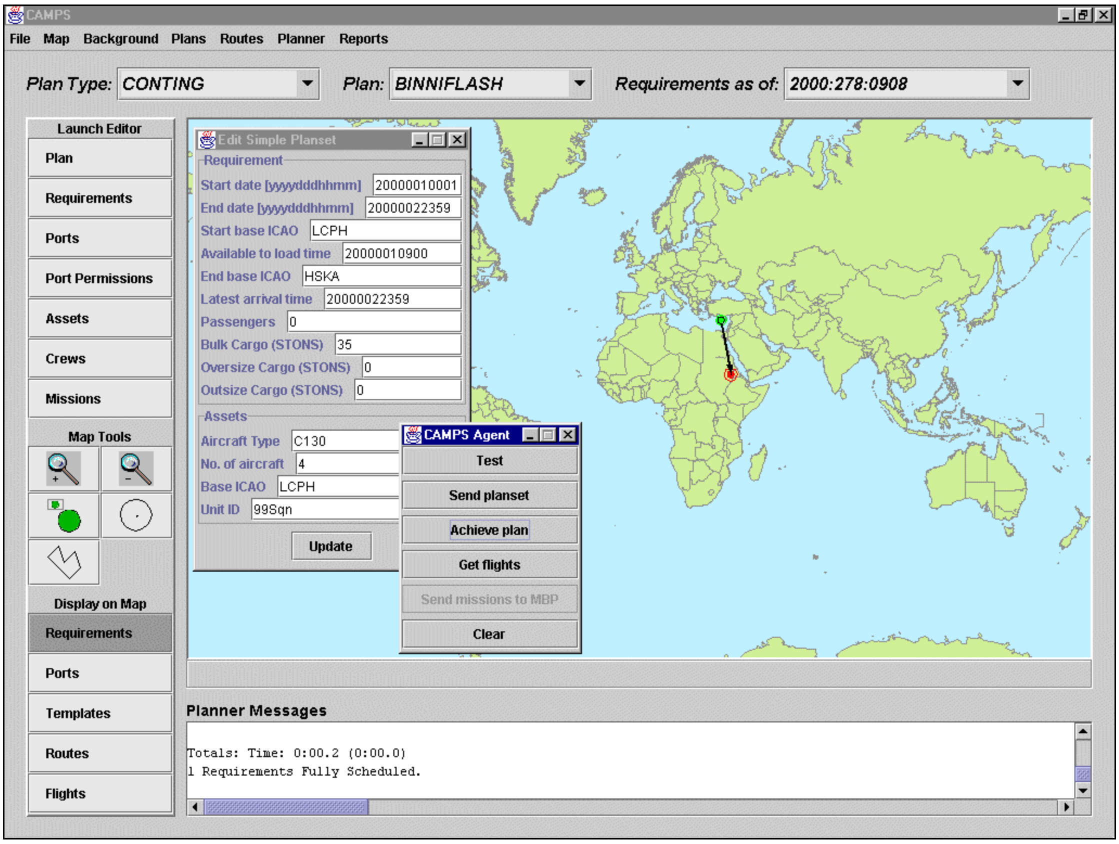
The operator can interact with these entities and can plan individual air missions (or more complex packages of missions) by dragging and dropping offensive units onto targets on the map. Supporting / defensive elements are added in the same way. The system gives the operator analytical tools to assess the planned air operations for:

- the best utilization of resources; (e.g. by highlighting air units that are over-tasked);
- time-phasing (through charts and animated 'fly-out');
- concordance with the military guidance given.

MBP is a monolithic C++ application, which has been agent-enabled by wrapping it in Java code, using the Java Native Interface. The agent-enabling of MBP allows it to receive all the scenario data (targets, assets, airspaces etc.) from multiple information-providing agents ('Intel Agents' — see Figure 4) and update this information in near-real time. Importantly, this process is integrated into the normal usage of MBP; when an operator views the status of an object, agents are automatically tasked to update the information. Agents may also 'push' changes of status to MBP. Information concerning other air missions can be accepted and merged with missions planned within MBP, as described below. Missions can also be saved and exported, enabling other agents to reason with the data.

4.2.2 Consolidated Air Mobility Planning System (CAMPS)

The second real military system integrated into the demonstration is the Air Force Research Laboratory's CAMPS Mission Planner (Figure 6). CAMPS develops schedules for aircraft to pick up and deliver cargo within specified time windows. It takes into account constraints on aircraft capabilities, port handling capabilities, crew availability and work schedule rules, etc. Users of the planner develop plans (schedules) for aircraft to carry a particular cargo, specifying the intermediate ports, air refueling tracks and the kinds of crews that will be available. They can also specify a number of constraints on the airports potentially involved in the plans to be developed (Emerson & Burstein, 1999; Burstein et al, 2000).



Plan Type: CONTING

Plan: BINNIFLASH

Requirements as of: 2000:278:0908

Requirement

Start date [yyyydddhhmm] 20000010001

End date [yyyydddhhmm] 20000022359

Start base ICAO LCPH

Available to load time 20000010900

End base ICAO HSKA

Latest arrival time 20000022359

Passengers 0

Bulk Cargo (STONS) 35

Oversize Cargo (STONS) 0

Outsize Cargo (STONS) 0

Assets

Aircraft Type C130

No. of aircraft 4

Base ICAO LCPH

Unit ID 99Sqn

```
Totals: Time: 0:00.2 (0:00.0)
1 Requirements Fully Scheduled.
```

Figure 6: The CAMPS airlift planner, and the demonstration agent used to task the CAMPS agent with a simple requirement: movement of cargo from Cyprus into the fictional country of Binni.

In the demonstration scenario, CAMPS schedules airlifts of cargo into Binni. These airlift flights could conflict with offensive air missions, so the scheduled flights are requested from the CAMPS agent, and sent to MBP, forming part of the normal MBP air visualization. This is achieved by an intermediate agent which tasks CAMPS, and also translates between the KQML messages used by CAMPS and the XML messages used by the MBP agent.

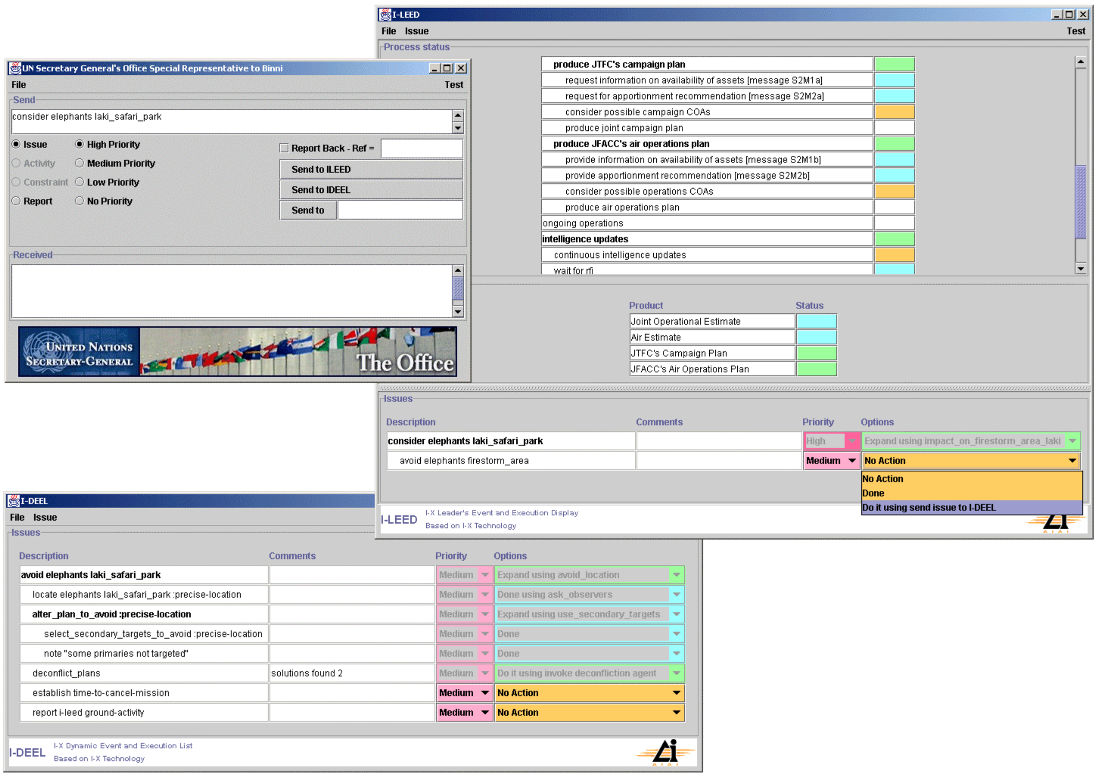
This is an interesting example, as only partial translation is possible; CAMPS and MBP differ fundamentally in their definition of air missions. A CAMPS mission consists of an arbitrary collection of flights, where a flight is a single journey from A to B by a single aircraft. However, an MBP mission consists of a starting point and a route, which must return to the starting point (perhaps by a different path), and may consist of multiple aircraft. CAMPS can therefore produce routes that have no fully valid representation in MBP, although they could be partially represented or indicated graphically.

4.2.3 Ariadne

In a similar manner, weather information extracted from websites by the Ariadne system from the University of Southern California, Information Sciences Institute, is translated and forwarded to MBP, again forming part of the normal picture of the air situation. Ariadne facilitates access to web-based data sources via a wrapper / mediator approach (Knoblock and Minton, 1998). Wrappers that make web sources look like databases can be rapidly constructed; these interpret a request (expressed in SQL or some other structured language) against a web source and return a structured reply. The mediator software answers queries efficiently using these sources as if they formed a single database. Translation of the XML from Ariadne into the XML expected by MBP was handled by custom code, but can now be performed more easily using XSLT (Extensible Stylesheet Language Transformations); this latter technique is used elsewhere in the demonstration (section 4.2.6).

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

This Coalition planning process is supported using I-X process panels. I-X is a research program with a number of different aspects intended to create a well-founded approach to allow humans and computer systems to cooperate in the creation or modification of some product such as a plan, design or physical entity — i.e. it supports synthesis tasks. I-X may also be used to support more general collaborative activity. The I-X research draws on earlier work on O-Plan (Tate et al, 1998), <I-N-OVA> (Tate, 1996) and the Enterprise Project (Fraser and Tate, 1995) but seeks to make the framework generic and to clarify terminology, simplify the approach taken, and increase re-usability and applicability of the core ideas. Within CoAX, the I-X approach is being used to provide task and process support and event-response capabilities to various Coalition participants (Figure 7).


UN Secretary General's Office Special Representative to Binni
File
Test
Send
Received
UNITED NATIONS SECRETARY-GENERAL
The Office
I-LEED
File Issue
Test
Process status
Issues
I-LEED I-X Leader's Event and Execution Display
Based on I-X Technology
I-DEEL
File Issue
Issues
I-DEEL I-X Dynamic Event and Execution List
Based on I-X Technology

consider elephants laki_safari_park

● Issue ○ Activity ○ Constraint ○ Report

● High Priority ○ Medium Priority ○ Low Priority ○ No Priority

□ Report Back - Ref =

Send to ILEED

Send to IDEEL

Send to

produce JTFC's campaign plan	🟩
request information on availability of assets [message S2M1a]	🟦
request for apportionment recommendation [message S2M2a]	🟦
consider possible campaign COAs	🟧
produce joint campaign plan	
produce JFACC's air operations plan	🟩
provide information on availability of assets [message S2M1b]	🟦
provide apportionment recommendation [message S2M2b]	🟦
consider possible operations COAs	🟧
produce air operations plan	
ongoing operations	
intelligence updates	🟩
continuous intelligence updates	🟧
wait for rfi	🟦

Product	Status
Joint Operational Estimate	🟦
Air Estimate	🟦
JTFC's Campaign Plan	🟩
JFACC's Air Operations Plan	🟩

Description	Comments	Priority	Options
consider elephants laki_safari_park		High	Expand using impact_on_firestorm_area_laki
avoid elephants firestorm_area		Medium	No Action

No Action
Done
Do it using send issue to I-DEEL

Description	Comments	Priority	Options
avoid elephants laki_safari_park		Medium	Expand using avoid_location
locate elephants laki_safari_park :precise-location		Medium	Done using ask_observers
alter_plan_to_avoid :precise-location		Medium	Expand using use_secondary_targets
select_secondary_targets_to_avoid :precise-location		Medium	Done
note "some primaries not targeted"		Medium	Done
deconflict_plans	solutions found 2	Medium	Do it using invoke deconfliction agent
establish time-to-cancel-mission		Medium	No Action
report i-leed ground-activity		Medium	No Action

Figure 7: I-X Process and Event Panels

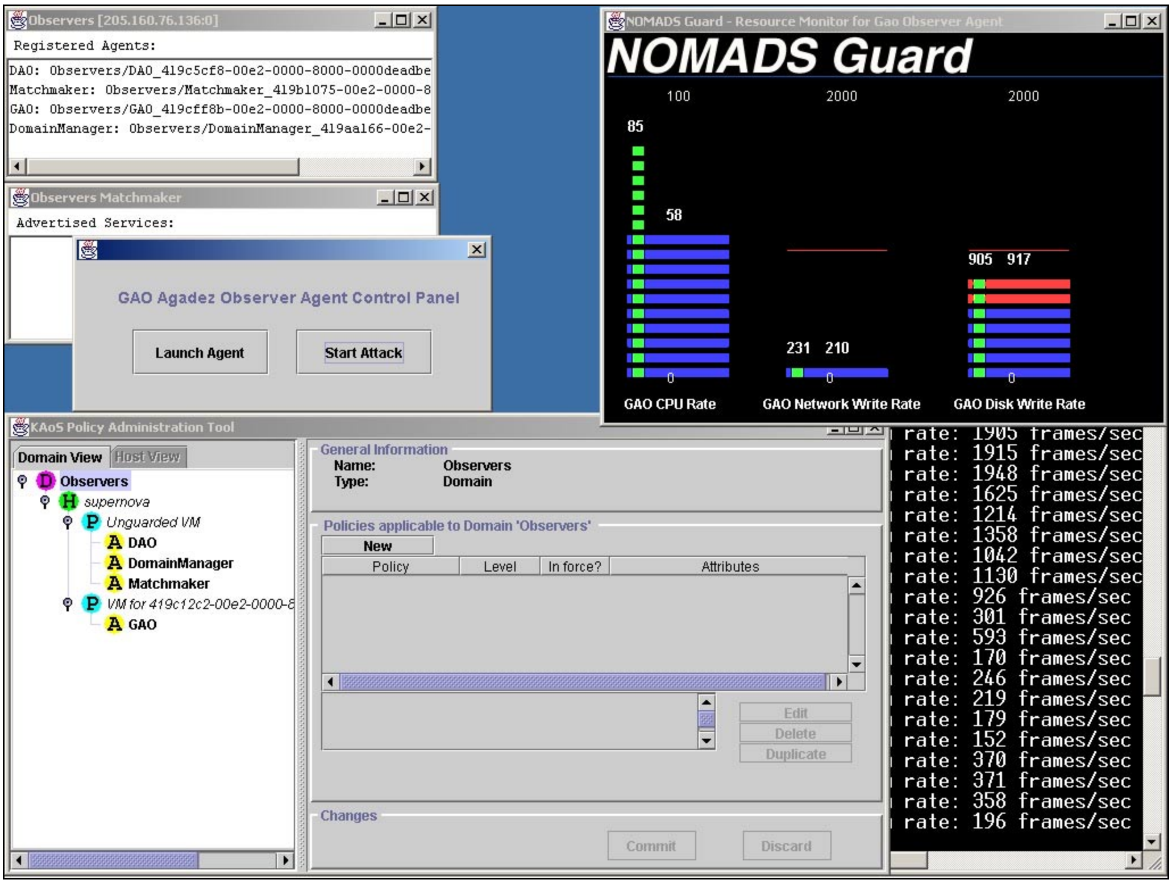
The aim of an I-X Process Panel (I-P^2) is to act as a workflow and messaging 'catch all' for its user. It can act in conjunction with other panels for other users if desired. A panel:

- Can take *any* requirement to:
  - Handle an issue;
  - Perform an activity;

  - (in future) Add a constraint.
- Deals with these via:
  - Manual (user) activity;
  - Internal capabilities;
  - External capabilities (invoke or query);
  - Reroute or delegate to other panels or agents (pass);
  - Plan and execute a composite of these capabilities (expand).
- Receives reports and interprets them to:
  - Understand current status of issues, activities and constraints;
  - Understand current world state, especially status of process products;
  - Help the user control the situation.
- Copes with partial knowledge.

4.2.5 Resource control via domain policies

Gao has host nation status within the Coalition but its intentions are unclear and it is distrusted. Special steps are taken to monitor the information passed to and from Gao within the Coalition. During the demonstration, misinformation feeds by Gao (intended to displace the firestorm to allow Gao to take an advantage and move forward) are detected and thwarted. Gao becomes belligerent and launches a denial of service attack against the Coalition's C3I infrastructure.



Observers [205.160.76.136:0]

```
Registered Agents:
DAO: Observers/DAO_419c5cf8-00e2-0000-8000-0000deadbe
Matchmaker: Observers/Matchmaker_419b1075-00e2-0000-8
GAO: Observers/GAO_419cff8b-00e2-0000-8000-0000deadbe
DomainManager: Observers/DomainManager_419aa166-00e2-
```

Observers Matchmaker

```
Advertised Services:
```

GAO Agadez Observer Agent Control Panel

NOMADS Guard - Resource Monitor for Gao Observer Agent

KAoS Policy Administration Tool

General Information
Name: Observers
Type: Domain

Policies applicable to Domain 'Observers'

Policy	Level	In force?	Attributes

Changes

```
rate: 1905 frames/sec
rate: 1915 frames/sec
rate: 1948 frames/sec
rate: 1625 frames/sec
rate: 1214 frames/sec
rate: 1358 frames/sec
rate: 1042 frames/sec
rate: 1130 frames/sec
rate: 926 frames/sec
rate: 301 frames/sec
rate: 593 frames/sec
rate: 170 frames/sec
rate: 246 frames/sec
rate: 219 frames/sec
rate: 179 frames/sec
rate: 152 frames/sec
rate: 370 frames/sec
rate: 371 frames/sec
rate: 358 frames/sec
rate: 196 frames/sec
```

Figure 8: A denial-of-service attack by the Gao agent is starving other agents of resources (note the decreasing rate of processing in the console, bottom right). The Guard (top right) is monitoring the resource usage of the Gao agent. The excessive resource usage triggers a change in domain policy, and the resource limits enforced by the AromaVM are lowered. The policy can also be changed manually using KPAT, the KAoS Policy Administration Tool (bottom left).

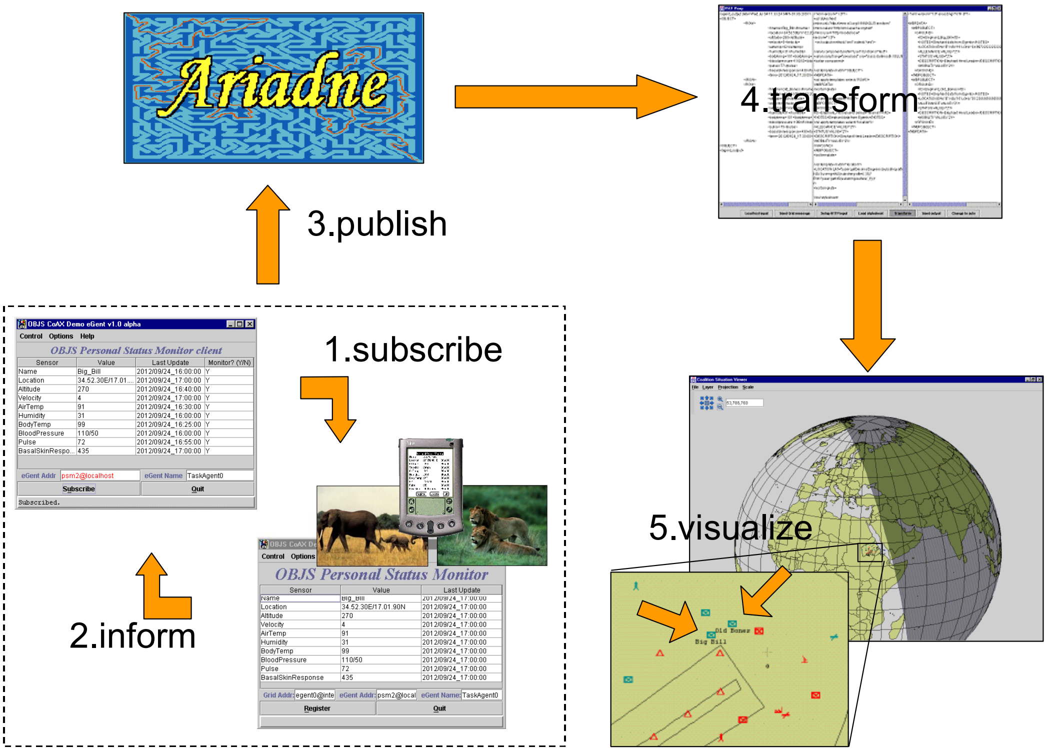
The Gao agent in the demonstration is run under NOMADS, a mobile agent system from IHMC. The NOMADS project aims to develop a set of distributed agent-based systems using the Java language and environment. The agent code runs in a new Java Virtual Machine, the AromaVM. The AromaVM provides two key enhancements over standard Java VMs: the ability to capture the execution state of threads and the ability to control resources consumed by threads. By capturing the execution state of threads, the NOMADS agent system provides strong or transparent mobility for agents.

In addition, the resource control mechanisms can be used for controlling and allocating resources used by agents as well as to protect against denial of service attacks by malicious agents. When the Gao agent exceeds certain resource limits, an automatic change in domain policy is triggered by a domain Guard, and the AromaVM is instructed to reduce the resources available to the malicious agent (Figure 8). An operator can manually reduce the limits further, using the KAoS Policy Admin Tool (KPAT).

4.2.6 Data feeds from mobile devices and observers

The firestorm mission has been planned and aircraft have already taken off. However the news media break a story that wildlife in an important safari park in Binni may be in danger as the park overlaps the firestorm area. With only an hour to go, the UN Secretary General's Special Representative to Binni asks the Joint Task Force Commander to consider the wildlife risk aspects of the planned approach. Dynamic information gathering and information feeds using agent technology are employed to create a real time feed of the position of some at-risk large mammals.

This urgent issue is noted and broken down into sub-tasks using the event panels. The progress of aircraft is monitored in near real-time on the Situation Viewer agent from QinetiQ, and the time left before aircraft are committed to their targets is determined from MBP. A search is made for information on the locations of animals in the safari park, and it is discovered that data are available on-line via agents running on monitoring devices attached to large mammals in the park. The agents are eGents (agents that communicate by email) developed by Object Services and Consulting, Inc (OBJS). Historical data from these devices is queried using a Natural Language Interface from OBJS. To aid the planners, a live data-feed is created from the safari park website, using Ariadne to extract data from the pages, and a translator agent using XSLT. The resulting message stream is sent to MBP and to the Situation Viewer agent, allowing the current position and track of the animals to be visualized (Figure 9).



OBJS Personal Status Monitor client

Sensor	Value	Last Update	Monitor? (Y/N)
Name	Big_Bill	2012/09/24_16:00:00	Y
Location	34.52.30E/17.01....	2012/09/24_17:00:00	Y
Altitude	270	2012/09/24_16:40:00	Y
Velocity	4	2012/09/24_17:00:00	Y
AirTemp	91	2012/09/24_16:30:00	Y
Humidity	31	2012/09/24_16:00:00	Y
BodyTemp	99	2012/09/24_16:25:00	Y
BloodPressure	110/50	2012/09/24_16:00:00	Y
Pulse	72	2012/09/24_16:55:00	Y
BasalSkinRespo...	435	2012/09/24_17:00:00	Y

eGent Addr: psm2@localhost
eGent Name: TaskAgent0

Subscribed.

OBJS Personal Status Monitor

Sensor	Value	Last Update
Name	Big_Bill	2012/09/24_17:00:00
Location	34.52.30E/17.01.90N	2012/09/24_17:00:00
Altitude	270	2012/09/24_17:00:00
Velocity	4	2012/09/24_17:00:00
AirTemp	91	2012/09/24_17:00:00
Humidity	31	2012/09/24_17:00:00
BodyTemp	99	2012/09/24_17:00:00
BloodPressure	110/50	2012/09/24_17:00:00
Pulse	72	2012/09/24_17:00:00
BasalSkinResponse	435	2012/09/24_17:00:00

Grid Addr: egent0@inte
eGent Addr: psm2@local
eGent Name: TaskAgent0

Figure 9: An eGent client subscribes to eGents running on mobile devices (wildlife tags). The data from these devices are published by the client on a web page. Ariadne extracts data from the webpages, and produces XML. The XML is transformed to another format by another agent using XSL Transformations, and finally sent to agents such as MBP and Situation Viewer for visualization.

Data about the movement of ground forces, from the D'Agents field observation system from Dartmouth College, are also transformed using another instance of the translator agent and visualized in the same way, allowing the coalition to identify a convergence of hostile forces on the Laki safari park area.

4.2.7 Plan export and deconfliction

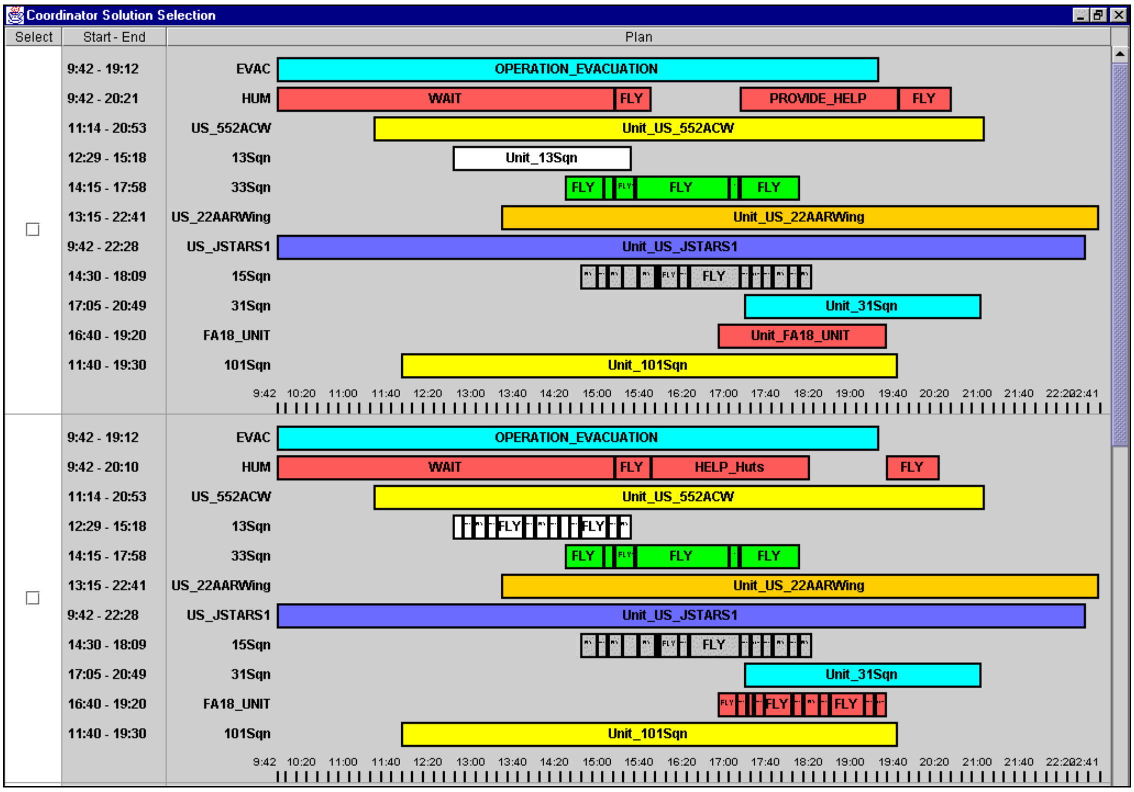
After consideration it is decided to continue with the firestorm mission, but to re-plan as necessary to avoid risk to wildlife. Firestorm targets are adjusted in time or secondary targets selected as necessary for the first wave of firestorm bombing. The impacts of these changes on the Coalition's medical and humanitarian operations are automatically detected, and unintended conflicts between disjoint Coalition operations are avoided.

The air plans are revised using MBP, and then sent to a deconfliction agent to check them against planned activities in other Coalition HQs. The Multi-level Coordination Agent (MCA) from the University of Michigan processes the plans, using multiple levels of abstraction to generate solutions (Clement & Durfee, 1999). The planners are kept informed of progress via their I-X event panels, and can view the results on the MCA display when ready (figure 10). The plans are adjusted iteratively until the conflicts are resolved.

4.2.8 Dynamic Forced Migration (Scram) of Observer Agents

Agadez seeks to use this complication to seize the initiative and launches fighter attacks against a Coalition airborne high value asset (JSTARS) that is monitoring the operation. When this attack is detected, the JSTARS starts to regress, which implies that the observer agents on the JSTARS will not be able to continue providing information to the coalition.

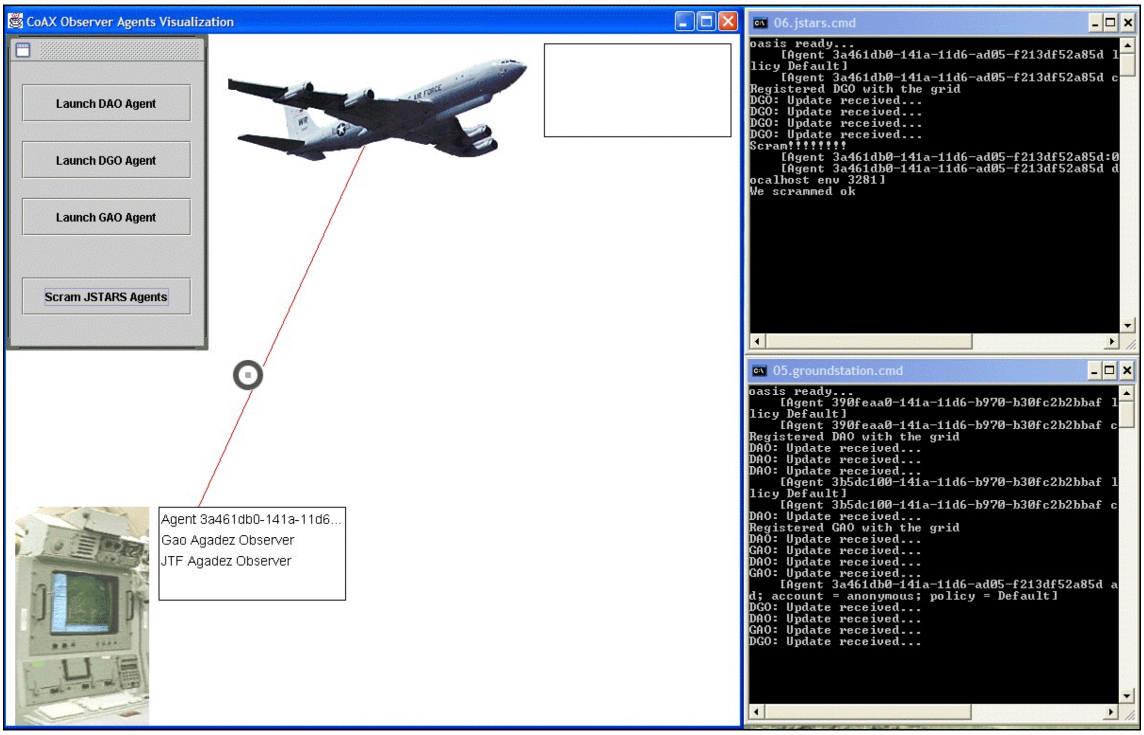
In order to solve this problem, the administrator uses the forced migration (scram) capabilities of the NOMADS mobile agent system to move the observer agents from the JSTARS platform to a secondary ground station platform. The NOMADS system uses the state capture mechanisms in the Aroma VM to capture the full execution state of the agents on the JSTARS. Once captured, the execution state is sent to a new platform where the agents can be restarted without any loss of their ongoing computations (figure 11). This allows the observation agents to continue to operate on the ground station and provide information to the coalition even after the JSTARS regresses.



Coordinator Solution Selection

Select	Start - End	Plan
	9:42 - 19:12	EVAC
	9:42 - 20:21	HUM
	11:14 - 20:53	US_552ACW
	12:29 - 15:18	13Sqn
	14:15 - 17:58	33Sqn
□	13:15 - 22:41	US_22AARWing
	9:42 - 22:28	US_JSTARS1
	14:30 - 18:09	15Sqn
	17:05 - 20:49	31Sqn
	16:40 - 19:20	FA18_UNIT
	11:40 - 19:30	101Sqn
	9:42 - 19:12	EVAC
	9:42 - 20:10	HUM
	11:14 - 20:53	US_552ACW
	12:29 - 15:18	13Sqn
	14:15 - 17:58	33Sqn
□	13:15 - 22:41	US_22AARWing
	9:42 - 22:28	US_JSTARS1
	14:30 - 18:09	15Sqn
	17:05 - 20:49	31Sqn
	16:40 - 19:20	FA18_UNIT
	11:40 - 19:30	101Sqn

Figure 10: Deconfliction of Coalition plans by the Multi-level Coordination Agent. In the second solution (lower half) two missions (13Sqn and the FA18_UNIT) have been broken down to a lower level of abstraction to seek more optimal coordination



```
oasis ready...
     [Agent 3a461db0-141a-11d6-ad05-f213df52a85d l
licy Default]
     [Agent 3a461db0-141a-11d6-ad05-f213df52a85d c
Registered DGO with the grid
DGO: Update received...
DGO: Update received...
DGO: Update received...
DGO: Update received...
Scram!!!!!!!!
     [Agent 3a461db0-141a-11d6-ad05-f213df52a85d:0
     [Agent 3a461db0-141a-11d6-ad05-f213df52a85d d
ocalhost env 3281]
We scrammed ok
```

```
oasis ready...
     [Agent 390feaa0-141a-11d6-b970-b30fc2b2bbaf l
licy Default]
     [Agent 390feaa0-141a-11d6-b970-b30fc2b2bbaf c
Registered DAO with the grid
DAO: Update received...
DAO: Update received...
DAO: Update received...
     [Agent 3b5dc100-141a-11d6-b970-b30fc2b2bbaf l
licy Default]
     [Agent 3b5dc100-141a-11d6-b970-b30fc2b2bbaf c
DAO: Update received...
Registered GAO with the grid
DAO: Update received...
GAO: Update received...
DAO: Update received...
GAO: Update received...
     [Agent 3a461db0-141a-11d6-ad05-f213df52a85d a
d; account = anonymous; policy = Default]
DGO: Update received...
DAO: Update received...
GAO: Update received...
DGO: Update received...
```

Figure 11: Forced migration of observer agents from mobile platform to ground station, using NOMADS and AromaVM. The updates from the DGO agent, initially on the JSTARS airborne platform (top right console) then start to appear on the new ground platform (lower right console).

5 Assessment of Software Agents

5.1 Technical Progress to Date

The CoAX project officially began in February 2000 and we believe that the demonstrations we have undertaken corroborate the hypotheses outlined in Section 1.3, 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, including both the planning and execution phases of operations.

The CoABS Grid and KAoS domain management capabilities have allowed us to interoperate, for the first time, previously stand-alone US and UK military systems as well as a variety of agent-based information resources. In particular, the CoABS Grid has played a vital role in rapid and robust integration of systems. We have shown how agent organization, behavior, security and resources can be managed by explicit domain policy control.

Assessment work funded by the DARPA CoABS program has reported favorably on the performance issues of agent-enabled infrastructures and the experiences of the CoAX team have shown that the agent-wrapping of legacy systems and the integration of different agent systems at short notice is relatively straightforward. This task is simpler where systems expose more of their internal information and methods. 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. 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.

Dynamic task, process and event handling is an important aspect of collaboration and Coalition C2. In the CoAX demonstrations a process panel was used to indicate the start of the tasking and lead into the heart of the demonstration. In the execution phase of operations, process panels in the main commands or headquarters were more extensively used as they enabled a clearer military relevant view of what was happening between the agents in less technical language than would otherwise be visible. Process and event panels have been found to be helpful in keeping users informed of the current stage of collaboration, and maintaining a shared picture of the current state of the collaborative efforts.

Our experience is that an agent-enabled environment gives the ability to create shared understanding and improved visualization. Specific benefits were gained when agents worked semi-autonomously in the background to process information and support decision making collaboratively with operators, and when agents were integrated into existing tools so as not to disrupt familiar methods of operation.

5.2 Future Research Program

An aim only partially addressed in the current work is the construction and maintenance of a fully dynamic virtual Coalition organization. This would involve:

- domains and agents added to the Coalition structure 'on-the-fly';
- Coalition partners joining / leaving unpredictably;
- handling of dynamic Coalition tasks, processes and events.

Capabilities under investigation for future demonstrations include

- obligation management, e.g. ensure that agents are meeting their commitments;
- improved agent collaboration and run-time interoperability achieved using semantic web languages and technology (Allsopp et al, 2001a);
- richer domain organization and security policies (Bradshaw et. al., 2001);
- richer task, process and event management with more dynamically determined agent relationships (Tate et al., 2002);
- a variety of agents providing new types of data, and data-processing capabilities such as threat classification and track prediction.

Aspects of this work will be included in the Fleet Battle Experiment-Juliet 2002, part of the Millennium Challenge joint integrating experiment.

5.3 Military Implications of the Results

The CoAX research program has shown how software agents can carry out tasks that enable interoperability between information systems and infrastructure services brought together in a 'come-as-you-are' Coalition.

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.

Thus an agent-enabled environment helps create shared understanding and improves the situational awareness of military commanders. Moreover, it could make a significant contribution to the aims of Network-Centric Warfare which is defined as follows: an approach to the conduct of warfare that derives its power from the effective linking or networking of the warfighting enterprise. It is characterized by the ability of geographically dispersed forces to create a high level of shared battlespace awareness that can be exploited via self-synchronization and other network-centric operations to achieve commander's intent.

One early lesson has been that Cyberspace 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) by ensuring that Coalition forces are able to act decisively through software agents acting on behalf of or mediating the actions of human users.

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.

6 Concluding Remarks

The central hypothesis being investigated in CoAX is that the agent-based computing paradigm is a good fit to the kind of computational support needed in Coalition operations. The evidence so far confirms this view: we have shown a number of disparate agent systems working together in a realistic Coalition application and indicated the value of the agent-based computing paradigm for rapidly creating such agent organizations. Agents can usefully share, and manage access to, information across a stylized Coalition architecture.

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 organizations — with Coalitions being an archetypal and timely example of an area where this technology is vitally needed.

Acknowledgements

The authors gratefully acknowledge all those who contributed to the CoAX project, including Mark Burstein, Thom Bartold, Maggie Breedy, John Carson, Jeff Cox, Brad Clement, Rob Cranfill, Jeff Dalton, Pete Gerken, Bob Gray, Arne Grimstrup, Paul T. Groth, Greg Hill, Heather Holmback, Renia Jeffers, Martha Kahn, Shri Kulkarni, John Levine, Jean Oh, Pradeep Pappachan, Shahrukh Siddiqui, Jussi Stader, and Andrzej Uszok. The various projects that participated in CoAX were sponsored by the Defense Advanced Research Projects Agency (DARPA) and managed by the U.S. Air Force Research Laboratory, except work by QinetiQ, which was carried out as part of the Technology Group 10 of the UK Ministry of Defence Corporate Research Programme. The US Government and the contributors' organizations are authorized to reproduce and distribute reprints 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 express or implied, of DARPA, the US Government, the US Air Force Research Laboratory, the UK MoD, or the contributors' organizations.

References

Alberts, D. S., Garstka, J.J., Hayes, R.E., Signori, D. A. (2001) "Understanding Information-Age Warfare", CCRP Publication Series, 2001. ISBN 1-893723-04-6

Allsopp, D.N., Beautement, P., Bradshaw, J.M., Carson, J., Kirton, M., Suri, N. and Tate, A. (2001) "Software Agents as Facilitators of Coherent Coalition Operations", Sixth International Command and Control Research and Technology Symposium, US Naval Academy, Annapolis, Maryland, USA, 19-21 June 2001.

Allsopp, D.N., Beautement, P., Carson, J. and Kirton, M. (2001a) "Toward Semantic Interoperability in Agent-based Coalition Command Systems", Proceedings of the First International Semantic Web Workshop, July 30-31, 2001, Stanford University, CA, USA, pp 209-228

Bradshaw, J.M., Suri, N., Kahn, M., Sage, P., Weishar, D. and Jeffers, R. (2001) "Terraforming Cyberspace: Toward a Policy-based Grid Infrastructure for Secure, Scalable, and Robust Execution of Java-based Multi-agent Systems". IEEE Computer, 49-56, July 2001.

Bradshaw, J.M., Dutfield, S., Benoit, P. and Woolley, J.D. (1997) "KAoS: Toward an Industrial-Strength Generic Agent Architecture," Software Agents, AAAI Press/The MIT Press, Cambridge, Mass., pp. 375-418.

Burstein, M., Ferguson, G. and Allen, J. (2000) "Integrating Agent-Based Mixed-Initiative Control with an Existing Multi-Agent Planning System", Proceedings of the Fourth International Conference on MultiAgent Systems, Boston, MA, 2000.

Clement, B.J. and Durfee, E.H. (1999) "Top-Down Search for Coordinating the Hierarchical Plans of Multiple Agents", Proceedings of the Third International Conference on Autonomous Agents, pages 252-259, May 1999.

Emerson, T. and Burstein, M. (1999) "Development of a Constraint-based Airlift Scheduler by Program Synthesis from Formal Specifications", Proceedings of the 1999 Conference on Automated Software Engineering, Orlando, FL, September, 1999.

Fraser, J. and Tate, A. (1995) "The Enterprise Tool Set — An Open Enterprise Architecture", Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995.

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

Knoblock, C. A., and Minton, S. (1998) "The Ariadne Approach to Web-based Information Integration", IEEE Intelligent Systems , 13(5), September/October 1998.

Rathmell, R.A. (1999) “A Coalition Force Scenario ‘Binni — Gateway to the Golden Bowl of Africa’”, Proceedings of the International Workshop on Knowledge-Based Planning for Coalition Forces, (ed. Tate, A.) pp. 115-125, Edinburgh, Scotland, 10th-11th May 1999.

Tate, A. (1996) “The <I-N-OVA> Constraint Model of Plans”, Proceedings of the Third International Conference on Artificial Intelligence Planning Systems, (ed. Drabble, B.), pp. 221-228, Edinburgh, UK, May 1996, AAAI Press.

Tate, A., Dalton, J. and Levine, J. (1998) “Generation of Multiple Qualitatively Different Plan Options”, Fourth International Conference on AI Planning Systems (AIPS-98), Pittsburgh, PA, USA, June 1998.

Tate, A., Dalton, J., and Stader, J. (2002) “I-P^2 — Intelligent Process Panels to Support Coalition Operations”, in Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations (KSCO-2002) (ed. Tate, A.), 23/24-Sep-2002, Toulouse, France.

Supporting Coalition Operations of Target Movements Exploration through evolutionary Computation and 3-d Visualisation

Jerzy Bala and Dr. Peter Pachowicz
Sigma Systems Research Inc.

Robert A. Witham
US Air Force

Increased responsibilities of today's military coalitions carry a greater need for imagery support. However, as the number of collected images continues to grow, their exploitation needs outpace our resources to analyze them and becomes a bottleneck for the intelligence community. This situation is especially apparent for Ground Moving Target Indicator (GMTI) based imagery data collections. GMTI supports such exploitation processes as target tracking and estimation of target location, identity and activity. The exploitation of information in GMTI could be further enhanced by incorporation of target movement prediction methods. Such methods can potentially provide important information on the enemy's intent that is not currently adequately exploited. However, from a computational point of view, the problem of predicting target movements is very complex. This is attributed to the fact that such prediction would require modeling of various cognitive processes (e.g., group behaviors) that are generally difficult to define or formulate abstractly.

Current tools provide some level of analysis (e.g., flow, sources and sinks, event formation). They apply advanced algorithms for pattern analysis (motion, behavioral), geo-registration, multi-sensor feature correlation (multiple platform tracking), and resource allocation and scheduling. However, as they are mainly processing GMTI tracked data, they lack the capability for prediction of movements, i.e., generating untracked future target flow traffic.

The paper describes a Genetic Evolution of Movement (GEM) approach for inferring opponents' strategic movements and for displaying such predicted movements in an interactive 3D Visualization Space. The prediction approach generates new movements based on past behaviors and application of inheritance mechanisms. Specifically, the approach applies Genetic Algorithms (GAs) learning techniques to evolve new individuals in the population of movements in order to converge the evolution process toward optimal (most probable) movements.

Autonomy of Decision-Makers in Coalitions

K. Suzanne Barber and Cheryl Martin

The Laboratory for Intelligent Processes and Systems
The University of Texas at Austin, Department of Electrical and Computer Engineering
201 East 24th Street, ACE 5.402, Austin, TX 78712
{barber,cemartin}@mail.utexas.edu

1 Defining Autonomy

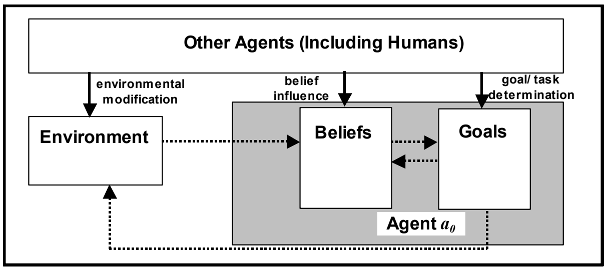
To dynamically form coalitions of decision-makers, the degree of autonomy assumed by each decision-maker must be explicitly agreed upon, beneficial for coalition members and result in productive development of solutions for the goals the coalition is pursuing. Autonomy is a very complex concept. This discussion develops a definition for one dimension of autonomy: decision-making control. The discussion highlights the notion of decision-making control (autonomy) in the context of decision-making groups or coalitions. The development of this definition draws salient features from previous work. Each stage in the development of this definition is highlighted by bold text.

The general concept of agent autonomy is often interpreted as **freedom from human intervention, oversight, or control** (Beale & Wood, 1994; Etzioni and Weld, 1995; Evans et. al., 1992; Jennings et. al., 1998; Wooldridge and Jennings, 1995). This type of definition corresponds well to the concept of autonomy in domains that involve single-agent-to-human-user interaction. However, in multi-agent systems involving numerous coalitions formed to solve specific goals, a human user may be far removed from the operations of any particular agent. Some researchers have defined autonomy in a more general sense as a property of self-motivation and self-control for the agent (Castelfranchi, 1995; Covrigaru and Lindsay, 1995; Jennings et. al., 1998; Luck and D'Inverno, 1995). This sense of the word autonomy captures the concept of **freedom from intervention, oversight, or control by *any other agent***, including, but not limited to, a human.

Unfortunately, this broad statement fails to account for many characteristics often considered necessary for the realization of autonomous agents. For example, the behavior of autonomous agents is generally viewed as *goal-directed* (Castelfranchi, 1995; Covrigaru and Lindsay, 1995; Etzioni and Weld, 1995; Luck and D'Inverno, 1995). That is, autonomous agents act with the purpose of achieving their goals. In addition, many researchers consider *pro-activeness* to be a defining property of autonomous agents (Beale & Wood, 1994; Etzioni and Weld, 1995; Jennings et. al., 1998). Autonomous agents must consider their goals, make decisions about how to achieve those goals, and act on these decisions. Incorporating these properties, autonomy becomes **an agent's active use of its capabilities to pursue its goals without intervention, oversight, or control by any other agent.**

No agent can be completely free from all types of intervention with respect to any goal. This discussion distinguishes among three types of intervention as illustrated in the figure and described below:

1. modification of an agent's environment – other agents modify the environment in which agent a_0 operates,
2. influence over an agent's beliefs – other agents assert facts or, in general, provide information to agent a_0 in order to change or influence beliefs held by agent a_0, and
3. control over the decision-making process determining which goals, sub-goals, or intentions the agent will pursue – other agents participate to a greater or lesser degree in telling agent a_0 how to pursue its higher-level goals.



Extending and modifying the argument presented in (Castelfranchi, 1995), the figure on the right depicts these three ways that other agents (automated or human) may intervene in the operation of agent a_0. The solid arrows in the figure represent interventions that primarily affect an agent's environment, belief base, or goals, respectively. The dotted arrows represent effects of secondary interactions. This discussion suggests that agent designers attempt to classify each agent interaction as one of the three types of intervention based on its primary effect, as pictured in the figure. For example, a task assignment message from agent a_x to agent a_0 should be classified as an intervention of type "goal/task determination" because its most salient effect is to change agent a_0's goals. Certainly, such a message would also affect agent a_0's beliefs (agent a_0 first believes agent a_x wants agent a_0 to perform the new task) and environment (the sending, propagation, and reception of the message imply environmental change). However, these other effects do not capture the nature of the interaction as completely.

Due to the interplay among an agent's goals, its beliefs, and its environment (pictured in the figure by dotted arrows), it can be difficult to ascribe causality for any particular internal agent modification to a specific intervention occurrence. Establishing this causality becomes especially difficult if the internal agent implementation is unknown. This discussion argues that task assignments creating internal goal changes are useful to model for the purposes of describing autonomy. In any system where agent a_x has authority over agent a_y (e.g. leader of a coalition, military command structure, employer/employee, etc.), agent a_x need not convince agent a_y that some goal needs to be done. Agent a_x simply assigns the goal to a_y. Much future work is required to develop classification algorithms for agent interactions, which may ultimately depend on knowledge of the internal design of the particular agents under study. Nevertheless, these suggested categories are useful at this stage to frame discussions of agent autonomy. Because autonomy relates directly to intervention, it is important to be able to identify the nature and impact of these interventions.

This discussion suggests that freedom from intervention of the type "goal/task determination" is the primary dimension of agent autonomy (Barber & Martin, 2000). Goal/task determination is modeled as the process of deciding and assigning which subgoals or subtasks an agent should perform in order to carry out its higher-level goal or inherent purpose. Since any actionable "oversight" or "control" would require such intervention, those terms can be removed from the proposed definition. Therefore, the primary dimension of **autonomy is an agent's active use of its capabilities to pursue its goals, without intervention by any other agent in the decision-making processes used to determine how those goals should be pursued**. This statement presents autonomy as an absolute value (i.e. either an agent is autonomous or it is not). However, it is more useful to model agents as able to possess different degrees of autonomy, allowing the representation of stronger or weaker intervention.

In addition, it is important to recognize that agents often have multiple goals, some of which may be implicit. This discussion considers an agent's degree of autonomy on a goal-by-goal basis, rather than attempt to discuss an agent's overall autonomy as an indivisible top-level concept. This view recognizes that an agent's autonomy may be different for each goal. For example, some would argue that a thermostat is autonomous and others would argue that it is not. This argument actually hinges on which goal is most important in the assessment of the thermostat's overall autonomy. It should be quite easy to agree that the thermostat does autonomously carry out the goal to maintain a particular temperature range but that it does not autonomously determine its own set point. Once an agent's level of autonomy has been specified for each of its goals, the argument can focus (properly) on determining how important each goal is in the assessment of the agent's overall autonomy. The final proposed definition of autonomy follows: **An agent's degree of autonomy, with respect to some goal that it actively uses its capabilities to pursue, is the degree to which the decision-making process, used to determine how that goal should be pursued, is free from intervention by any other agent.**

Agents in a multi-agent system must coordinate to achieve their goals, in general. Establishing an organizational structure (coalition) that specifies how agents in the system should work together helps multi-agent systems achieve effective coordination. Among other things, an organizational structure specifies agent decision-making frameworks. A decision-making framework identifies the locus of decision-making control for a given goal and the authority of decision-makers to assign subtasks in order to achieve that goal. Agents may participate in different decision-making frameworks for each goal they pursue. Agents who implement the capability of Adaptive Decision-Making Frameworks (ADMF) are able to dynamically modify their decision-making frameworks at run-time to best meet the needs of their current situation. Through ADMF, agents are able to reorganize decision-making coalitions by dynamically changing (1) who makes the decisions for a particular goal and (2) who must carry out these decisions. Discussions regarding computational representations of Decision-Making Frameworks (DMFs) can be found in (Barber et. al., 2000) and experiments demonstrating the utility of ADMF are documented in (Barber & Martin, 2001).

References

Barber, K. S., Goel, A., and Martin, C. E. (2000) "Dynamic Adaptive Autonomy in Multi-Agent Systems", "Special Issue on Autonomy Control Software," The Journal of Experimental and Theoretical Artificial Intelligence, vol. 12, no. 2, pp. 129-147, 2000.

Barber, K. S. and Martin, C. E. (2000) "Dynamic Adaptive Autonomy in Multi-Agent Systems: Representation and Justification", in "Special Issue on Intelligent Agent Technology", International Journal of Pattern Recognition and Artificial Intelligence, 2000.

Barber, K. S. and Martin, C. E. (2001) "Dynamic Reorganization of Decision-Making Groups", Autonomous Agents 2001, pp. 513-520, Montreal, Canada, May 28-June 1, 2001.

Beale, R. and Wood, A. (1994) "Agent-based Interaction", People and Computers IX: Proceedings of HCI'94, pp. 239-245, Glasgow, UK, 1994.

Castelfranchi, C. (1995) “Guarantees for Autonomy in Cognitive Agent Architecture” In Intelligent Agents: ECAI-94 Workshop on Agents Theories, Architectures, and Languages, Wooldridge, M. J. and Jennings, N. R., (eds.)., 1995, Springer-Verlag, Berlin.

Covrigaru, A. A. and Lindsay, R. K. (1991) “Deterministic Autonomous Systems”, AI Magazine, vol. 12, no. 3, pp. 110-117, 1991.

Etzioni, O. and Weld, D. S. (1995) “Intelligent Agents on the Internet: Fact, Fiction, and Forecast”, IEEE Expert, vol. 10, no. 4, pp 44-49, 1995.

Evans, M., Anderson, J., and Crysdale, G. (1992) “Achieving Flexible Autonomy in Multiagent Systems Using Constraints” Applied Artificial Intelligence, vol. 6, pp. 103-126, 1992.

Jennings, N. R., Sycara, K., and Wooldridge, M. (1998) “A Roadmap of Agent Research and Development”, Autonomous Agents and Multi-agent Systems, vol. 1, no. 1, pp. 7-38, 1998.

Luck, M. and D'Inverno, M. P. (1995) “A Formal Framework for Agency and Autonomy”, First International Conference on Multi-Agents Systems, pp. 254-260, San Francisco, CA.

Wooldridge, M. J. and Jennings, N. R. (1995) “Intelligent Agents: Theory and Practice”, Knowledge Engineering Review, vol. 10, no. 2, pp. 115-152, 1995.

Knowledge-based coalition planning and operations for Medical Applications

Savitri Bevinakoppa, Dr Dinesh Kant Kumar and Asst. Prof. Jim MacGovern
RMIT University, Melbourne

Dr Kailash Narayan and Rod Hicks
Peter MacCallum Cancer Institute

This paper will investigate the characteristics of women who are diagnosed with cervical cancer, features of cancer tissue on X-ray/MRI/PET images and correlation of the research findings with oncologist data. There is a need to design of image processing techniques for the diagnosis and monitoring of cervical cancer at Peter MacCallum Cancer Institute (Melbourne). These techniques will require an advanced software platform to store and retrieve cancer patient images in database. Advanced algorithms for analysing stored images are required to help the detection of the degree of the spread of cancer with patients. This project aims at designing and testing such techniques. The outcomes of this project will be a software and display of graphical view on computer screen which will be used doctors of Peter MacCallum Cancer Institute to improve the current detection cancer detection techniques.

This work will be carried out among expertise of database management, neural network and medical image datas. All authors will be from these expertise and will work on planning to build suitable patients database.

Towards an Ontology for Intelligence Analysis and Collection Management

Roberto Desimone and David Charles

QinetiQ Ltd
Knowledge & Information's Systems Division
St Andrews Road, Malvern, Worcs WR14 3PS, UK
{rdesimone, dcharles}@QinetiQ.com

Abstract. This short paper discusses research within the "Intelligence Support to Commanders" project as part of the UK MoD Applied Research Programme. It presents preliminary results in exploring medium/long-term concepts for the application of knowledge systems technology for intelligence support activities. An initial ontology is briefly described for intelligence analysis and collection management. The research is predominantly aimed at joint operations, but also addresses coalition issues.

1 Introduction

With the ever-increasing availability of sensor data and other intelligence, it is essential that coherent intelligence support is provided to commanders from strategic and operational commands, down to the lower echelons in the tactical component commands. The intelligence analysts that provide this support, whether in the J2 cells or in tactical intelligence cells, need tools that facilitate collaboration with the whole defence intelligence community, including the intelligence collection agencies and coalition partners.

Collection co-ordination and intelligence requirements management (CCIRM) and intelligence analysis (including fusion) are two key activities currently undertaken by intelligence staff at strategic, operational and tactical levels. Greater decision support is needed for these activities beyond limited office automation tools. Effective collection management requires knowledge of the available intelligence products and their currency, determining gaps and planning for new intelligence to be collected to fill these gaps. The results of intelligence analysis helps commanders make command decisions based on reasoned interpretation of the enemy situation, backed up by solid evidence from intelligence sources. Incorporating intelligence from coalition partners and the sharing of intelligence with them in a reliable and secure manner is becoming increasingly important, but is complicated by differences in doctrine that could result in ambiguity, security constraints that prevent connections between information systems, and other cultural differences.

The "Intelligence Support to Commanders" project started in April 2001 as part of the UK Ministry of Defence (MoD) Applied Research Programme (ARP). The research will take place over the next few years with the following objectives:

- ❑ Confirming user needs for intelligence support to commanders
- ❑ Performing experiments to validate these user requirements by prototype and storyboard development
- ❑ Providing validated technical advice to inform UK MoD procurement decisions.

This paper discusses preliminary results from concept development work within this project.

2 Complex user needs for medium and long-term

The project has been conducting a comprehensive review of current processes for intelligence support and eliciting user needs for improving support in the short and medium-term over the next 2-5 years. This paper addresses user needs in the medium to long-term over the next 5-10 years and possibly beyond. It explores user needs that are complex, involving more dynamic processes than currently in force, and a level of collaboration potentially beyond current doctrine and security constraints. Thus, non-technical, as well as technical, barriers have to be explored to convert these complex user needs into validated user requirements.

Figure 1 depicts an intelligence support environment where intelligence analysts and CCIRM officers can access a multitude of intelligence products and tools that assist them in presenting the right information at exactly the right time and in the right format to support commanders' decision-making. Security permitting, analysts would be able to incorporate the rationale for their recommendations within evidential analyses that would dynamically change in response to new intelligence. Explicit representation of this rationale would help minimise misunderstandings with joint and coalition partners. CCIRM officers would be able to prioritise their requests for intelligence more effectively and work closely with the collection agencies to manage expectations for receipt of specific intelligence material.

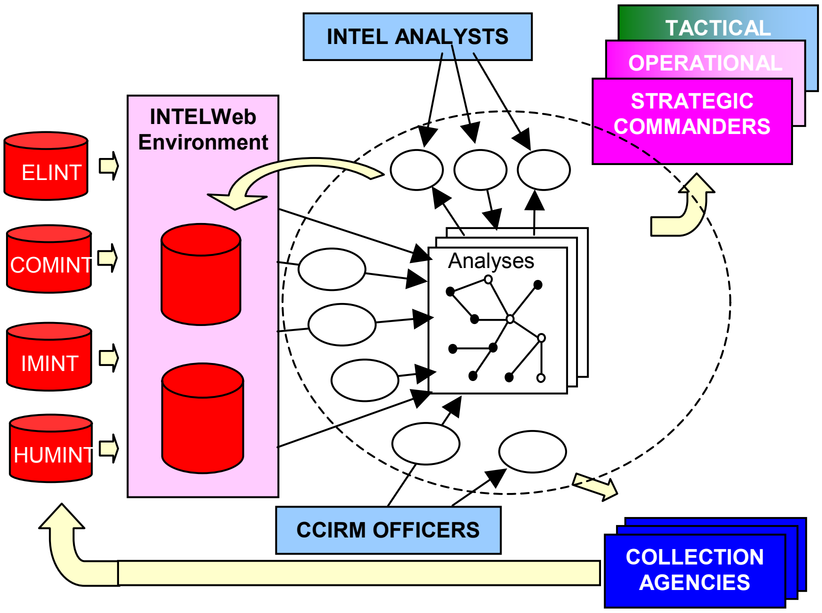


Figure 1. Intelligence support environment

The key to the delivery of these complex user needs is *explicit representation* not only of the intelligence information itself, but also of the processes by which the intelligence has been produced. In effect, an ontology is required for intelligence analysis and collection management. Such an ontology would help provide the basis for semantic interoperability between the plethora of intelligence systems and databases, and encourage an environment where critical information could be shared appropriately with joint and coalition partners. An initial ontology is described later.

2.1 Intelligence analysis concepts and user needs

Commanders normally receive intelligence information in the form of briefings and summaries (INTSUMs), reports (INTREPs) and other intelligence estimates. Battlefield commanders receive more specific documents, entitled intelligence preparation of the battlefield (IPB). These textual reports and oral briefings present critical information, often with recommendations for their most favoured enemy intention. Assumptions for these interpretations are generally recorded, but not in a strong *evidential* sense, pointing exactly to the specific intelligence information that justifies these interpretations. As a result, it is not always easy for the commander to determine whether a particular interpretation has been compromised by new intelligence information, without constant interaction with the intelligence analysts. Conversely, security constraints may prevent the analyst from explaining exactly why a particular command decision might compromise existing intelligence gathering operations. As a result, most of the detailed intelligence analyses, including alternative hypotheses and interpretations, remain in the heads of intelligence officers who rely on individual communication skills to present their brief and keep the commander informed when the situation changes.

The rapidly changing environment and the need for intelligence to flow to exactly where it is needed, both in higher and lower level echelons of command, from where the intelligence analysis has been conducted, means that reliance of face-to-face or voice-to-voice communication is not always going to be achievable. Emerging technologies promise support for the following activities:

- ❑ Assisting the analyst in structuring evidence for their interpretations within evidential graphs, accessing generic and past analytical patterns that recur in similar situations.

  Benefits: *Evidential graphs could provide explicit audit trails for linking textual intelligence summaries and reports to validated intelligence, and facilitate sharing of rationale with joint and coalition partners.*

- ❑ Recording alternative hypotheses and interpretations, together with subjective (pragmatic) and/or objective (quantifiable) metrics for justifying them and for performing sensitivity analyses on them.

  Benefits: *Permits sharing of alternative hypotheses with commanders including their relative weightings, helping them to determine the level of risk associated with their command decisions.*

- ❑ Maintaining dynamic linkages between critical intelligence and interpretations derived from them, and propagating consequences of situation changes, often highlighting alternative hypotheses.

  Benefits: *Commanders can be alerted to consequences of situation changes and alternative hypotheses.*

2.2 Intelligence collection management concepts and user needs

Intelligence officers are being faced with the dilemma of information overload in many areas and yet critical information gaps still occur. Inevitably some of these gaps could be filled by relevant information residing somewhere in the vast repositories, including the heads of intelligence officers and their notebooks. But even when the information is identified, it may not exactly fit the commanders' needs. The information may only partially fulfil the gap, or be too old, inaccurate or unreliable. Thus, the gap still needs to be filled.

Having identified a new intelligence requirement (IR), it is often essential to decompose the request until a number of more specific requests that are pertinent to different collection assets. From these it is possible to determine which collectors should be asked to deliver the necessary information. CCIRM officers then generate a collection requirement (CR), which is disseminated to relevant collection agencies. Negotiation is nearly always required to manage the trade-off between competing IRs/CRs for limited collection assets. Security limitations currently make it difficult for the collection agencies to share their collection plans, even with CCIRM officers. This makes it difficult for CCIRM officers to respond rapidly to dynamic requests from commanders. Ideally, there should be a shared understanding of the IRs, CRs and collection plans between CCIRM officers and the collection agencies. This is going to be difficult enough at national, let alone coalition level; security constraints being the most limiting factor, followed by doctrine and other cultural differences.

Emerging technologies promise support for the following activities:

- ❑ Confirming new intelligence/collection requirements (IR/CR) and satisfying others from existing products.

  Benefits: *Maximise the benefits of existing intelligence collected and minimise over-utilisation of limited collection assets.*

- ❑ Decomposing complex information requests into more detailed specific requests to avoid duplication with other complex requests.

  Benefits: *Partial responses to requests may be provided more rapidly and several requests partially satisfied by the same information.*

- ❑ Managing the trade-off between limited intelligence collection assets, informing scheduling and load balancing tasks by highlighting critical constraints.

  Benefits: *Limited collection assets would be used more effectively to address the highest priority information requests.*

- ❑ Assisting incremental IR/CR development by modifying and re-prioritising activities within existing collection missions in-flight to incorporate new objectives.

  Benefits: *Reduces the intelligence collection cycle significantly.*

3 Initial ontology for intelligence analysis and collection management

Underlying all the complex user needs described earlier is the need for information to be shared between a myriad of different systems. Hence, the project team has been exploring the benefits of developing an ontology that provides an explicit representation for intelligence analysis and collection management applications. Such an ontology would provide a means for bridging the information divide between several intelligence systems and databases and moving towards semantic interoperability at a higher abstract level of understanding.

The ontology should comprise taxonomies of terms for describing objects and activities that are being monitored and analysed, in other words, descriptors for the enemy threat, environment and other situation data. It should also comprise process models for the analysis and collection management processes, clearly identifying the roles and ownership of particular activities. Each activity should include the following:

- ❑ Description – within the context of an accepted verb classification
- ❑ Resources – needed to perform the activity
- ❑ Constraints – quantitative, qualitative and temporal
- ❑ Products – effects of the activities
- ❑ Duration – time to complete the activity.

The products of the intelligence support processes, such as the intelligence estimates, reports, briefings; collection plans, information and collection requirements are all part of the ontology. In addition, the evidential graphs, the structure of the intelligence databases and the systems from which information should be accessed, also comprise the ontology. Effectively, the ontology provides a theory of the domain, with terms for describing products within the domain, activities, players, organisation and authority (policy).

For example, an evidential graph might point to evidence of the presence of enemy that could offer them control of movement within an area of interest (AOI) if they held key terrain. The latter needs to be confirmed. In addition, the commander requires information about enemy strength, composition and disposition, and also which routes should be cut to prevent the key terrain being occupied. Other factors could enhance or prejudice these interpretations, but could also compromise the intelligence collection operations. Expectation of bad weather (low cloud and fog) might require all-weather sensors to be tasked in addition to other, more prevalent, sensors. Knowledge of enemy unit composition would help determine whether signals intelligence (SIGINT) could confirm their location.

3.1 Literature review

During the past few years, there has been a flurry of academic papers reporting attempts at applying ontologies, especially for search and retrieval of information repositories (Uschold & Gruninger, 1996; McGuiness, 1998; Guarino *et al*, 1999; Jasper & Uschold, 1999). Although the term ontology is still relatively new, ontologies have been used effectively under different names in many domains. Astronomers, archaeologists, palaeontologists, and biologists have been refining taxonomies to share research results within their research communities for decades. The international standards community for process applications has been active in disseminating a variety of process formats, IDEF0 being an example.

Even within the military community, standards have been established at the national and international level (NATO STANAGs) for many types of military information formats: NATO AdatP3, the UK Defence Command Army Data Model (DCADM), to name just two. Often these standards are at the detailed data level rather than at more abstract information and knowledge-level, which explains why there are so many of them, and yet interoperability is still a major problem.

Review of the ontology literature suggests that agreeing common standards at higher levels of abstraction is much easier to achieve than at the data-level. There is less need to enforce common data formats that must be adopted by all players, as long as information can be mapped between them at higher abstraction levels. There is still a requirement for common languages to be agreed at some abstraction level. But this can be carefully selected to minimise cost for legacy systems compliance, since data in legacy systems need not be modified. Instead, effort is placed on providing mappings of terms to the common languages.

Although no papers were found on intelligence analysis and collection management ontologies, there is related work on smart workflow technology for intelligence collection management (Berry, 2001a) and on document collection templates for web management systems (Ko *et al*, 2000). Other papers have described research into various prototype intelligence support tools (Gorrell, 1991; Tomlin, 1995; Gonsalves & Rinkus 1998; Jones *et al*, 1998), and lessons learned from collection management operations during Operation Desert Storm (Franz, 1995). These papers provide starting points for an initial process language described next.

3.2 First steps towards a process language

Figure 2 expands on the previous figure, highlighting support processes for intelligence analysis and collection management. A detailed study of the these processes has been conducted within the project, relating them to Joint Essential Tasks (JETs) from the UK Permanent Joint Headquarters, and presented within a storyboard (Storyboard, 2001). The processes are hierarchical with activities being undertaken at different echelons for strategic, operational and tactical purposes. These processes help to determine terms for describing activities, players, products and information flows for each activity.

Table 1 provides a verb classification of key activities, derived from a verb classification for intelligence collection management (Berry, 2001b), which has been refined to include terms for intelligence analysis tasks. Associated with each verb are other verbs that describe related activities, very much like a thesaurus. Such a classification provides a foundation for a hierarchical set of terms for describing how these activities fit together. The next steps involve defining a corresponding noun classification that identifies key players, products and information flows. This is in progress and will be reported in future papers.

We believe that these verb and noun classifications will provide a basis for building a process language. Together with a corresponding taxonomy of terms (nouns and verbs) for describing situation information (e.g. enemy threat, environment, and other situation data) they will form major parts of the overall ontology. Other elements of the ontology would include representation of information flow and delegation of authority. The workflow community has been developing tools that are relevant, and have been explored recently for collection management (Berry, 2001a). Commercial workflow tools are still limited, since they tend support well-defined processes, rather than dynamic ones, but do provide a starting point for exploring transferring delegation of authority.

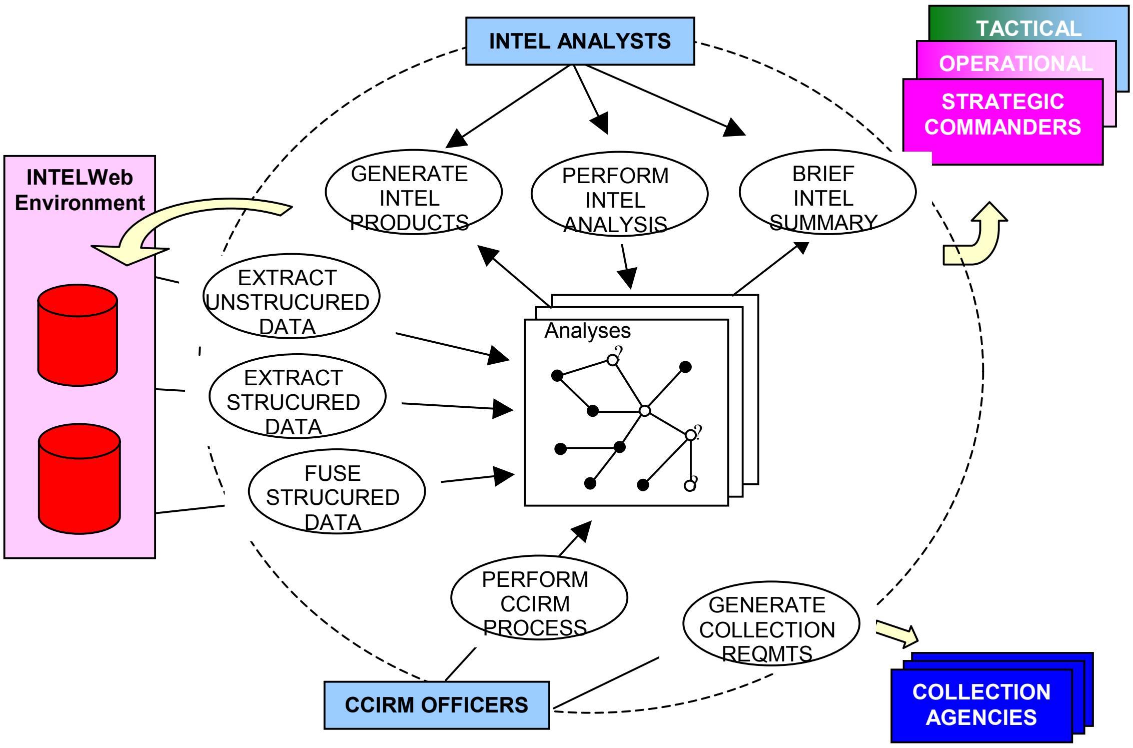


Figure 2: Intelligence support processes

ANALYSE – predict, determine, monitor, diagnose, measure
ASSESS – estimate, expect, consider, ascertain, determine, evaluate
ASSIGN – apportion, delegate
COMMUNICATE – request, acknowledge, reject
DECIDE – complete, finalise, approve, terminate, choose
DEVELOP – build, construct, create, compose, generate, prepare
EXTRACT – retrieve, search, mine
FUSE – collate, correlate, aggregate and reduce
IDENTIFY – classify, group, match, select, compare, resemble, detect
ISSUE – circulate, transmit, publish, deliver, release
MODIFY – combine, join, link, refine, integrate, evolve, augment
OBTAIN – receive, acquire, establish
ORGANISE – co-ordinate, regularise, formalise, de-conflict, phase, sequence, plan
PERFORM – execute, undertake
PRIORITISE – order, rank
PROVIDE – supply, furnish, equip, offer, give, input
REVIEW – learn, appraise, summarise, critique
SUPPORT – sustain, aid, assist, approve

Table 1: Preliminary verb classification for intelligence analysis and collection management

3.3 Next steps

The next steps involve extending the verb classifications, integrating them with relevant noun classifications and building up the process language for intelligence analysis and collection management. The emerging process language will be applied to prototypical, but, initially, small analysis and collection tasks that match the user needs identified earlier, and tested for expressiveness and effectiveness. In addition to a process language, the ontology requires a domain language for describing terms within the intelligence reports and estimates. Eventually, experimental plans will be defined that validate the complex user needs outlined earlier, so that the relevant military requirements can be informed.

4 Summary and conclusion

This short paper describes concept development work within the "Intelligence Support to Commanders" project. Complex user needs are outlined in support of intelligence analysis and collection management tasks. A review of ontology research is briefly described, and an initial ontology for intelligence support tasks is proposed. The first steps towards a process language for describing intelligence analysis and collection management tasks is presented, together with next steps. Eventually, this research will lead to experimental plans that aim to validate the complex user needs, so that relevant military user requirements can be informed.

Acknowledgements

This research has been funded partly under the UK MoD Applied Research Programme (ARP), package 13 for the Command Control and Information Infrastructure (CCII) capability area.

References

Berry, P. (2001a) http://www.ai.sri.com/~swim

Berry, P. (2001b) http:/www.ai.sri.com/~swim/publications/capabilities-template.html

Frantz, G. (1995) "Beyond Desert Storm: Conducting intelligence collection management operations in the Heavy division", Monograph, Army Command and General Staff College, Fort Leavenworth, USA, December 1995.

Gonsalves, P.G. and Rinkus, G.J. (1998) Proceedings of the 1998IEEEInformation Technology Conference, Information Environment for the Future, New York, NY, USA, September 1998.

Gorrel, B.J. (1991) "An experts system for collection management operations (ESCiMO) technical Report, Royal Military College of Science, Shrivenham, UK, 1991. UK RESTRICTED.

Guarino, N., Masolo, C. and Vetere, G. (1999) "Ontoseek: using large linguistic ontologies for accessing on-line yellow pages and product catalogues", IEEE Intelligent Systems, Vol 14 (3), pp 70-80.

Jasper, R. and Uschold, M. (1999) "A framework for understanding and classifying ontology applications", Proceedings of the Ontology Workshop, International Joint Conference on Artificial Intelligence (IJCAI-99), August 1999.

Jones, P., Hayes, C., Wilkins, D.C., Bargar, R., Sniezek, J., Asaro, P., Mengshoel, O., Lucenti, M., Choi, I., Tu, N., and Schlabach, M. (1998) "CoRAVEN: Modelling and design of a multimedia intelligent infrastructure for collaborative intelligence analysis" Proceeding of IEEE Systems, man and Cybernetics Conference, 1998.

Ko, I-Y., Neches, R. and Yao, K-T. (2000) "Semantically-based active document collection templates for web management systems", Proceedings of the International Workshop on the Semantic Web, Lisbon, Portugal, September 2000.

McGuiness, D. (1998) "Ontological issues for knowledge-enhanced search" In Guarino, N., (ed) Formal Ontology in Information Systems, pp 302-316, Trento, Italy.

Storyboard (2001) "Intelligence Support to Commanders Requirements Capture Storyboard", UK RESTRICTED.

Tomlin, K. (1995) “The design and implementation of an automated intelligence collection management tool”, Masters Thesis, Naval Postgraduate School, Monterey, CA, September 1995.

Uschold, M. and Gruninger, M. (1996) “Ontologies: Principles, methods and applications”, Knowledge Engineering Review, Vol.11(2).

Agent-Based Modelling for Environmental Coalition Formation

Jim Doran

Department of Computer Science
University of Essex
Colchester, UK, CO4 3SL
doraj@essex.ac.uk

Abstract. Planned intervention to achieve stakeholder cooperation and coalition is essential for successful environmental management. Agent-based modelling on a computer has the potential to build a practical theory of intervention in this and related contexts. Potentially we can compare real intervention strategies with those an agent-based model suggests and hence obtain new insights and guidelines of practical value. But the technical problems of model building for this purpose are formidable. We explain and discuss these problems by reference to an example model specification framework, and seek ways forward. Insights obtained may be generalised to coalition formation in general.

1 Introduction

Based on multi-agent systems (MAS) theory (Weiss, 1999), computer and agent-based modelling of social and organisational systems (Doran, 1997, 2001a) is becoming of practical value in a range of application domains (Moss and Davidsson, 2001) including the military (Tessier et al., 2000), the environmental (Bousquet et al., 1999) and the social (Gilbert, 2000).

Here we take the view that a multi-agent system is an interacting collection of agents sharing a common (possibly simulated) environment, where an agent may loosely be viewed as an “object” in the software engineering sense that possesses a degree of autonomy and a modicum of cognitive ability. This indicates the relevance of artificial intelligence theory and practice (Russell and Norvig, 1995).

Cooperation is a key topic in MAS (Doran et al., 1997). Most agent work on cooperation concerns how to design cooperation into a MAS, or how to model existing cooperation, rather than how to achieve it in a pre-existing non-cooperating set of agents. But *achieving cooperation* in a pre-existing situation is very often the real-world problem. We view real-world *coalitions* as involving the mutually agreed temporary cooperation of large organisations without loss of organisational identity or rights. Often the word “coalition” has international military or national political connotations.[1] However, Keohane and Ostrom (1995) have demonstrated the close relationship between cooperation for the solution of environmental problems, and more general international cooperation.

2 An Environmental Problem: Integrated Watershed Management

Integrated watershed management is the task of organising the activities and requirements in a river basin to achieve multiple and conflicting goals (Abu-Zeid and Biswas, 1996; Westervelt, 2000). Stakeholder cooperation is essential. Typically there are conflicting requirements to be balanced of:

- water supply (domestic, agricultural, industrial uses)
- pollution control
- fisheries management
- flood control
- hydropower production
- navigation and wetlands management
- recreation provision

Always there will be many stakeholders associated with different activities in the basin, all with their own objectives and agendas. Conflicts of interest are inevitable. A good example of this is the Fraser River basin in British

[1] Compare the connotations of the word “consortium”.

Columbia (Healy, 1999; Doran, 2001). Large-scale *river engineering* projects can involve even wider issues, but are beyond the scope of this short paper.[2]

3 Interventions and Models of Intervention

Integrated watershed management, and similar ecosystem management problems, typically involve *intervention*. That is, some person, some group or some organisation, has the task of intervening in the ecosystem in order to bring about desirable change, often using the notion of a search for *sustainability*. The intervener may be, for example, a branch of the UN, an NGO, an academic research team or even a lone doctoral student. The practice of intervention is so much a part of the ecosystem management task that, in our view, it is unrealistic to ignore it for modelling purposes. The intervention history of the Fraser River Basic is a revealing example of just what issues can arise in intervention, and what can go wrong (Dorcey, 1997; Marshall, 1998; Doran, 2001).

It is evident that there can be a range of *intervention strategies*. A number of these have been discussed in Doran (2001). Here we are particularly interested in a two-stage intervention process, in which intervention first seeks to build an effective coalition and only then to set that coalition into action on the actual management task.

Symbolically we may write the intervention task as:

$$\mathit{INTERVENTION} \rightarrow (\mathit{MAS} + \mathit{ENVSYS})$$

or recognising, that coalition formation may be part of the intervention process, as:

$$\mathit{INTERVENTION} \rightarrow \mathit{COALITION} \rightarrow (\mathit{MAS} + \mathit{ENVSYS})$$

We would like to model *all* of this intervention process on a computer in order to explore possible intervention strategies with the minimum of habitual and cultural pre-conceptions.

3.1 Essentials of a Typical Environmental Resource Management Problem

We assume that environmental “harvesting”[3] requires:

- distributed action coordinated in space and time.

Furthermore actors (individual or organisational) must show restraint if they are to achieve, as we shall require:

- collective long-term survival (i.e. *sustainability*)
- the *protection* of specified environmental components
- some kind of *equity*[4] between actors

The central difficulty is that human beings tend to be individually, collectively and organisationally “greedy” and with bounded rationality. In particular, we tend to think short-term. Any potentially informative model must capture these characteristics. Compare “common pool resource” (CPR) problems of which this formulation may be seen as generalisation (Hardin, 1968; Ostrom, 1990, 1995).

3.2 A Research Plan

For clarity and focus, we foreground the following five-stage computer-based research plan:

1. Formulate a representative ENVSYS in mathematical/computational terms. The ENVSYS must reward distributed coordination and embody the sustainability, equity and protection problems identified above. Examine its long-term dynamics.

[2] The Three Gorges Project on the Yangtze, for example, involves further issues such as massive population movement and destruction of archaeological sites – and for this and other reasons has become highly politically charged.

[3] The work “harvesting” is here used in an extended sense to cover the collective exploitation of natural resources.

[4] Not all would agree that the last of these requirements, the equity requirement, should be included in a general definition of natural resource management.

2. Generate a sample of MAS connected to the ENVSYS. They should be neither incoherent nor successfully achieving sustainability, protection and equity over the chosen time span, that is, the generated MAS should function but fail to solve the problems.

3. Try to interpret the generated sample MAS in first abstract then human/social terms. This will probably include recognition of different types of MAS.

4. Search the space of all possible interventions to find those that are most successful for MAS of each type, where success refers to a high degree of maintenance of harvest, without depletion of protected environmental components, and with equal distribution of harvest over the set of agents.

5. Interpret the interventions found in both abstract and human/social terms

Throughout the execution of such a research plan it is essential not to confuse two distinct domains of investigation:

- Intervention to achieve cooperation in a human social system with initially conflicting stakeholders
- Intervention to achieve cooperation in an abstract MAS on a computer with initially conflicting agents

It is the latter computational domain to which the research plan directly refers. The central questions are whether effective intervention strategies can be identified in the computational domain, and then whether or not these identified intervention strategies have relevance to the real world domain.

4 A Framework for a Model

To proceed we need a precise and programmable specification of a MAS+ENVSYS and of possible interventions upon it that is sufficiently realistic for conclusions drawn from it to be reliable. In spite of all the advances made in agent technology and artificial intelligence over the past half century, this is difficult to achieve. The following framework should therefore be regarded as, at best, pointing the way ahead.

4.1 Two Basic Assumptions

We work from two basic assumptions, both controversial. The first is:

All social phenomena can in principle be captured within a computer-based model

This is analogous to the strong AI assumption that all aspects of intelligence can be captured within a computer-based model. It lies at the heart of multi-agent-based social modelling, but certainly not all social scientists or practitioners of agent-based social modelling would subscribe to it. Its significance here is that it encourages us to be optimistic that the model we want can in principle be found.

The second assumption is:

The social is emergent from the individual and the neural, and should be modelled accordingly

If anything this is even more controversial, for it is strongly reductionist and therefore unfashionable. Its significance here is that it suggests that to design and build an explicitly high-level social model is to omit its most important property, *emergence* (see, for example, Conte and Gilbert, 1995, pp 8-12). Rather the objective must be to explore the space of *low-level* models, seeking those that display high-level emergent phenomena and structures. Thus the specification that follows delineates a *class* of models rather than a specific model. Indeed, the aim is not to design a model ourselves, but rather to discover what models are *possible*, employing in effect a process of intelligently designed and efficient "generate and test".

4.2 ENVSYS

An ENVSYS is structured as a set of Boolean, integer or real-valued variables inter-related by recurrence relations of the general form

$$x_n(t+1) = f(x_1(t) \ldots\ldots x_q(t))$$

where t refers to time and the subscripts index variables.

It is *not* intended that the ENVSYS be a model of a particular real-world environmental system. Rather *the recurrence relations, together with the "actions" available to the agents (see later) and the agents' "localities"(see later), should be chosen to provide the required resource management problem characteristics, that is, the need for distributed and coordinated harvesting together with difficulty in achieving sustainability, protection and equity* (see section 3.1). Distributed and coordinated harvesting may be a matter of a specified pattern of actions upon a particular set of variables (actions and variables distributed in time as well over localities) having a disproportionate and "beneficial" impact upon key harvestable variables. Motivating real-world instances range from large-scale irrigation systems and specialised artefact production to simple group cooperation activities such as ditch digging and tree felling. Problems of sustainability (and protection) may be posed by so choosing the ENVSYS relations that harvesting beyond a certain amount results in the harvestable (or protected) variables being driven beyond acceptable limits or permanently set to zero. Equity is naturally expressed as the requirement that all agents harvest to roughly the same degree.

The ENVSYS may be formulated in many ways. For example, the recurrence relations may form something akin to a classic systems dynamics model (see Westervelt, 2000). Alternatively, the ENVSYS may be more in the tradition of "Artificial Life" studies with a spatial interpretation that has agents moving and harvesting localised resources on a plane (e.g. Epstein and Axtell, 1995).

4.3 MAS Agents

Agents must harvest at a minimum total rate or they are deleted.

Each agent is structured as a set of tokens, the contents of its working memory (WM), together with condition-action rules that execute upon and manipulate the working memory and which observe and manipulate the agent's external context.

Tokens

EITHER a simple token

a (bounded) string of letters, possibly prefixed by not (the negation character)

OR a variable-value token

a pair: a (bounded) string of letters, and a value

Rules

A pair:

a (bounded) set of tokens and a (bounded) set of actions

where an action is of one of the following types:

Harvest -- deplete a specified ENVSYS variable by a specified amount

Set -- set a specified ENVSYS variable to a specified value

Read -- read the value of a specified ENVSYS variable and deposit a corresponding variable value token in the WM

Deposit -- deposit a specified token in (own) WM

Send -- deposit a specified token in WM of another specified agent

Locality

Each agent has its own locality, which is fixed in time, that is, each agent can set, read and harvest a specified subset of the variables – its "local" variables.

4.4 Agent Processing

Rule Firing {

```
          Find all rules whose LHS match in the current WM where a match
          requires that every LHS token occurs in the current WM

          Select a matched rule at random

          Execute the selected rule's action(s)
        }
```

Token Reconciliation

We say that two tokens *contradict* if they differ only in the negation character.

It is assumed that the initial contents of the WM are contradiction free. If a token is introduced (by an internal or external rule firing or an intervention) that contradicts an existing token (i.e. differs from it only in the negation character) then the pre-existing token is deleted from the WM. This conflict removal procedure is very simplistic and certainly not, of course, logically complete.

Rule Set Reconciliation

We say that two rules *contradict* if their conditions are identical but their actions differ.

It is assumed that the initial rules set is contradiction free. If a rule is introduced into the rule set (by intervention) that contradicts an existing rule, then the pre-existing rule is deleted. Again, this conflict removal procedure is not logically complete.

4.5 Intervention

An *intervention element* is the deposition of one token or one rule into a particular agent's working memory at time t. An *intervention* is a set of N intervention elements. The impact of an intervention element is determined by the reconciliation procedure.

4.6 Processing MAS+ENVSYS + interventions

Initialise MAS+ENVSYS at random and set clock to zero

```
Repeat
{
      Advance clock (t)

      Activate each agent once
            (in a varying random order)

      Pass any inter-agent messages

      Apply any interventions at this time

      Reconcile each agent's tokens and rules

      Update the ENVSYS

      Collect statistics
}
until time limit reached
```

This semi-formal specification is, in artificial intelligence terms, very simple. "Filled in" with rules and initial token sets for the agents' working memories, it is clearly programmable (in, for example, C++) and model instances can therefore certainly be "run" and experimented with. However, the combination of tokens and rules is computationally sufficiently powerful that complex cognitive processes such as learning and planning are certainly possible. It is not easy to anticipate in any detail what specific types of dynamics will occur within an agent or a

MAS in particular circumstances.[5] Nevertheless some aspects of the behaviour of this type of model are foreseeable, as we shall discuss in the next section.

5 Major Characteristics of the Model

We now turn to the foreseeable characteristics of this model, and the technical difficulties that any attempt to use it will encounter.

5.1 Properties and Problems

It is important to appreciate that complex cognitive processes, for example the use of internal representations, goal setting, plan formation and execution, and learning, are potentially present in an agent's working memory dynamics even though agents are "merely" rule based. This follows from the fact that the contents of an agent's working memory both determine and are modifiable by the rules that "fire". That does not mean, of course, that agents with cognitive processes are easily generated nor, less obviously, that it is easy to recognise them when they are. Indeed, just how cognitive processes can be recognised in practice in such a context is an interesting and far from trivial question.

The behaviour of any particular instance of the model that meets the specified requirements (a *solution model instance*) is primarily determined by the rule sets within the agents. To serve our purposes, these rule sets must be such that the MAS, without intervention, has the specified properties with respect to the ENVSYS, notably that it does successfully "harvest" resources, but not so that it is immediately sustainable, equitable and protective. But the probability that an arbitrary or randomly generated MAS will function in this way, or even function coherently, is very small indeed. There is therefore a significant combinatorial problem merely to find functioning and effective MAS. Some form of "hill-climbing" algorithm or evolutionary algorithm[6] could be used, at least on the micro-scale. Just how complex are the effective MAS that could be found in this way is an open question. Of course, one could set out explicitly to *design* an effective MAS (a kind of programming exercise) but this would be to pre-determine what we wish to discover, and it encounters head-on the difficulty that our ability to program the needed artificial intelligence capabilities is limited. A compromise might be to design some basic structures and capabilities into the model's agents, perhaps sufficient for their minimal survival by purely uncoordinated action in the ENVSYS, and to leave the rest to some form of heuristic or evolutionary search.

Once discovered, effective MAS may or may not display (emergent) collections of agents that may reasonably be labelled "organisations" (compare Prietula et al., 1998). They may or may not display centralised decision-making and/or collective planning. Agents (and agent organisations) will typically be heterogeneous, perhaps in a patterned way and, as just suggested, may or may not incorporate cognitive processes. All discovered MAS are likely to be "noisy" in the sense that their rules and working memory contents will often include much that is inessential to their required functioning.

Recall that the purpose of generating MAS that can successfully interact with the ENVSYS is precisely to *discover* what form such MAS can take (rather then prejudge that issue) and to then take the next step to consider intervention.

5.2 Interventions and Intervention Strategies

Organised patterns of intervention (*intervention strategies*) may be *discovered* to be structured in various ways, and they may either prompt a successful pattern of action, or may prompt a social structure (e.g. a coalition) which will itself achieve the required pattern of action, or may prompt something even more complex.

Assuming a fixed instance of a MAS+ENVSYS, optimal interventions can be defined and (in principle) determined without addressing the issue of the intervener's knowledge of MAS+ENVSYS. However, this issue cannot be avoided if the requirement is changed to that of finding a *decision procedure* that gives an effective intervention. Such a decision procedure would be a function of the intervener's knowledge of the MAS+ENVSYS.

5.3 Translation to and from the Model

To make practical use of a solution model instance requires that we are clear about the structural relationship between the two domains of intervention strategy. For example, what corresponds in the abstract model to

[5] Compare Turing Machines (Turing, 1937), and also the well-known Agent0 agent-oriented programming language (Shoham, 1993).

[6] We are here using the phrase "evolutionary algorithm" in a technical sense. There is no question of modelling human evolution.

centralisation and decentralisation? social capital? organised conflict? a coalition? And how may these specifically be achieved by intervention? Here we focus briefly on coalitions.

5.3.1 Coalitions

Assuming the model specification of section 4, and given our initial attempted definition of a coalition as "involving the mutually agreed temporary cooperation of large organisations without loss of organisational identity or rights", what form would a coalition take in such a model, under what circumstances might intervention lead to the formation of a coalition, and when might that coalition be effective?

It seems reasonable to suggest that we are looking for a set of agents that are in some sense "leaders of" organisations and that further, for a significant period of time:

- have a pattern of inter-communication amongst them, and
- display some degree of shared goals, and
- display a degree of coordinated action.

It follows that the recognition of coalitions in a MAS rests upon the recognition of lower-level phenomena such as goals, communication and coordinated action. But more is needed: specifically a precise account of just what is involved in the formation, action and dispersal of coalitions. A possible basis is the formal account of the various stages of a group cooperation and action process provided by Wooldridge and Jennings (1999). Although their account is formulated in terms of a quantified multi-modal logic, and at first sight seems too abstract to be helpful here, in fact it does go at least part way to providing the kind of precise recognition procedure required. If a recognition procedure can be established, it then becomes feasible to address the ways in which different interventions strategies impact upon the MAS+ENVSYS combination, and to identify those classes of intervention strategy that lead to effective coalition formation.

6 Discussion

It may be argued that a study of this type can have very little practical value, since (i) only the simplest solution model instances can be found however sophisticated the combinatorial search procedure deployed, and these models will therefore be unrepresentative, and (ii) there are deeper reasons, in any case, why such models can *never* be relevant to real human social situations.

Point (i) seems unduly pessimistic. The success of techniques for finding solutions to complex problems by evolutionary and other heuristic techniques is well known. To assume that they will be useless in this context is surely unjustified. Furthermore, the structure of the problem, involving specific and well-defined requirements that must be met, means that the search for solution models is through a space that is in fact quite tightly defined. Coupled with ever increasing available computer power, it is at least feasible that interesting discoveries may be made.

The second objection (ii) is essentially a "philosophical" one based upon a perception that there is something intrinsically different about human society compared with an artificial agent society. In particular, it is a perception that human and agent societies must differ in how they collectively address resource acquisition and distribution tasks. This perception runs counter to our initial assumption that all social phenomena can be captured within an agent-based model. More importantly, it also runs counter to much current research that assumes and demonstrates that there is indeed a fruitful basis for the exchange of ideas about the two types of society. Is it really the case that, say, groups of robots and groups of humans faced with the same foraging task will never deploy similar strategies?

7 Conclusions

We have suggested how social intervention strategies can be discovered and classified in the abstract by generating and exploring a "space" of relevant agent-based models. The objective is to match discovered abstract strategies to those in actual "everyday" use, and vice-versa, in an insightful and practical way. In principle, this includes intervention strategies that use coalitions as a "stepping stone". But there are major technical problems to be overcome of two kinds: exactly how to generate specific model instances of sufficient complexity to be representative and informative, and how to interpret complex model instances once generated. Thus although there is substantial potential payoff, the prospect is a long-term and challenging one.

References

Abu-Zeid, M.A., and Biswas, A.K., (1996) “River Basin Planning and Management”, Oxford University Press.

Bousquet F., Barreteau O., Le Page C., Mullon C. and Weber J. (1999) “An Environmental Modelling Approach: the Use of Multi-Agent Simulations”, in “Advances in Environmental and Ecological Modelling” (F. Blasco and A. Weill, eds.), Paris, Elsevier, pp. 113-122.

Conte, R. and Gilbert, N. (1995) “Introduction: Computer Simulation for Social Theory”, in “Artificial Societies: the Computer Simulation of Social Life” (N. Gilbert and R. Conte, eds.), UCL Press, London, pp. 1-15.

Doran, J. E. (1997) “From Computer Simulation to Artificial Societies”, Transactions of the Society for Computer Simulation International, 14, 69-78

Doran, J. E. (2001) “Intervening to Achieve Co-operative Ecosystem Management: Towards an Agent Based Model”, Journal of Artificial Societies and Social Simulation, 4(2), 2001, ON LINE: http://www.soc.surrey.ac.uk/JASSS/4/2/4.html

Doran, J. E. (2001a) “Can agent-based modelling REALLY be useful?”, in “Cooperative Agents: Applications in the Social Sciences” (N. J. Saam and B. Schmidt, eds.), Philosophy and Methodology of the Social Sciences Series, Volume 32, Kluwer Academic Publishers, Dordrecht.

Doran, J. E., Franklin, S., Jennings, N. R. and Norman T. J. (1997) "On Cooperation in Multi-Agent Systems", The Knowledge Engineering Review, 12(3), 309-314.

Dorcey, A. H. J. (1997) “Collaborating Towards Sustainability Together: The Fraser Basin Management Board and Program”, in “Practising Sustainable Water Management: Canadian and International Experiences” (D. Shrubsole and B. Mitchell, eds.), Canadian Water Resources Association.
http://www.interchg.ubc.ca/dorcey/chcwra/fccwra.html

Epstein, J. M. and Axtell, R. (1996) “Growing Artificial Societies: Social Science from the Bottom Up”, The Brookings Institution Press, Washington, D.C., The MIT Press Cambridge, Mass.

Gilbert, G. N. (2000) “Modelling Sociality: The View from Europe”, in “Dynamics in Human and Primate Societies” (T. A. Kohler and G. J. Gumerman, eds.), Oxford University Press, pp. 355-372.

Hardin, G. (1968) “The Tragedy of the Commons”, Science, 162, 1243-1248

Healey, M., ed. (1999) “Seeking Sustainability in the Lower Fraser Basin: Issues and Choices”, Institute for Resources and the Environment, Westwater Research Centre, University of British Columbia, Vancouver.

Keohane, R. O. and Ostrom, E. eds. (1995) “Local Commons and Global Interdependence: Heterogeneity and Cooperation in Two Domains”, Sage Publications, London.

Marshall, D. (1998) “Watershed management in British Columbia: The Fraser Basin Experience”, Environments, 25(2/3), 64-79.

Moss, S. and Davidsson, P. eds. (2001) “Multi-Agent-Based Simulation” Proceedings of the Second International Workshop on Multi-Agent Based Simulation (MABS 2000), Boston, MA, USA, 8-9th July, 2000, Springer, Berlin

Ostrom, E. (1990) “Governing the Commons: The Evolution of Institutions for Collective Action”. Cambridge University Press: Cambridge, UK.

Ostrom, E. (1995) “Constituting Social Capital and Collective Action”, in: “Local Commons and Global Interdependence: Heterogeneity and Cooperation in Two Domains” (R. O. Keohane and E. Ostrom, eds.), Sage Publications, London. pp. 125-160.

Prietula, M. J., Carley, K.M., and Gasser, L. (1998) “Simulating Organizations”, AAAI Press/The MIT Press Menlo Park, California, Cambridge, Mass., and London, England.

Russell, S. and Norvig, P. (1995) “Artificial Intelligence: a Modern Approach”, Prentice-Hall.

Shoham, Y. (1993) “Agent-Oriented Programming”, Artificial Intelligence, 60(1), 51-92.

Tessier C., Chaudron L., Müller H-J., eds. (2000) “Conflicting Agents: Conflict Management in Multi-Agent Systems”, Kluwer Academic Publishers, Dordrecht.

Turing, A.M. (1937) “On Computable Numbers with an Application to the Entscheidungs-problem”, Proc. London Math. Soc., 42, 230-265; corrections, ibid, 43 (1937) 544-6.

Weiss, G. ed. (1999) “Multiagent Systems”, The MIT Press, Cambridge, Massachusetts, and London, England.

Westervelt, J., (2000) “Simulation Modeling for Watershed Management”, Springer-Verlag, Telos

Wooldridge, M. and Jennings N.R. (1999) “The Cooperative Problem Solving Process”, Journal of Logic and Computation, 9(4), 563-592

Agents On The Semantic Object Web: Information Management for Coalition Operations

G. Edwards, B. Kettler, K. Olin, and B. Tsurutani

ISX Corporation
4301 N Fairfax Drive, Suite 370
Arlington, Virginia 22203
gedwards@isx.com

Abstract. Coalition operations pose clear challenges for information sharing and for the integration of disparate information processes. ISX Corporation has been investigating the combined use of agent-based approaches to implement Information Management Agents™ operating on a semantically organized Semantic Object Web™ to provide highly flexible means of delivering information services within a large, distributed and diverse enterprise. The techniques described demonstrate their particular suitability to rapidly standing up a "come as you are" organization of coalition partners whose information protocols, requirements, and processes might vary widely. The use of a Semantic Object Web provides an agent-navigable information substrate which can be used by service-provider agents to discovery relevant information or services, to map their own content into a shareable semantic space, and to exploit available content and services. Using agent-based techniques, the approaches described provide the ability to deconstruct information requirements to guide matching of source / consumer relationships within the enterprise, and then to compose, aggregate, and transform information from various sources to meet those requirements. In this vein, the techniques provide services registration, matching and brokerage, agent facilitation and control, and semantic matching and transformation necessary to compose the right information for the coalition force member. In addition, this work also addresses the dissemination and delivery of information. Information Management Agents provide the means to constrain and guide information access and delivery within the coalition means to implement selective information management business rules and policy server-imposed access and publication restriction. Together, these technologies promise the capability to support interoperability across coalition organizations while maintaining necessary policy and process-based constraints on information access and dissemination constraints.

1 Introduction

Information technology to support integrated coalition operations in future military environments poses a two-edged opportunity. Effective coalition operations will rely on information technology as a key enabler. Coalition operations are all about bringing together a diverse set of organizations, each with their own capabilities, processes, and infrastructure, and forming them into a working enterprise tailored to the military operation at hand. Information technology-enabled sharing and coordination are at the heart of successful coalition operations. At the same time, coalitions are not seamless organizations. Effective sharing and coordination, to occur at all, must also mean acceptable sharing and coordination. In the real world, information technology must also enforce constraints on what information can be shared and what services can be provided, under what conditions, and to whom.

In this paper, we will explore selected key facets of the problem of integrating diverse command and control organizations into a unified enterprise in the face of such constraints. Our discussion will focus on the interplay of three key problem elements: interoperability, or how information can be practically exchanged and shared across coalition elements; process, focusing on how information technology can enforce workable C2 enterprise "business relationships" between organizations in the coalition; and policy, or how information technology can enforce organizationally-imposed rules that let coalition partners share and coordinate within constraints they impose on their own information assets. Our technology discussion will focus on primarily on the roles of two key technologies. The first of these technologies, the Semantic Object Web ™, provides an ontologically grounded means of indexing diverse coalition information sources into a single integrated information space. The second technology approach exploits self-organizing collections of software agents, capable of querying and navigating the Semantic Object Web, as tools to deliver both information access and delivery services, and to implement information management constraints, processes and policies. We believe these technologies, in concert, provide an interesting approach to dealing with key problems in the coalition enterprise space. While we are not proposing a comprehensive architecture or solution to the coalition management problem, we believe the ideas expressed here, and the initial experimental work we have undertaken, provide some interesting and potentially powerful approaches.

2 Operational Demands

Information is the commodity that enables coordinated and effective military operations. Effective exploitation of that information is key to successful operations. One of the challenges for a military coalition commander and his staff is to organize a command and control (C2) enterprise built around the requirements and characteristics of a specific military operation. This enterprise must stand up quickly to put working operational capabilities into place. And increasingly, in a world where a broad community of international partners share a global interest in regional stability, this enterprise will include coalition partners. Creating a working command and control enterprise in this context poses a wide range of issues, from simple interoperability issues to complex mechanisms for services discovery and negotiation of roles. Unlike the past century, these coalitions will not be limited to NATO-like organizations with long-standing alliances and deeply rooted operational practice refined and coordinated in numerous exercises and operations. The problem for information technology is to take a complex set of extant command and control capabilities, often with partners whose command and control organizations have never worked with each other before (and were not designed to do so), and to integrate and shape those capabilities into an operationally-tailored working enterprise that enables both interoperability and coordination.

Information technology must facilitate the commander's ability to exercise control over the information flows within this enterprise, and must also meet all of the information management constraints imposed by participating organizations. Modern coalitions will be composed of more loosely associated groups of nations, each with its own level of commitment to the coalition, each with its own agenda (which will certainly only partially overlap interests of its coalition partners), and each engaged in a limited role within the operational enterprise. Certainly, many concerns about information management, dissemination, access, etc. are driven by security concerns. Loose coalition partnerships may include nations whose partnership is very limited, such as Pakistan and India both participating in a counter-terrorism operation. The level of participation offered by such partners in intelligence and C2 processes will depend in part on their ability to protect and manage their own assets. This may often mean protecting intelligence sources and methods, information gathering capabilities, and battlefield capabilities as well as specific situation assessment or plan products, while making some of those information elements available to coalition command to drive the command and control process. A related concern is the level of security maintained once content leaves, for example, US command and control enclaves and is posted within enclaves which might have less strident security features in place. Information assurance beyond access control must be considered an equally important source of constraints on how information is handled. From an information assurance point of view, where information is coming from, and how it was processed to produce decision-level information, is as important as what is done with the information or who has access to the product. Finally, because of the complexity of a large coalition command and control enterprise, the business processes of the enterprise itself must be enforced. These business processes define what each coalition organization is responsible (and allowed) to provide to the overall enterprise as well as what they are allowed to demand from the enterprise. These additional constraints on information access, dissemination, and processing are necessary to maintain a coherent and stable set of processes that can effectively perform command and control information processing and decision making activities without grinding to a halt or making conflicting or poorly informed decisions.

In short, the issues we have been addressing in our research are driven by the need to stand up a working command and control environment with supporting technology that integrates diverse information producer and consumer organizations. We are much less concerned about how an individual operator or staff member accesses information, and much more concerned with how organizations that do not typically work in concert pull together into a working enterprise. We believe the best conceptual example of this problem is the Joint Battlespace Infosphere model, developed in 1997 by a U.S. Air Force Scientific Advisory Board ad hoc study committee. This concept is the basis for several related US Air Force sponsored research initiatives, and provides the conceptual framework in which many of our working examples and problems are framed (some under US Air Force / AFRL sponsorship, some under independent sponsorship). We have also been fortunate to explore related problems in information discovery and information sharing across enclave boundaries in the military intelligence community. As we look across these various operational models and the use cases that derive from them, we find ourselves consistently facing three key problem areas: interoperability, enforcement of information management policy, and enforcement of enterprise information processes. In looking at mechanisms to address these issues, we have found a high degree of utility from two particular classes of technology: Semantic Object Webs, and Information Management Agents.

3 The Semantic Object Web ™ – Organizing Content with Semantic Underpinnings

One very fruitful area of technology exploration at ISX has focused on the exploitation of semantic underpinnings to aid in information discovery and interoperability. Across many domains, including command and control, highly structured databases and standard publication formats (not to mention paper documents) are giving way to rapid publication and update of less structured content augmented with XML-based metadata and embedded markup. XML and derivative tag-language based interoperability is becoming a de facto standard mechanism for externalizing information, for packaging information for transport to targeted applications, and for allowing content to be exploited in ways often unanticipated by the information producer. These approaches are driven by the recognition that, unlike interoperability based solely on shared data models and common format shared repositories. By adding a layer of semantic mapping between information (content and services) and a shared domain ontology, these approaches have demonstrated advantages of more robust interoperability between diverse information providers and consumers. Just as HTML provided the means to express how information format should be interpreted by a browser, XML-based domain languages have provided the means to guide how an information exploitation capability should understand the content provided by an information source. Increasingly, these techniques are being exploited not only to provide enhanced interoperability, but to provide a model of the information content or services provided by a source, or the information requirements of a consumer. More advanced semantic markup languages like the DARPA Agent Markup Language (DAML), combined with an emerging generation of DAML-based tools, provide the mechanisms to develop and exploit embedded markup and metadata for both unstructured sources (such as analysis reports) and highly structured sources (such as databases). These tools give us the ability to not only understand the semantic mappings necessary to enable interoperability, but provide the means for humans or software agents to browse large collections of information elements and to explore complex links and relationships. On the information integration side, this ability to build a highly connect space of semantic links provides the means to integrate information that is divers in both content and structure. On the exploitation side, this enables much more powerful information discovery and retrieval as well as the advantages of interoperability noted above.

Rapidly escalating trends toward the exploitation of semantic markup and metadata offer potentially powerful approaches that promise to meet some of the most challenging problems in the coalition environment. First, consider the nature of future coalition operational problems. Traditional fights with traditional enemies are becoming increasingly infrequent. New kinds of battles, against non-traditional adversaries, highly asymmetric threats, and non-traditional battlefields are becoming increasingly frequent. Many of the information processes and products of today's command and control environments will be hard pressed to meet the demands of these new classes of problems. For example, when the US and its coalition partners take on Al Qaeda operations worldwide, new information processes will be needed. The products of intelligence gathering and operational planning processes will not be able to take days or weeks to massage data into standardized formats, resolve uncertainties and contradictions, and publish "authoritative" databases. Nor will operational plans be able to rely on standardized publications at regular intervals to keep everyone "on the plan." Reaction cycles are getting too short, and information is getting stale too quickly. Information will need to come from whatever sources are available, and will have to be updated and shared in increasingly raw form, with tools to help information analysts identify and augment key content, and for consumers to quickly find and extract the content they need. Second, consider the changing nature of a real-world coalition environment and the roles of coalition players. In the past, coalition has often meant a U.S. run operation, where other players offer cooperation and support, even battlefield resources, but the primary challenge was coordination of forces. Increasingly, we will face problems where timely reaction to intelligence dominates over our ability to coordinated force as the key to the winning formula, and where organizations across the coalition play important roles in keeping the flow of global intelligence connected to the coordination of forces. In these coalition problems, the flexibility of semantically grounded representation offers

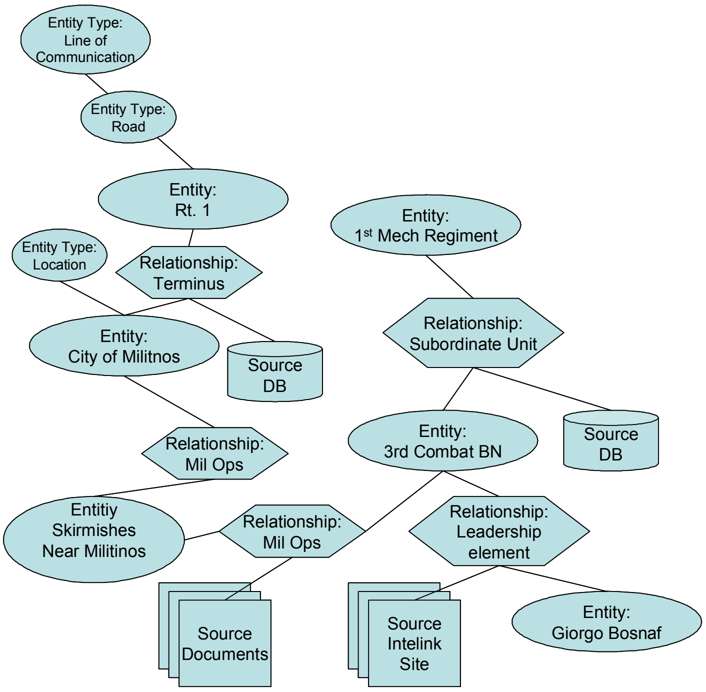


Figure 1: Semantic Object Web example

the potential key to information discovery and sharing between very diverse organization, each with their own tools, their own representations, and their own business processes. Such representations promise to form the basis for interoperability and exploitation of information products without detailed, pre-defined agreements on definitions of specific information product formats. This technology is moving the military command and control world rapidly from a data format-based model of interoperability to a semantics-based model of interoperability.

In our approach to the exploitation of semantic tag languages (in our case DAML being the ontologically grounded language of choice), we are building on the notion of the Semantic Object Web. Recognizing the Web as today's best example of a large, shared information structure, Tim Berners-Lee coined the term "Semantic Web" to describe the evolutionary model of a future Web that allowed machines, not just people, to exploit content and services. Using an expressive semantic markup that makes content, metadata, and services available and understandable, and using tools that exploit shared semantics represented as ontologies, user applications or automated software agents could both publish and consume information from the Web. DAML represents the current results of a body of DARPA-funded researchers pursuing the notion of the Semantic Web by providing the semantic standards to extend simple XML. We have found the notion of the Semantic Web a powerful metaphor for the kinds of information exploitation needed to support future command and control environments. The Semantic Object Web implements some of our ideas about how to take advantage of a practical Semantic Web implementation for distributed enterprise information sharing.

The Semantic Object Web takes an object-oriented approach to modeling the available content in a Semantic Web. This model consists of a semantic network of objects, each of which represents an entity whose type is defined in a domain ontology, with links to other entity objects based on relationships also defined in a domain ontology. The Semantic Object Web is built by processing the available markup-based metadata and content markup, along with XML-externalized database schema and content (schema to provide exploitable structure, and content to help resolve enitiy and relationship unification). As each element of information is processed in the construction of the Semantic Object Web, an inference engine (currently the PARKA system, developed by University of Maryland) maps the new content into the existing model. The inference engine operates over the new content, existing model, and a pre-defined set of ontological specifications and mappings to unify references to entities in the domain, to recognize information that implies relationships between these entities, and to identify references to content about the attributes of the entities. Each object maintains a set of pointers to source information related to its definition and its attributes, and each relationship object is used to maintain a set of pointers to the source material supporting each inferred relationship. These objects and link references can be stored in one or more distributable databases for efficient exploitation by humans with appropriate query and navigation tools, or by software agents capable of doing an initial query and then "walking the structures" to search for complex object/relationship substructures of interest. The Semantic Object Web can be thought of as an efficient agent-exploitable index into the content of a Semantic Web. Having found the right entities and relationships in the "index" model, an agent can pursue the specific links to supporting source material to extract and deliver necessary content.

Our initial experiments offer promising results to address the problems of information integration, discovery, retrieval, and interoperability across organizational boundaries. We have demonstrated the ability to integrate information metadata and markup from unique sub-domain ontologies into a single Semantic Object Web by creating fairly simple mappings between ontologies. And while our initial experiments were based on human query and navigation tools, we believe that agent-based exploitation of these structures are quite practical, as we hope to demonstrate in future work. Given such a capability, we can address some key problems in coalition interoperability by breaking down semantic barriers, providing mechanisms for the discovery of relevant information, and providing and extensible information architecture for a large, diverse, and distributed coalition community.

4 Agent-Based Information Management

Another focus area for coalition-relevant research at ISX is in the use of software Information Management Agents (supported by the COABS Grid services) as rapidly and dynamically composable components for both accessing information and for implementing information management strategies. Considering some of the key aspects of information management problems facing the coalition enterprise, we believe such approaches offer a useful and necessary capability. We have observed that these approaches can provide services to help information requestors find and access relevant information, transform it into useful abstractions or formats, deliver it on demand, and monitor for relevant changes.

Our exploration of agent-based approaches to information management stems from our recognition of some key problems encountered when trying to interoperate across organizational information processes, each with its own semantics for information representation. The most apparent problem is the simple semantic data model mismatch between organizational models. When an organization establishes a flow of information into its processes, it must be able to exploit that information with tools that were designed to work with a specific data representation, which captures a specific level of abstraction and aggregation of source information. To establish a flow of useable information into an organization's process, several steps may be necessary to bridge this semantic mismatch. First, the consumer must be able to find the kind of information sources it wants. This might not be a simple mapping, as the available data might be represented using different semantics, and may even need to be composed out of various pieces of information aggregated or assembled into the needed product. Information sources must be identified. Abstractions and representation in the request must be deconstructed to find a workable mapping between requestor and provider information elements. Queries must be decomposed to match those mappings, and the results passed through the right aggregation and fusion processes to create the desired content. Finally, the resulting information must be transformed into the right abstraction and format to enable interoperability with the consumer's tools.

A second class of problem involves the enforcement of constraints on these information flows. Here, our concern is on enforcing various restrictions on what kinds of information or services an organization can exploit, and on what kinds of information content and services that organization can make available. Today, such constraints are largely enforced by limiting access to entire broad classes of information, such as restricting access to systems which operate at certain levels of classification, or limiting access to information from certain sources. A more desirable capability would be to apply policies that define more specific information protection and process management objectives of the organizations involved, and to provide an automated mechanism to make sure that these policies are properly applied. Such policies could express the intent to protect certain sources or methods, to limit dissemination of certain sensitive intelligence or operational plans, and to enforce specific role-based information service constraints on partner organizations. Ideally, consideration of such policies would be considered as part of the process of establishing information flows between organizations, and their application would allow some flexibility in meeting policy requirements while supporting key information requirements. For example, a policy to protect an information source might be implementable (perhaps with human oversight and approval) by delivering needed content only in an aggregate form, where source-specific relationships are abstracted away. Our research is focused on the role Information Management Agents can play in both implementing interoperability bridges between disparate sources and consumers, and in implementing mechanisms to enforce information management policies and processes.

Work at ISX has explored several specific classes of agent-based information management problems relevant to the objectives described above. In the first, our goal was to de-couple information consumers from information sources, and to demonstrate that software agents could enable more loosely coupled modes of interoperability. In our experiments, we provided a mechanism to handle information requests from consumers which would typically access some shared, common-format repository. Instead, we introduced a collection of software agents designed to implement these information requests by dynamically organizing various agent-base functions operating across a range of information sources. Facilitation agents implemented selected classes of information requests by soliciting and organizing the activities of various other functional agents. Some of these agents provided the ability to decompose the information request into finer grained requests to match available information source services, while others provided the ability to re-assemble or re-aggregate the component results to satisfy the information request. Other agents provided translation services, mapping information requests into the right semantics to match information sources, and translating the resulting information into the form needed by the requestor. And of course, a collection of service brokering and matching agents were required to help the facilitators find the right agents to compose into a working access service.

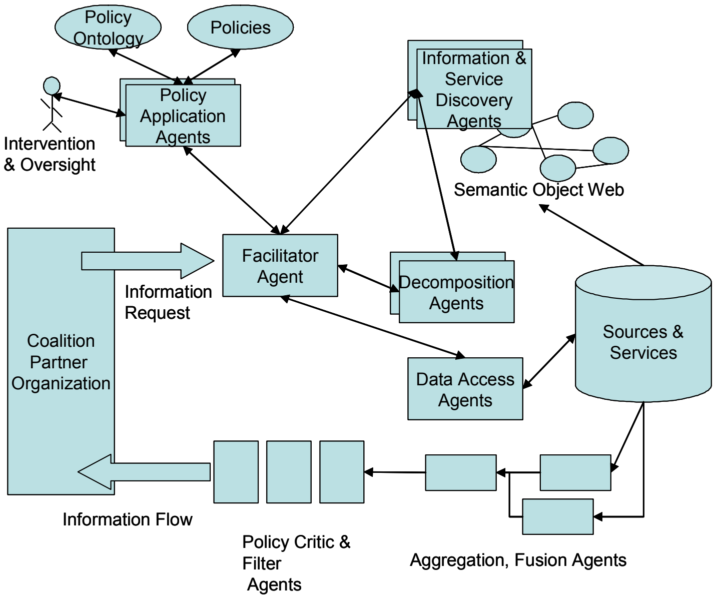


Figure 2: Information Management Agents concept

In a second research project, funded by the USAF / AFRL Joint Battlespace Infosphere (JBI) project, we use a similar agent-based approach to augment an existing publish and subscribe service on the JBI platform. In this work, we provided a lightweight agent framework to construct "fuselet" agents capable of simple information aggregation and transformation operations. Given a consumer request to the publish-subscribe mechanism which fails to match any available information publication source, the failed request is handed over to the "fuselet" mechanism. This mechanism is capable of decomposing the request, matching the decomposed elements to available subscriptions, establishing those subscriptions, and re-composing the results for delivery to the requestor. The re-composition might be as simple performing simple set aggregation or counting, or might involve a lightweight fusion operator to combine multiple inputs into the desired information. In the end, transformational agents turn the acquired information into the desired abstraction and format for delivery as a new "subscription source."

In both of these initial experiments, we were able to demonstrate that simple agent-based information management functions could be automatically composed on demand to implement fairly complex information management tasks. While these experiments focused on the interoperability-oriented aspects of decomposing requests, gathering data, recomposing results, and transforming semantics, they illustrate a useful level of complexity and robustness we believe could be applicable in coalition- related information policy and process management problems. Currently proposed extensions to this work will attempt to generalize these agents to go beyond information access, and to directly implement constraints imposed on information access and dissemination by organizational policy. In these experiments, we intend to provide facilitation agents that serve as critics on information subscription or publication requests from a producer or consumer organization to the JBI platform. By checking these requests against a hierarchy of organizational policy models and a process model for the command and control enterprise, these agents can either reject un-allowed requests, or can assemble a collection of information transformation agents that can filter or abstract information to meet policy-imposed or process-imposed constraints.

5 The Future: Agents On The Semantic Object Web

In our future research, we hope to begin to put together many of the pieces described in the previous sections to provide a more comprehensive experimental model for coalition information management. Our focus will be the exploitation of the Semantic Object Web by software agents designed to implement both interoperability and information access and dissemination management services.

To support interoperability, we intend to extend the notions of agent-based decomposition, search, and retrieval to exploit semantic structure. Information requests today depend wholly on shared data models, and queries typically request known data structures or content elements. Today, we can provide the ability to query the Semantic Object Web using more general ontology-based queries, such as "Give me all the entities of type Threat" and expect to find matches to various sub-types and specializations of threat. In the near future, we anticipate asking agents to handle much more complex queries, like "Find any link between John Hatfield and David McCoy," or more relevant to our domain, "Find any hostile organizations or forces that might resist SOF team deployment in Village X." And we will expect these agents to deliver the content, pulled from the best available sources, delivered in a form we can use. Such capabilities offer the promise of much richer interaction between organizational elements, based on better indexing and better retrieval of all available information despite organizational semantic barriers.

To support better information management, we intend to explore the use of Information Management Agent-based implementation of information access and dissemination control policies and enterprise process policies represented as Semantic Object Web structures. While we are very interested in ongoing work in policy servers and policy ontology, we intend to specifically explore the power of semantic indexing and retrieval mechanisms to allow agents to quickly find relevant policy elements, and to intelligently apply those policies to specific information instances. Using these techniques, we believe that agents will be able to consider metadata about specific object abstractions and relationships, individual attributes, and specific sources and source types as the basis of dissemination constraints. Given this capability, a policy might be expressed in fairly abstract terms that address categories or types of sources, content attributes and relationships, representational abstractions. Agent critics could use the ontology-base inferences in the Semantic Object Web to compare these policies to actual instances of information access or publication requests.

In summary, our initial research activities have shown both the potential power and practicality of both Semantic Object Web and Information Integration Agent technologies to address key problems of information interoperability and management in domains like coalition operations. This early work features successful experiments with problems whose characteristics and complexities match those we anticipate for more comprehensive coalition problems. However, we also recognize that we have only scratched the surface of these problems. We expect to make significant steps forward by transitioning our agent capabilities to exploit our semantic information substrate both for content and services exploitation, and for exploitation of information management policy. We also are eager to identify opportunities for joint experiments with related technology research to help expend these ideas and their application to real problems.

6 References

Berners-Lee, Tim. 1999. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. HarperCollins.

Berners-Lee, Tim, James Hendler, and Ora Lassila. 2001. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American* (May, 2001).

DARPA Agent Markup Language (**DAML**) Project and Language: http://www.daml.org

Parka Knowledge Representation System: http://www.cs.umd.edu/projects/plus/Parka/

Simple Ontology HTML Extensions (**SHOE**) Project and Language: http://www.cs.umd.edu/projects/plus/SHOE/

Resource Description Framework (**RDF**): http://www.w3.org/RDF/ (RDF Schemas: http://www.w3.org/TR/2000/CR-rdf-schema-20000327/)

World Wide Web Consortium's Semantic Web Activity: http://www.w3c.org/2001/sw/

World Wide Web Consortium's Web-Ontology Working Group: http://www.w3c.org/2001/sw/WebOnt/

Heflin, Jeff. 2001. Towards the Semantic Web: Knowledge Representation in a Dynamic Distributed Environment. Ph.D. diss., Dept. of Computer Science, Univ. of Maryland at College Park.

Fuselet Project Final Report, ISX Corp., Jan 2002

Marmelstein, R., *Force Template White Paper*, (Notes from the AFRL Minnowbrook Joint Battlespace Infosphere workshop, Force Template Working Group), AFRL Nov 2001

Linderman, M*., The JBI Platform*, Briefing to the Minnowbrook Joint Battlespace Infosphere workshop, Nov. 2001

McCarthy, J. (ed.), *Building the Joint Battlespace Infospher*e, US Air Force Scientific Advisory Board, Summer Study Report, Dec. 1999

McCarthy, J. (ed.), *Information Management to Support the Warrior*, US Air Force Scientific Advisory Board, Summer Study Report, Nov. 1998

JACK: A System for Building Holonic Coalitions

Martyn Fletcher

Agent Oriented Software Ltd
Mill Lane, Cambridge CB2 1RX, UK
martyn.fletcher@agent-oriented.co.uk

Abstract. Coalitions between military and non-government organisations to manage operations other than war (OOTW), e.g. earthquake evacuations or food distribution to refugees require sophisticated knowledge management, decentralised control, and the ability to conduct flexible negotiations. Holonic systems are a research topic well suited to these requirements as they treat each organisation as an autonomous cooperative entity that simultaneously adopts a part-whole relationship within the coalition. This paper discusses a model of coalitions based on the application of holonic principles. The paper also outlines how this model could be implemented using JACK, the flagship software development product from Agent Oriented Systems.

1 Introduction

Arthur Koestler initially proposed the principles of holonics (Koestler, 1967). He postulated that many biological and social organizations display simultaneously part-whole relationships. In other words, every entity is self-contained, while concurrently being an individual member of a larger collective. Thus each entity (or *holon*) must act autonomously and cooperatively to achieve the goals of itself and the wider system. Hence the entire system can be seen as a *holarchy*, i.e. a recursive hierarchy or heterarchy of holons with no centralized control, which relies on collaboration among holons to achieve the system's goals. These generic ideas have been expanded on and delivered into agile manufacturing scenarios, and much has been learnt of holons' behaviour. Now it is time to apply these abstract concepts and the experiences gained in deploying such systems into other domains. A very suitable domain is coalition management as part of operations other than war (Tate, 1999). Here every military force and non-government agency can be viewed as a holon. The holarchy structure is then created in a 'bottom-up' manner via the aggregation of holons (royal air force, red cross and so on) to satisfy the requirements and services needed for handling the crisis.

A suitable foundation for implementing such holons is the agent-based development environment JACK from Agent Oriented Software (AOS, 2001). JACK is a realization of the belief-desire-intention model of agency and is one of the very few industry-strength systems for building autonomous-agent and team-based applications. This commercial product has a history of solid implementations through being deployed into defence, air traffic control and telecommunications environments. The utilization of JACK will provide a firm foundation for experimenting with agent-based coalition ideas during OOTW (Maughan, 2001) (Thomas, 2000). In the paper we illustrate how JACK can be applied to build, manage and control our new vision of holonic coalitions. The paper is structured to reflect these conceptual design and implementation issues, together with providing a simple illustrative example.

2 Conceptual Model of Holonic Coalitions

Holonic systems represent a novel paradigm for addressing some of the most critical problems encountered by military, charity and non-governmental organisations as they come to grips with the 21st century theatre of relief and humanitarian operations. These problems include:

- The demand from stricken governments and aid charities to have their specific relief/humanitarian requirements delivered to the crisis region with *short* 'request-to-deployment' times. People cannot wait a year for shelter, food or medical supplies to be delivered, or be evacuated from a hostile environment; they need it in 2 days.
- The need to support *mass customisation* of OOTW efforts, i.e. 'relief-to-order' rather than having dedicated military and non-government agencies on permanent standby ready to be deployed anywhere around the globe. This helps the agencies to regularly react to rush operations and new relief specifications.
- The need to have *tightly and loosely integrated* cooperation between agencies and hold/exchange appropriate private, protected and public knowledge.
- The requirement to cope with a hybrid combination of operational *variety* and *volume* within a single crisis area. Agencies are discovering that there is a need to distribute food to 1,000,000 refugees and conduct military actions against an enemy simultaneously. Traditional thinking and technology is not geared to this imbalance.

The benefits of applying holonic technology to OOTW include, but are not limited to:

- The holonic model helps the various agencies and military forces to make *maximal use* of available personnel, transport capacity, resources and assets to satisfy current/anticipated demand for relief. In other words, the system is able to support the re-allocation of tasks in a dynamic coalition through intelligent processes, reasoning, cooperation and negotiation – see (Shehory, 1998).
- Holonics treats alterations in coalition configurations, relief requirements, personnel, transport schedules and so forth as 'business as usual'. Moreover a holonic model reacts to the removal of, as well as introduction of new agencies, missions and information management facilities in a graceful fashion. In other words, the system is *agile* and does not crash due to changes in the operational environment.

Centralized solutions to controlling the coalitions between civil, government, charity and military organisations that satisfy such relief/humanitarian demands do not work since they are slow to react, impose operational bottlenecks and are a critical point of failure. Holonics is a decentralized 'bottom up' approach and provides principles to ensure a higher echelon of responsiveness and handling of system complexity. The building blocks (or components) of a holonic coalition architecture are called *holons* to reflect the fact that these entities behave simultaneously in an autonomous and cooperative fashion. Holonics is not just a new technology, but rather it is a system-wide philosophy for developing, configuring, and managing the next generation of OOTW where *flexibility* is paramount.

2.1 The Holonic Coalition System Architecture and Inter-Holon Cooperation

Coalitions unite people and organizations that share a common purpose. This section contains information on a few of the ideas that work toward awareness and improvement of holonic coalitions. The objective of a holonic coalition is to "attain in OOTW the benefits that a holonic system architecture has provided to intelligent manufacturing". Koestler observed the dichotomy of 'part-ness' and 'whole-ness' in natural systems (e.g. ant colonies), and devised the term ***holon*** from the Greek word ***holos*** (signifying whole) with suffix ***on*** (a particle, as in prot***on***). These generic principles have been studied in an intelligent manufacturing context to make production of high-variety low-volume artefacts more agile (Fletcher, 2001). Here we apply these same principles, and some of the experience gained as a result of these studies, to operations other than war. We model each charity (e.g. Red Crescent), civil government (e.g. local fire service) and military (e.g. Navy) agency as an autonomous cooperative holon. These agencies may be from different countries, represent diverse political/cultural/religious beliefs, have access to distinct resources/knowledge and may harbour resentment at being commanded by a military organisation. As discussed by (McFarlane and Gruver, 2001) within a manufacturing context, a holon is a basic building block in a holonic system. We propose that by applying these abstract ideas, there are the following holon types in a holonic coalition:

- *Agency holons* provide all the generic resources active in the OOTW system. Each of these agency holons is an entity (often a specialization of a particular class) that performs an action over an item. Such actions include those needed to transport, and disseminate relief materials to refugees and control the evacuation of people from hazardous environments. The items encompass food, trucks and refugees. Such agency holons include charities, military bodies, police forces, food collection companies, aircraft leasing companies, medical institutions etc.
- *Demand holons* represent the requirements of operations like relief work, peacekeeping and so forth. The requirements often originate from either external bodies (e.g. a stricken government), from other departments within an active organisation (for example the Army asking the Navy to supply a ship) or from anticipated need (for instance when the forecasts for a country indicate that a crop will fail next year then it is wise for an aid agency to stock pile food in readiness). These holons also provide knowledge on how to achieve the mission objectives, can offer expert advice, and may also act as an information server to disseminate knowledge among the other holons in the coalition. Each can be re-used in the scope of different operations and each could negotiate with various agency holons in order to secure the desired services. In other words, each demand holon is an active entity responsible for performing the crisis management work correctly and on time, while explicitly capturing all data and information processing needed for a specific job. Such demand holons might represent the need to evacuate 100,000 people after an earthquake and give multiple options how this could be achieved (e.g. by aircraft quickly, or alternatively via road slowly and so need a temporary shelter).

We hypothesise that the entire holonic coalition system can be modelled as a *holarchy*, namely a recursive aggregation of *cooperation domains* (see below). These cooperation domains solve a set of decomposed and inter-related OOTW tasks. Every task is modelled as a demand holon. The notion of a holarchy (see Figure 1) simplifies our architecture because we only need consider the structure of a single cooperation domain and the interactions agency and demand holons have through it. Using this holarchy principle, a simple *holonic team* is constructed to manage each cooperation domain. The members of this team could be either agency holons or other sub-teams (in a recursive manner). The lowest level holons are always *agency holons.* The structure created by this holarchy is specific for each crisis being managed and can be dynamic because:

1. Agency holons arrive/leave when their schedules or commitments change.
2. Demand holons enter/exit when their corresponding crisis task or knowledge is required or no longer needed.

Agency holons respond to task requests from the cooperation domains' demand holons that they are interested in participating within. Therefore either interaction is carried out (through the existing cooperation domain) or new crisis management tasks are generated (as demand holons) according to these responses. If a crisis management task cannot be executed due to a lack of an agency's resources (for instance inadequate equipment or peoples' skills) then the task may be altered. Otherwise a new functional component could be introduced into a holon to provide the necessary resource and so satisfy the cooperation domain's requirements.

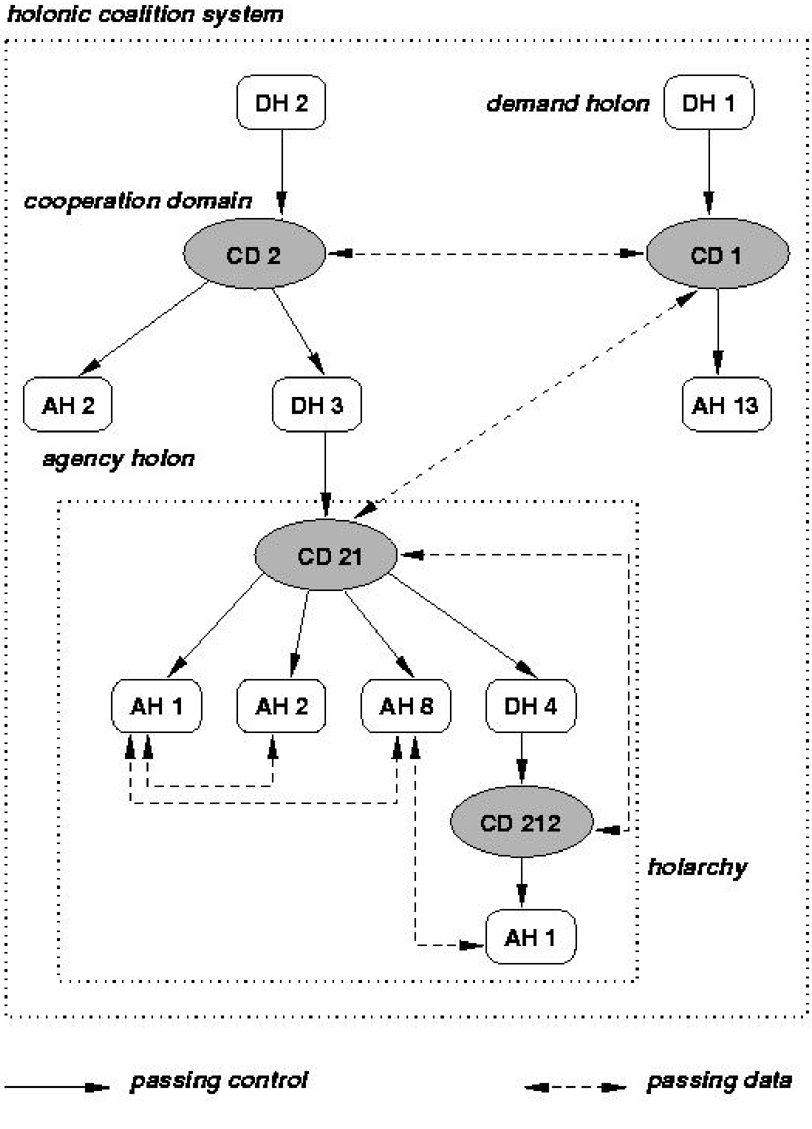


Figure 1: Coalitions, Cooperation Domains and Holons.

A cooperation domain is a logical space through which: (i) agent-based holons communicate and operate, and (ii) a context is provided where holons may locate, contact and interact with each other. We assert that a cooperation domain cannot exist by itself, and that all cooperation domains must be dynamically generated by the needs and services of individual holons. The following premises are valid with respect to cooperation domains:

- A holonic coalition system must contain at least one cooperation domain.
- An agency holon can be simultaneously a member of one or more cooperation domains.
- A cooperation domain can only exist if it has: (i) a demand holon; plus (ii) one or more member agency holons.

A cooperation domain comprises the following key elements:

- Coordination and information management facilities. These can be handled by a demand holon acting with the role of a coordinator to administrate a joint task, and retain/disseminate knowledge among agency holons.
- Data structures through which holons may write and read knowledge to control cooperation, e.g. querying the value of a variable that indicates the status of a joint food distribution task by the Red Cross and Air Force.
- Logical framework for connecting together heterogeneous holons. We model this property using a temporary alliance between a coordinator (demand) holon and one or more cohort (agency) holons that support:
    - Decision making mechanisms and rules to aid holons' task planning, scheduling, negotiation, information dissemination and so forth.
    - Facilities to monitor the status of distributed tasks, and take appropriate corrections to compensate for any anomalies during execution of actions within this task.
- Physical communication platform. We assume holons pass messages using a reliable transport mechanism.

Holons can join a cooperation domain, query attributes associated with a domain, exchange information amongst one another through the cooperation domain, and depart the domain when their crisis management tasks are completed. Furthermore we visualize that a cooperation domain supports a 4-phase protocol (agreement, planning, interaction and termination) to provide a formal model of inter-holon collaboration for joint actions.

2.2 The Intra-Holon Architecture

As stated earlier, we define an agency holon as an autonomous system having a compulsory knowledge-based element and an optional physical element. For instance the Red Cross has a people to negotiate and decide how to best deploy its resources (knowledge-based element), while its resources include medical personnel, food, trucks and so on (physical element). A demand holon has no physical element. Moreover suitable interfaces to humans, other holons and the OOTW environment must also be present. In terms of its behaviour, an agency holon's knowledge-based element consists of an intelligent control system (ICS) and a processing system interface.

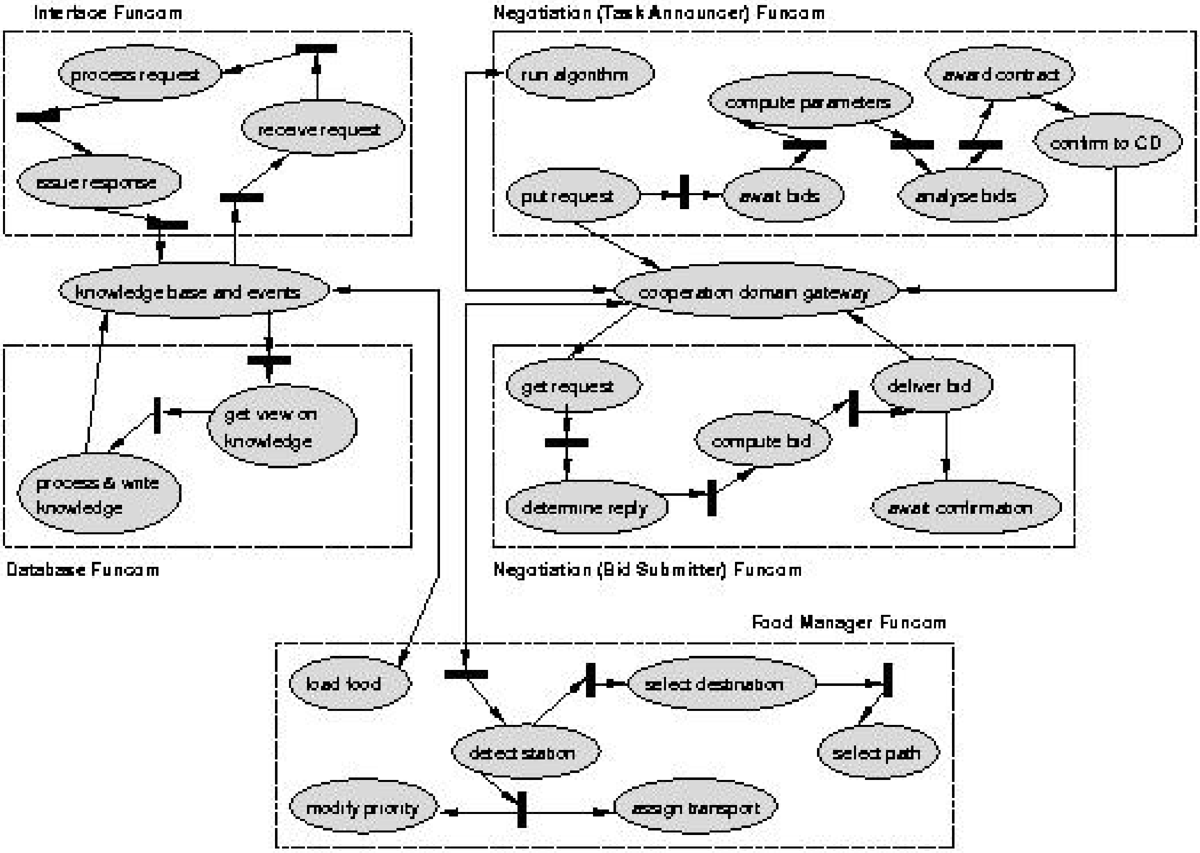


Figure 2: Generic and Application-Specific Funcoms in Demand/Agency Holons.

The ICS is responsible for the holon's internal functionality through a set of procedural rules and decision making functions. The ICS also supports cooperation via inter-holon interfaces, acquaintance modelling and so forth. In short, the intelligent control system of an agency holon is modelled as an *agent* as understood in multi-agent systems (Pechoucek, *et al.*, 2001). The processing system interface provides the individualistic skills of the agency holon and is responsible for the relief, humanitarian and military functionality according to rules and operating strategies imposed by the ICS. The processing system interface is divided into a collection of *functional components* (or *funcoms*) necessary to realize a wide variety of skills needed in operations other than war. Each funcom has independent control over its activities. For example, an Army agency holon (as part of a food distribution task) contains the subsequent functional components:

- *Load-Food*: Loads/unloads volumes of food (e.g. bags of rice) from the local sea docks or airfield into trucks.
- *Detect-Station*: Identifies the status (i.e. in the range full to starved) of food distribution stations.
- *Select-Destination*: Chooses which food distribution station should get the next delivery of goods.
- *Select-Path*: Chooses the best possible route from food entry points to the selected destination station.
- *Modify-Priority*: Alters the foods' priority or the requirements of peoples' need for this particular food type.
- *Assign-Transport*: Assigns the task of transporting the food to a given truck.

Agency holons' funcoms are designed so that they contain all the knowledge and skills required to manage operations effectively and efficiently. In this sense, we regard knowledge as being the database tuples, trigger rules events, and the beliefs, desires and intentions of the associated agents. The justification for requiring this knowledge is that it supplies both a structured semantic representation to generalize a quantity of items related to the holonic coalition system, and an anchor point to which a future implementation can be attached. Such knowledge may be classified as being local (i.e. obtained from monitoring the state of the adjacent environment), regional (i.e. that which is received from neighbouring holons) or global (namely data acquired from a directory holon). In this sense skills are the operations needed by an agency holon to utilize and maintain such knowledge, together with the corresponding manipulation of their resources (e.g. food) as they are received, transported, stored and disseminated. These skills are modelled as application-specific funcoms. As described in the Figure 2, there are also some general-purpose functional components that are used to build up each holon. These generic funcoms ensure that the agency/demand holon has sufficient autonomy and cooperation (the negotiation funcom) and can form suitable association with other agency/demand holons (the interface funcom):

- The **Negotiation (Task Announcer)** functional component operates within the demand holon to implement the first half of the entire negotiation cycle. Within the scope of the aforementioned holarchy, this funcom negotiates with the funcoms in agency holons to agree, plan, execute and terminate an operation to satisfy the OOTW demand. To achieve this planning etc, the task announcer funcom supports a number of protocols like the contract net protocol (CNP), various styles of auction, or a market economy; we consider the CNP here. The task announcer funcom is the element of the demand holon that starts a negotiation cycle; that means: (i) putting into the cooperation domain a request for some OOTW task to be accomplished, (ii) computing its parameters using the proprietary algorithms, (iii) waiting for the bids submission, (iv) analysing the bids from the various military/charity/non-government organisations, (v) running its proprietary algorithm to evaluate these bids and award the contract, and (vi) putting the confirmation of the contract into the cooperation domain.

- The **Negotiation (Bid Submitter)** functional component operates within the agency holon to implement the other half of the negotiation cycle. Briefly, this funcom replies to the task announcement to complete a negotiation cycle, in particular this means: (i) getting the OOTW task request from the cooperation domain, (ii) accepting it and deciding if reply to it using its proprietary algorithms, (iii) computing the bid using its private knowledge base and its proprietary algorithm, (iv) delivering the bid, and (v) waiting for the confirmation that award the contract to the agency holon.

- The **Interface** functional component operates within every agency and demand holon and allows the interaction between: agency-to-agency holons, agency-to-demand holons and demand-to-demand holons. The most complex and complicated of these interactions is the agency-to-agency exchange because some charity and military organisations do not want to share all their private knowledge within every other agency within the holarchy. To acquire essential information from another organisation, the agency holon uses an "information protocol" that offers a mechanism to call for information that is proprietary to the other agency holon. Of course this request can be rejected or false information can be given depending on how the two agencies consider each other. As proposed in some of the holonic manufacturing system literature (Van Brussel, *et al.*, 1998), a centralised *'staff'* holon can be used to suggest a solution (for example the allocation of how much food each of three charities should distribute in the relief operation over the next week) and the agency holons ask for such compromises and information using their respective interface functional components.

Using the above definitions, we can now model the coalitions between military and non-government holonic agents. To clarify this point, the next section presents an illustrative example of how holonic coalitions in OOTW can work.

3 An Illustrative Example

Once a basic infrastructure among the relevant agencies is established, new forms of holonic coalitions and advanced cooperation between these agencies will naturally emerge. The satisfaction of the demand holon's functional requirements in the OOTW theatre will also begin to be comprehensively supported. In particular, holonic coalition formation requires mechanisms to facilitate the controlled 'introduction' of a military, charity or non-government body (e.g. the US Marines to act as a food distributor) into the 'territory' of the relief operation (e.g. a humanitarian effort in a West African country) without impinging on the roles and attributes of its partners (e.g. the Red Crescent for giving food to Muslims, the local police force to guard food supplies and Christian Aid disseminate food to non-Muslim refugees). An initial illustrative example of this introduction is sketched out in Figure 3. The introduction is properly supported by the above holonic model, and administrates the access to selected (authorised by the agreements made when joining the relevant cooperation domain) subsets of the necessary resources (for instance food stocks, distribution personnel, trucks, helicopters etc). But this process may assume more extensive forms. Consider the case where the US Marine commander wants to 'open a window' on coalition partners to get an overall picture of how well the food distribution process is going and even have an interference on, i.e. supervise from distance and in cooperation with native-speaking local police, the dispatching processes.

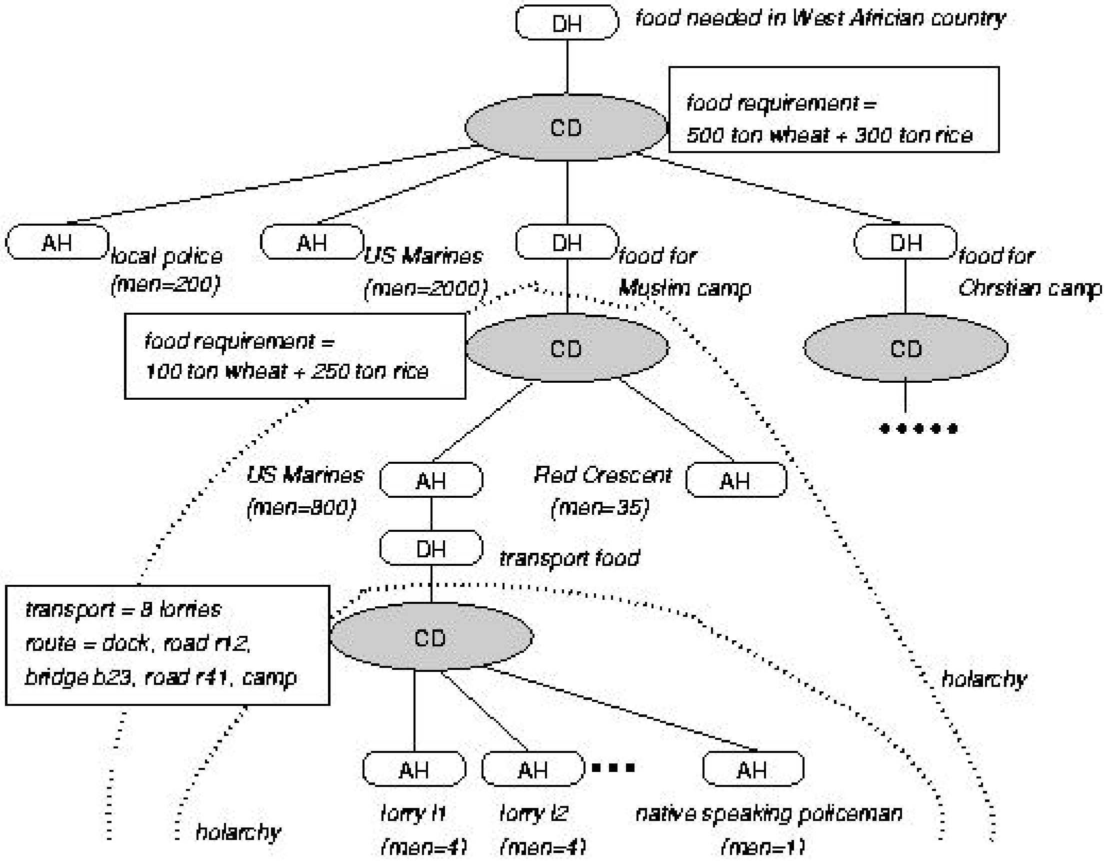


Figure 3: An Example of Holonic Coalitions – Initial View.

Such supervision represents a collection of inter-agency holon and demand-to-agency holon activities including:

- The dispatch and execution of task requests to move food from point A (e.g. the docks) to point B (a camp set up by the Red Crescent for Muslim refugees fleeing from an on-going guerrilla war in their home town).
- The monitoring of execution, for instance knowing accurately how much wheat and rice has been moved to each refugee camp, what are camps' expected demands and what additional resources will become available.
- Error diagnosis and recovery, for instance discovering that a bridge along the main route to a refugee camp has been destroyed and deciding to instead transport 2/3 of the food via a different route and use helicopters to transfer the remaining 1/3.

When viewed in a decentralised operations-other-than-war environment, the concept of *coalitions* emerges. If the coalition demands the collaboration among multiple agencies, each acting as an independent body and part of a team, each being located in remote places, being managed in different ways and adopting distinct internal structures and rules, then we have *holonic coalitions.* Therefore we need a model like that presented in section 2 with agency and demand holons, funcoms and cooperation domains to support such holonic coalitions. The design of a proper support system for holonic coalitions can benefit from contributions coming from a number of topics. These areas are at present addressed by research communities with little interaction between them. The four key contributing areas to holonic coalitions are holonic manufacturing systems, multi-agent systems, political science and defence/military studies. This paper focuses on the application of the generic principles and experiences coming from holonic manufacturing systems and how these concepts could be implemented using multi-agent technology – see next section. We leave the investigation of the roles political science (to model charity and non-government bodies) and defence/military studies (to represent military organisations) play in holonic coalitions for later work.

Coalitions between charities and military organisations have been addressed for a few years, mainly for small-scale OOTW exercises where the military body is in overall command. However the growing number of peace keeping, relief and humanitarian operations around the globe, and the requirement for military bodies not to impose on the charities and non-government agencies has opened up new opportunities for coalitions due to their lower cost, even distribution of workload and widespread acceptability. What makes holonic principles most appealing as a basis for coalitions is how they model each operation's demands and every agency involved as autonomous cooperative entities that can operate independently, collaborate and exchange knowledge in a structured fashion to achieve the OOTW objectives. However the application of holonic principles to coalitions suffers from several problems:

1. Is a solution based on today's implementations of agents, cooperation domains, funcoms, or an amalgamation versatile enough to solve the multitude of diverse problems encountered when building real holonic coalitions? The response must be a decisive no; at present, it is not. These models support some aspects of holonic behaviour very well (e.g. the concept of encapsulation of software executing at the real-time level of control), and some issues slightly less well (for instance the lack of ability to dynamically decompose a demand for some relief work into atomic tasks without pre-defined static rules). But let us not fool ourselves by saying that everything is finished. For example if we use a combined solution then where is the boundary to be set between one agency holon's autonomous activities and the actions it must manage via a loosely-coupled coalition among bodies that have distinct goals. Furthermore how are these independent technologies (with no obvious shared protocols) to interact in a cohesive manner?
2. When reasonably practical and complex OOTW domains are considered, high levels of heterogeneity are expected in the available agencies and the demands put upon them. This interoperability requirement, together with the volume, accuracy and type of knowledge to be exchanged among agencies, can degrade the agility, robustness and saleability of demand/agency holons operating in the holonic coalition system.
3. Coalitions are characterised by the short and irregular durations, also they are negotiation intensive due to the peer-to-peer level of collaboration between agencies where the military body cannot impose on the charities. These attributes mean that the coalitions can often suffer from low levels of trust, limited private resources forthcoming from non-government bodies and restricted exchange of knowledge between coalition 'partners'. This raises new challenges in what concerns the reliability and efficiency of the implemented holonic coalition system and its dependence upon the characteristics of the constituent allies.
4. The composition of the environments where holonic coalitions are to be executed are potentially unstructured and unknown. This means that it is inadequate to resort to deterministically programmed systems or monolithic centralised systems. Complementary, the increased use of military bodies to support peace keeping, refugee evacuation and so forth requires multiple interaction periods, of varying durations, with agencies that: (i) might not behave in a altruistic fashion; or (ii) have their own goals to achieve beyond the present coalition's scope.

In order to cope with the mentioned difficulties, an approach based on autonomous agent and multi-agent system technologies has been developed. Multi-agent systems originate from research into Distributed Artificial Intelligence (DAI) (Hewitt, 1981) and use mentalist approaches to problem solving by imitating human actions and interactions. These concepts are often based on speech acts (Searle, 1969) or the belief-desire-intention (BDI) model. Like people, such models are inherently unpredictable, can be unstable and may make wildly different decisions based on uncertain knowledge. Hence agents may not be best suited for every real-world coalition case, especially whose where there exist safety critical and secrecy constraints of tasks. Yet their benefits are numerous (e.g. fault-tolerance, dynamic reconfiguration etc) and so their exploitation is ensured. The BDI model was initially introduced as the foundation for single-agent architectures by (Bratman, *et al.*, 1988) and was developed further by, amongst others, (Rao and Georgeff, 1995). Since its conception, the BDI scheme has become a solid foundation for research into multi-agent architectures and their application to several problem domains. The scheme defines both:

- An agent's internal processing through a set of mental categories with a control framework for the rational selection of action plans to satisfy goals using some knowledge of the environment.
- A team (as part of *Team Oriented Programming*) that encapsulates multiple agents into a group with a concerted goal and set of beliefs. This group then has a specific coordinated activity to perform, and so assigns roles to independent agents to get the joint task achieved.

These principles have been further extended by Agent Oriented Software (AOS, 2001), made into a commercial product called JACK (Howden, *et al*., 2001) and has been successfully applied to control various application domains including a manufacturing cell at the University of Cambridge's Institute for Manufacturing (Jarvis, *et al*., 2001). From which we have gained a lot of valuable experience in deploying agent-based holons. Here we use some of this experience to build coalitions based on holonic ideas using JACK.

4 Building Holonic Coalitions with JACK

JACK Intelligent Agents is an agent-oriented development environment that is built on top of, and is fully integrated with, the Java programming language. JACK consists of:

- *JACK Agent Language (JAL)*. JAL encompasses Java and is used by software engineers to build holonic coalition systems by providing a 'super-set' of agent-oriented constructs. JAL extends Java by: (i) Providing new base classes, methods and interfaces; (ii) Extending Java syntax to support new classes, declarations and reasoning method statements; and (iii) Providing semantic extensions to support agent-oriented execution.
- *JACK Agent Compiler*. This compiler pre-processes JAL source files and converts them into standard Java. This can then be compiled into Java Virtual Machine code and executed upon some target holonic system.
- *JACK Agent Kernel*. This kernel provides all the runtime facilities to execute these holonic agent constructs (written in JAL).

The structure of a JACK agent and how it works is as follows: Each agency and demand holon is an instance of a particular agent class, and interacts with its physical OOTW environment through a set of functions that read data in from people in the physical operations theatre (e.g. information is supplied by charity people with Internet mobile phones and PDAs, or through military personnel with laptop computers and secure satellite communications systems etc) and write instructions out to the same human beings. Every agent representing an agency holon has one or more capabilities modelled as application-specific funcoms (for example fault diagnosis, scheduling and the food manager as shown in Figure 2) that it can perform. Each capability encapsulates a number of goals (or desires), plans (or intentions), knowledge (or beliefs) and event templates that the agent will react to.

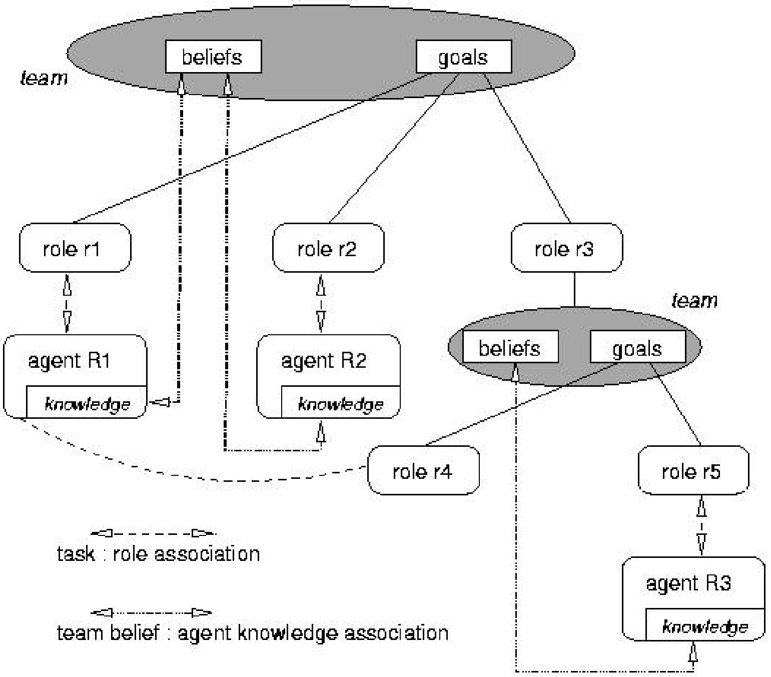


Figure 4: Team Oriented Programming.

When this agent-based agency holon is instantiated into the holonic coalition system, it will wait until it receives an event that it must respond to, or is presented with a goal. Agent-based demand holons have equivalent functionality for handling where and when the operations are needed and their parameters. Events are used to support reactive behaviour in the agents while goals are utilized to focus an agent's proactive behaviour. When it receives such an event (or goal) then it searches for and then executes a suitable plan(s) to handle an instance of this event type. Such event handling may be either synchronous or asynchronous to when it was posted. The execution of this plan may demand: (i) the exchange of data and instructions with other holonic agents via a suitable protocol, (ii) interaction with the agent's private permanent database relations, or (iii) manipulation of other Java-based non-permanent data structures. The plan being executed can create other sub-tasks, which in turn may generate sub-sub-tasks, and so on, thus creating a recursive hierarchy, that is adequate for modelling the conceptual holarchy organisation outlined above. Each plan may either succeed or fail, in which case the agent may attempt to execute another plan.

A *simple team* is an extension of JACK and allows for the definition of agent groups where coordination of joint activities is distributed across team members. In our holonic framework, each cooperation domain is modelled as a team in order to help with group functionality and share workload. JACK also supports teamwork by providing a set of concurrency management and event handling functions. This team-engineering concept is flexible and does not impose rigid criteria on the formation of multi-agent collectives or on the dissemination of beliefs among team members. These ideas are shown in Figure 4. The system developer has the freedom to choose the subsequent team attributes to build holonic coalitions:

What the team is capable of doing, i.e. what is the team's overall goal? In our holonic coalition context, the goal is to ensure optimal and efficient movement of food from docks to various refugee camps through the alliance of distinct non-government and military bodies as they enter, leave and reconfigure their actions/interactions in the coalition.

What are the *roles* of individual member agency and demand holonic agents within the scope of achieving this team's goal? Here the roles assigned can be either:

- One demand holon per multi-body operation to represent the task’s parameters, and some functionality to monitor and advise on how the task should be handled by the available agency holons. In our earlier example, the demand holons include ‘food needed in a West African country’ at the highest level, going down the recursive tree, to ‘transport food to Muslim refugee camp’.
- One coordinator agency holonic agent responsible for managing the operation and one or more coordinatee agency holonic agents that obey the commands given them by the coordinator. This is a master slave relationship where the coordinator identifies the potential operation, isolates what options can be taken, assigns tasks, issues commands to other agency holons to achieve the goal and monitors these actions to ensure success.
- Multiple negotiator agency holonic agents responsible for collaborating together to discover and execute the best overall food dissemination strategy. This is a peer-to-peer relationship where all the agency holonic agents have an equal vote on what joint action to take.

What is the assignment of roles to actual team members? Here we allocate the coordinator role to agency holon *US Marines* and coordinatee roles to *Local Police, Red Crescent* and *Christian Aid*. The allocation of the roles is also bound to any resource-dependent constraints on the task. For **hard** resource-bound tasks, e.g. some chilled food must be delivered by time *t1* using refrigerated lorry *l12*, the action must always be completed by the specified due time. While for **soft** resource-bound tasks, the actions must be completed to a certain percentage of occasions by the set finish time and using the requested resource. To reflect this distinction, the role is given to a particular agency holonic agent the completion time, resource specifications and the ratio is also presented. For **hard** resource-bound tasks the ratio is 100%, for **non** resource-bound tasks (i.e. common agent-based actions) the ratio is 0%, and for **soft** resource-bound tasks the ratio is in the range 1% …. 99%.

What functional components are needed to form this particular class of team? Here we can say that the coordinator agency holon must have suitable plans to discover, determine, asses potential food logistics and the ability to formulate a solution. While the coordinatee agency holons need to execute the distribution plan assigned to them and report their status. In other words, a number of algorithms are needed at each different type of holonic agent.

When is a team willing to take on a particular role within the confines of another team? Namely what is the recursive nature of these agency holon aggregations. Let us illustrate by example, suppose military organisation *US Marines* enters into a cooperation domain with non-government charity *Christian Aid* and the resolution of this food transport operation is that *Christian Aid* should stop moving medical equipment while *US Marines* uses some common lorries to proceed through their shared hostile working envelope, e.g. an area with an on-going armed conflict. This means that *Christian Aid* will not meet its expected food delivery schedule at its refugee camp to distribute food the people. Hence *Christian Aid* must resolve this secondary coordination problem (e.g. getting the food delivered to the camp via another transport option like using police vehicles via a subordinate team) without impinging on the solution of the top-level team.

How is behaviour across the team members to be coordinated? What techniques and methods are needed to ensure synchronised mutually-agreed actions are taken throughout the team community. Here we hypothesise that a simple publisher subscriber model will suffice: the agency holon representing the *US Marines* writes knowledge to the cooperation domain, suitable monitoring functions are invoked, the allied bodies like *Christian Aid* are informed, and act according to their prescribed role. More sophisticated contract bidding, auction or economic market solutions could be used especially when the agency holonic agents might wish not to disclose private knowledge.

How is the knowledge in the team to be encoded, disseminated, and replicated between autonomous team member agents? We postulate that suitable ontologies, multi-casting, and consistency protocols can be called upon respectively. No system stands alone, and so a holonic coalition system built in JACK must work both on its own and integrate tightly with other agent-based solutions. JACK agents are not compliant with the existing standards from the Foundation for Intelligent Physical Agents (FIPA, 2001). Therefore you cannot put together intelligent software agents constructed in JACK and other systems like CPlanT (Pechoucek, *et al.*, 2001) in a haphazard way and expect them to work. There are some alternatives that can be used to ensure these heterogeneous agents can integrate smoothly:

- Send and receive events through a common operational environment, e.g. the battlefield and OOTW theatre that both types of agents can observe.
- Send/receive knowledge to a shared database that generates appropriate events that both agent types can utilise.
- Have a bridge agent to convert messages from FIPA-compliant format to a suitable format for JACK agents to use, and vice versa.

A team is defined in terms of the roles played by its members and so may be composed of either autonomous agents (agency/demand holonic agents) or subordinate teams (with the lowest level of this recursive organisation always being an agency holonic agent). In short, the teams principle allows for the encapsulation and engineering of coordinated activity among heterogeneous holonic agents. The teams concept extends the notion of autonomous agents into multi-agent systems via the association of tasks with roles. Yet each agency holonic agent remains autonomous and is privately responsible with determining how its plans can best satisfy the role(s) assigned to it. We now present some JAL code for implementing such holonic coalitions. These coalitions are relatively well defined, with several roles, and involve a reasonable amount of parallelism.

```
package aos.simpleteam.core;
import aos.simpleteams.rt.*;

team HolonicCoalition extends SimpleTeam {
        #requires role CoordinatorHolon coord_h;
        #requires role FoodTransporters[2] trans_h;
        #requires role FoodDistributer dist_h;
        #requires role Interrupter int_h;

        #uses plan Transport_and_Distribute_Food;
}
```

We note that the team has two distinct food transporters; otherwise the roles are singular. Since only the food transporters are distinct (i.e. ground-based transport – lorries, and air-based transport – helicopters), other roles may be filled by the same team member or by different team members according to the formulation of the plan. For the FoodDistributer role, we model two alternatives. If the actual OOTW holonic coalition is less complex then the distribution role can be handled by a simpler team that directly performs the task. For complex coalitions requiring very large movements of food, a larger distribution team maybe needed as modelled by ComplexDist below.

```
team SimpleDist extends SimpleTeam {
        #performs role FoodDistributer dist_h;

        #uses plan GetDistributer;
}

team ComplexDist extends SimpleTeam {
        #performs role FoodDistributer dist_h;

        #requires role PoliceLiaison pl;
        #requires role TechLead lead;
        #requires role FoodDispatcher disp;
        #requires role Administrator admin;

        #uses plan AcquireDistributer;
}
```

The larger food distribution team thus includes an explicit role separation for police liaison, technical leadership, dispatch of food, and administration. We continue the illustration of JACK's team-based features by suggesting a team plan for the HolonicCoalition team according to the following principles:

```
team_plan Transport_and_Distribute_Food extends TeamPlan {
        #uses team HolonicCoalition team;

        body () {
                @team_achieve(team.coord_h.ManageCoalition());
                @parallel() {
                        @team_achieve(team.trans_h[1].Unload-Food());
                        @team_achieve(team.trans_h[2].Detect-Station());
                }
                @parallel() {
                        @team_achieve(team.int_h.Contact-Local-Leader());
                        @team_achieve(team.dist_h.AcquireDistributer());
                }
        }
}
```

The reader may verify that the team plan represents an equal assignment model, with necessary jobs within the food transportation process broken down into parallel tasks. For instance, the group of people attached to the US Marines truck unit (distributor holon 1) unloads the food [Unload-Food] from the dock while concurrently the helicopter unit (distributor holon 2) selects which refugee camp this food consignment should be sent to [Detect-Station]. We note that the plan includes a declaration that enables access to the team structure. The team plan is a sequence and parallel set of actions to be performed by the team entity (in other words by the holonic coalition) with the goal of coordinating how and when these actions are to be performed by the team members. From this example, though it is far from complete, we can highlight some features of JACK's team oriented modelling approach and also point out some of its shortcomings:

- It allows for the description of team-based and autonomous agent-based activities in a clear and concise fashion.
- It enables the abstraction of what needs to be done from how it is to be accomplished, and facilitates for the team plan to be constructed without considering how the roles are to be fulfilled. This can be clearly observed by having two very different groups of holonic agents that can perform the FoodDistributer role.
- It shows how rapidly even simple team-oriented programming can become complex. Building a robust team application (in our case for holonic coalitions in OOTW) demands good software engineering practices, knowledge and computer-based tools.

We hypothesise that designing the same holonic coalition example without JACK's team-oriented programming concepts – namely developing the plans and messages for conventional autonomous agents – would easily result in a system that is very complicated and almost impossible to maintain. A change to the team's behaviour (i.e. a modification to the demand holon's requirements in a specific cooperation domain) would then impact many agency holons, and the centralised specification would be lacking. At the same time, although the above example illustrates a neat team structure within a holonic coalition, together with a realistic knowledge and activity flows, it implements an idealised view that may be difficult to realise in pragmatic military/non-government operations. For instance, as coalition management may run in parallel with the other activities, there may also be intricate control structures spanning the carious controlled activities (e.g. regular status reporting and so forth). It is not immediately clear whether the team modelling approach sufficiently enables such coordination to be captured in a natural manner. We now make some concluding remarks.

5 Concluding Remarks

There is a 21st Century demand for agile combinations of non-government and military agencies to conduct disaster relief work, peace-keeping and provide humanitarian support to people in stricken areas. These areas are often the scenes of on-going fighting and so operations other than war must be conducted in a way that protects civilian workers while not impeding battle objectives. This is a difficult balance to achieve. Owing to these requirements, application of flexible coalitions will typify how many organisations involved in war avoidance operations will have to operate. The paper has hypothesised that agent-oriented holonic behaviour could realise this next generation of decentralised knowledge-based coalition systems. We envisage that the introduction of "holonic" ideas into such OOTW coalitions will lead to a significant increase in the following characteristics:

- Robustness and stability in the face of disturbances: the system of coalesced agencies has monitoring methods to replace holons and reschedule their tasks. Also message passing is supported by resilient platforms.
- Adaptability and flexibility to rapid change: interactions between governments (as part of treaties like NATO) control the behaviour of holons for given tasks by specifying cooperation strategies as and when needed. Strategies use high-level commands and encourage holon transparency and accountability. To be scaleable, holons interact through logical spaces called cooperation domains. Holons can create, join, leave and destroy cooperation domains at run-time to satisfy the individual requirements of the crisis.
- Efficient use of available resources: holons manage their own failures and take appropriate actions to compensate for any lost effectiveness. Holons may also balance load amongst themselves to ease any strain.

(Koestler, 1967) brilliantly envisaged what such holonic behaviour should look like with holon/human societies, distributed intelligence and system construction based on every social entity being simultaneously a whole system and part of a larger structure. Translation of these abstract ideas into the technical realm of OOTW demands much research: all the problems must be identified, solved and software implemented. Only then can we provide the complete holonic solution ready for such agencies to take onboard. This paper addressed some initial ideas on how a model of holonic coalitions could be constructed. We also demonstrated how this model could be implemented using JACK, one of very few commercial-strength implementations of the BDI autonomous agent model. Though some of the concepts and coding presented here are speculative, their importance should not be overlooked. Such ideas are needed as part of a more comprehensive methodology for building and deploying coalitions.

References

- "Agent Oriented Software Ltd", (2001), http://www.agent-oriented.co.uk/.
- Bratman, M.E., Israel, D.J., and Pollack, M.E., (1988), "Plans and Resource Bounded Practical Reasoning", *Computational Intelligence*, vol 4, no. 4.
- Van Brussel, H., Valckenaers, P., Bongaerts, L. and Peters, P. (1998), "PROSA: A Reference Architecture for Holonic Manufacturing Systems", *Computers in Industry*, vol. 37, pp. 155-274.
- "FIPA – Foundation for Intelligent Physical Agents", (2001) http://www.fipa.org.
- Fletcher, M., Norrie, D.H. and Christensen, J.H. (2001), "A Foundation for Real-time Holonic Control Systems", *Journal of Applied Systems Studies*, special issue on Holonic and Multi-Agent Systems, vol. 2, no.1, Cambridge International Science Publishing.
- Hewitt, C. (1981), "Open Information Systems Semantics for DAI", *AI journal*, vol 4, no. 2.
- Howden, N., Ronnquist, R., Hodgson, A., and Lucas, A., (2001), "JACK Intelligent Agents - Summary of an Agent Infrastructure", Proceedings of the 5th ACM International Conference on Autonomous Agents, Montreal, Canada.
- Jarvis, D., Jarvis, J., Lucas, A., Ronnquist, R., and McFarlane, D.C., (2001), "Implementing a Multi Agent Systems Approach to Collaborative Autonomous Manufacturing Operations", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Tuscon, USA.
- Koestler, A. (1967), "*The Ghost in the Machine"*, Arkana, London.
- Maughan, D. (2001), "Overview of Dynamic Coalitions Programme", DARPA Information Technology Office, http://www.darpa.mil/ito/research/dc/index.html.
- McFarlane, D.C., and Gruver, W., (2001) "Workshop on Distributed and Intelligent Manufacturing", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Tuscon, USA.
- Pechoucek, M., Barta, J. and Marik, V. (2001), "Acquaintance Model Based Coalition Planning in Humanitarian Relief Operations – Progress Report", Technical report of the Gerstner Laboratory, Czech Technical University in Prague.
- Rao, A.S., and Georgeff, M.P., (1995), "BDI Agents: From Theory to Practice", Proceedings of the 1st International Conference on Multi Agent Systems, San Francisco, USA,
- Searle, J.R. (1969), "*Speech Acts*", published by Cambridge University Press.
- Shehory, O. and Kraus, S. (1998), "Methods for Task Allocation via Agent Coalition Formation", *Artificial Intelligence*, vol 101 (1-2), pp 165-200.
- Tate, A. (1999), "Proceedings of the International Workshop on Knowledge-based Planning for Coalition Forces", Edinburgh, UK.
- Thomas, J.P. (2000), "The Military Challenges of Transatlantic Coalitions", Adelphi paper 333, IISS publications, http://www.iiss.org/pub/prap333.asp.

C-CINC21: Command and Control for the Coalition Commander in the 21st Century: A Report on the Advanced Concept Technology Demonstration (ACTD)

S. Fouse, R. Delgado, and B. Beaton
ISX Corporation
760 Paseo Camarillo, Suite 401
Camarillo, California 93010
fouse@isx.com

Advanced Concept Technology Demonstrations (ACTDs) exploit mature and maturing technologies to solve important military problems. The CINC 21 ACTD program is in the third year of a three year development effort. It will be followed by a two year transition effort. The objective of this program is to develop and demonstrate information technologies that enhance senior decision making through the application of visualization tools, knowledge management tools, and enterprise services to improved command and control processes. This year the program is focused on the development of four Mission Solutions, one of which is specifically in support of Coalition Operations. The four mission solutions are: Rapid Force Employment, Consequence Management and Response, Coalition Non-Combatant Management, and Theater C4I Coordination Center (TCCC). Each of the Mission Solutions will be built using a commercial portal tool that is riding on top of a set of enterprise services including knowledge management and collaboration services. A key focus for these mission solutions is the creation and tailoring of a C2 portal that will provide the commander with an enhanced understanding of the overall situation and will provide the staff with a clear understanding of what they each need to do to support the commander's intent. In addition we have a concept of how an Information Management Officer will create and maintain the C2 Portal so that it can support the staff. At the workshop we will present the plans for two of these Mission Solutions, current status and a demonstration of one or more of the mission solutions.

A Group-Oriented Framework for Coalitions

Eric Hsu

Artificial Intelligence Center
SRI International
333 Ravenswood Ave. Room EJ211
Menlo Park, CA 94025
hsu@ai.sri.com

Abstract. Coalitions exemplify the fundamental challenge of inducing coherent group behavior from individualistic agent structures. The Collective-Agents (CA) framework rejects the distinction between individual and group deliberation, on a functional basis. Acknowledging that a group does not think using a brain, and an individual brain is not divisible into multiple minds, the CA framework nevertheless seeks an analogous correspondence between the intentional attitudes of individuals and groups alike. Resulting agents are extremely elegant, allowing hierarchical decomposition of coalitions into sub-groups and providing savings in communication costs. More importantly, such principles allow the use of abstract software wrappers for transferring advances in individual planning, control, and scheduling directly to a group setting. After formalizing the framework, we will demonstrate such claims in principle and in an implemented system.

1 Introduction

The fundamental challenge of constructing coalitions can be expressed as the transition from individual to group action. Agent architectures based on beliefs, desires, and intentions ("BDI architectures") are particularly well-developed for individual agents, but such mental constructs have defied easy translation to groups. In practice and in principle, most computer scientists and philosophers are skeptical of collective intentional attitudes, on the grounds that minds must be individual and indivisible (Bratman 1992;Grosz & Kraus 1996.) Hence, research has focused on applying individual attitudes to group content: agents must develop communicative protocols for sharing their beliefs, promoting their own desires, and cultivating intentions to perform roles in teams.

The Collective-Agents (CA) framework rejects the distinction between individual and group deliberation, on a functional basis. Certainly a group does not think using a brain, and an individual brain is not a congress of disparate voices. However, individuals and groups might employ analogous processes that operate equivalently over corresponding intentional attitudes. The CA framework seeks to establish such correspondence by using the same logical constructions to express the desires, intentions, and actions of individuals and groups alike.

As a result, CA implementations are extremely elegant, employing the same high-level data structures and algorithms for single agents as for networks of agents. This allows for hierarchical decomposition of teams into sub-teams, and recursion through multiple layers of planning and action. At the same time, communication costs can be greatly reduced. While the benefits of such a straightforward approach might be clear to the distributed or parallel computing communities, it might appear quite unprincipled to anyone who is familiar with traditional BDI architectures where individuals and groups are quite distinct.

To that end, any introduction to the Collective-Agents framework should begin by explaining the intuition behind viewing groups as individual entities and individuals as group-like constructs. The next step will be to present the framework itself, followed by a more concrete justification based on its adherence to such intuitions. After potential intuitive objections are addressed, CA can be further characterized by comparison with existing approaches. Before concluding, we will present our generalized CA implementation and evaluate it in a sample domain.

2 Background

Groups When the Dean of a university asks how much money the Computer Science department intends to budget for hardware purchases, no particular professor has the discretion to decide for everybody.[1] Traditional individualistic formalisms (Grosz and Kraus 1996) might state that a professor "intends that" the budget be set at some level. Yet if the final figure is a compromise then we cannot say that any one of the professors intended that the department thus allocate its resources. It would seem that their original preferences were more akin to desires

[1] This scenario is a combination of examples due to David Velleman and to John Searle.

than intentions; if the department's spending depends on any entity's intention, why not the department's?

To speak of a group, in and of itself, holding an intention raises the question of how such intentions could be formed, and then executed. While formal groups like faculties might obey explicit constitutional rules for transforming individual desires into courses of action, more casual situations like organizing a departmental party might be decided by very arbitrary means. Likewise, financial agreements might be so explicit as to precisely specify individual courses of action, while a mandate to hold a party might leave individuals mostly to their own devices. To make an unambitious generalization, intentional group action can be seen as a function mapping various individual desires into a cohesive aim, together with another function that apportions the various individual roles that will achieve that goal.

Individuals The Collective-Agents formalism aims for efficiency by treating individual agency as an analogous process, allowing full integration between agents and their groups. Hence, an individual agent operates as a composite of the roles it plays in various groups, characterized by processes that mediate between such roles and integrate them into a decisive course of action. For instance, being a Computer Science professor means playing the roles of instructor, administrator, and researcher, among others (Sonenberg 1994.) This means coordinating conflicting goals generated by these roles; any given afternoon might require holding office hours, attending a departmental meeting, and writing a conference paper.

More analytically, the distinction is between what Sellars calls the "plain I" and the various "I_{we}"'s which pursue various group activities[2] (Sellars 1968.) A professor who skips a conference in order to prepare a lecture might explain, "The teacher in me got the better of me." This is not to argue that human brains are divided into multiple sub-brains. Rather, CA relies on a conceptual scheme that represents the coordination of an agent's commitments by correspondence with the roles that generate such goals.

3 Toward A New Framework

Individualistic Alternatives At a crudely abstract level, most BDI architectures for individual agents resemble the following loosely-defined syntax:

Agent ::= (*Arbitrator*, *Mental-State*)

Arbitrator ::= (*Planner*, *Executor*)
Mental-State ::= {beliefs, desires, intentions, etc.}

Planner ⊂ {(*Mental-State*, *Plan*)} (a functional relation)
Executor ⊂ {(*Plan*, actions affecting the world)} (a functional relation)

Hence an agent's actions are a function of its plans, which are themselves a function of its current mental state. Several important details are left out of this sketch. Beliefs must be incurred by perceptive processes, and interact with the other mental information. The planner and executor should operate concurrently; changes in the agent's mental state can affect its plan, and hence its course of action. However an architecture addresses such complexities, though, it does so using these operational modules, or equivalent models.

Many multi-agent systems are reluctant to depart from this framework, and use the same structures to implement group activity as depicted in Figure 1. In the figure, "Group 1" does not exist as a computational entity, so much as it results implicitly from the communication between Agents A and B. In order to deliberate concerning their group as a whole (for instance, to decide whether to serve as a sub-unit in a larger group,) the agents must explicitly refer to "Group 1" in the contents of their communications.

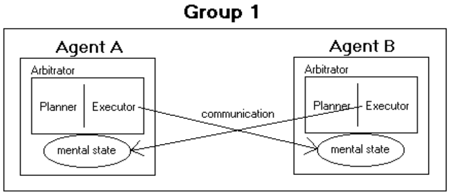


Figure 1: A Two-Member Group of Individualistic Agents.

[2] For instance, a professor might harbor an "I_{we}" which participates in such declarations as "We the faculty intend to alter the budget."

Figure 2 illustrates two forms of complexity that arise from such amorphousness. First, the addition of a third agent multiplies the number of communicative channels. In general, the number of possible pairings increases quadratically with each additional agent, endangering communication bandwidth. Secondly, Groups 2a and 2b must be specifically disambiguated within messages since the individualistic agent architecture does not automatically capture hierarchical group structures. That is, there is no architectural difference between 2a, a two-entity group whose first member is itself a two-entity group, and 2b, which is a three-entity group. At an implementational level, such generality should not be mistaken for flexibility. For instance, for both groups to function simultaneously, Agent A must index its correspondence with B and C by relevant group for lack of inherent context.

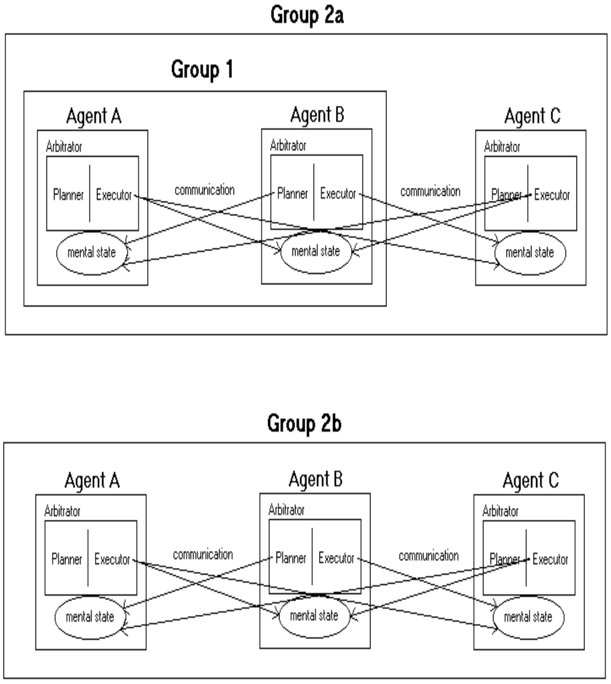


Figure 2: Individualistic Groups of Greater Complexity.

The Collective-Agents Framework The observations made in Section 2 suggest that group-orientation can relieve the above problems, and provide some additional benefits. The following syntax outlines the basic structure of Collective-Agents (the '+' symbol designates "one or more"):

Collective-Agent ::= *Individual* | *Group*
Individual ::= (*Arbitrator*, *Individual-Role*+)
Group ::= (*Arbitrator*, *Collective-Agent*+)

Arbitrator ::= (*Planner*, *Executor*)
Constituent ::= *Collective-Agent* | *Individual-Role*
Individual-Role ::= one of the roles which an individual plays

Planner ⊂ {(*{Report+}*, *Plan*)} (a functional relation)
Executor ⊂ {(*Plan*, *Instruction*+)} (a functional relation)

Report ::= (*Constituent*, {beliefs, desires, intentions, etc.})
Instruction ::= (*Constituent*, actions affecting the world)

Hence the same arbitrating structure governs groups and individuals alike, allowing hierarchical decomposition of teams, and centralizing deliberative processes for team-members. Figure 3 presents the same two group structures from Figure 2 as represented using the Collective-Agents framework. C-Agent 2a is a two-member entity, whose first member is a team composed of C-Agents A and B. Within this structure, neither A nor B communicates directly with 2a; the arbitrator for C-Agent 1 acts as an intermediary instead. C-Agent C, which is not involved in the activities of C-Agent 1, only communicates with A and B concerning the affairs of C-Agent 2a, thus speaking solely through the arbitrator of that group.

Unlike other "collective" agent architectures, this one does not distinguish between individual or collective arbitrators, meaning that they function identically.

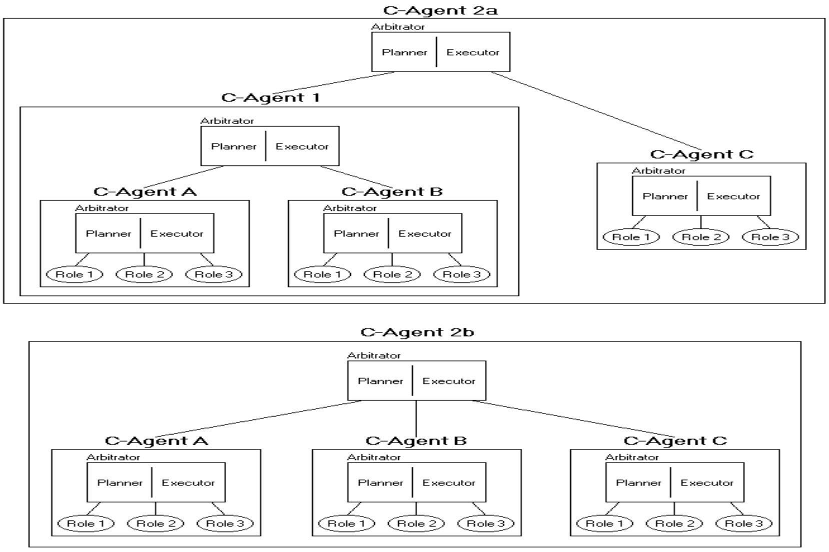


Figure 3: Two Complex Collective-Agents

4 Benefits

Communicative Efficiency By imposing such rigid structure upon agent interaction, the CA framework can improve efficiency for a large class of group topologies. Table One summarizes some easily observed characteristics resulting from a tree-structured, CA-based coalition structure.

	CA Best Case	CA Worst Case	Individu-alistic
Arbitrators	*O*(n)	*O*(n)	*O*(n)
Channels	*O*(n)	*O*(n)	$O(n^2)$
Hops	*O*(1)	*O*(n)	*O*(1)

Table 1: Comparison of Computational Complexity for Groups Assembled from n Individuals.

These observations measure three types of cost. The first is the number of arbitrator processes that would have to be run for a group of n agents. While any arrangement of n individualistic agents would merit n arbitrators, the number varies for Collective-Agents based upon their structure. In the best case, the n individuals would simply comprise a single group, and n+1 arbitrators would be necessary—one for each individual and one for the entire group. The worst case is based on the limitation that a group must contain at least two collective-agents; otherwise a single individual could spawn an arbitrary number of arbitrators if it were nested deeply enough as a subgroup of a subgroup of a subgroup, and so on. Accordingly, the worst-case structure is a balanced binary tree of Collective-Agents with the n individuals as leaves. Even so the number of arbitrators would be just 2n-1.

The second measure is the number of communicative channels that must be available between pairs of arbitrators. While this might not be significant in laboratory simulations, an agent deployed in practical applications cannot always communicate with every other agent without a cost, if at all. Just as the Internet's hierarchy of gateways, routers, and backbones counters the impossibility a running cable between every pair of computers in the world, CA hierarchies enable efficient communication between individuals. There are (n^2-n) / 2 potential pairs within n agents. For each individualistic agent within a group to be able to broadcast to all its partners, a channel must exist for each pair. On the other hand, a CA individual within the best-case structure already described would need only report to the group arbitrator, which would then pass the relevant information back down to the other individuals. This would require n channels, while the worst-case scenario (also a binary tree) would rely on just 2n-2 channels.

The disadvantages of such parsimony affect the third area. Just as IP packets must make a number of hops across the Internet before arriving, Collective-Agent communications might pass through a number of managing arbitrators before reaching interested parties. Given the previous assumption that CA groups must contain at least two members, the maximum number of hops occurs in the deepest possible binary tree with n individuals as leaves. In such a case information from agents in the most deeply nested group must traverse n channels before reaching the top-level group and its individual member. However, in best-case hierarchies, which consist once again of single groups, any message must make at most two hops. While ubiquitous channels between individualistic agents provide one-hop connections, the complexity introduced by CA is at worst linear, and constant at best.

Abstraction Beyond such metrics, the hierarchical decomposition employed by C-Agents can provide additional implementational advantages akin to those offered by functional decomposition in programming languages. Since Arbitrators can govern individual roles, or other C-Agents alike, they can be implemented using the same computational structures. "Belief-Desire-Intention" architectures are well developed for individual agents, and can be applied directly to C-Agent groups once communication between agents is encapsulated as mental events within the collective whole.

For instance, when two CA partners report conflicting desires during negotiation, their arbitrator can reconcile this difference in the same way that traditional individuals would make decisions based on conflicting goals. Where before the two desires would reach the planning module directly from an agent's mental repository, now they would arrive as communications from two constituent C-Agents. While an individual executor would directly instigate agent actions, now a CA arbitrator sends instructions to its subordinates.

This is not just an implementational shortcut, but a new way to formally conceptualize and then implement at worst the same patterns of agent coordination. If two traditional collaborators devote some percentage of their processing and messaging to negotiate a plan for their shared activities, only conceptual prejudice prevents us from extricating such activity into a new process. If two conventional "collective" agents seek to centralize their negotiations through an intermediary server, why should this server's operation deviate from existing well-developed individual planning methodologies?

Because C-Agents can be generated dynamically from heterogeneous classes of arbitration schemes, this framework is not a design document for agent interaction, but for a wrapper system to generate such dynamics from individual architectures. This point is best understood through the implementation and its use in a sample domain. sample domain.

5 Implementation

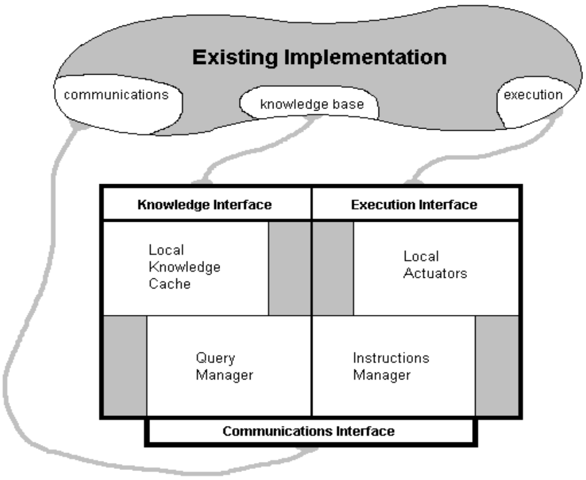


Figure 4: C-Agents Wrapper Design.

The C-Kit The generalized C-Agents implementation (named "C-Kit") is a domain-independent control system that operates in a self-contained thread. In addition, it is installed independently from whatever existing planning and execution system a researcher is using, connecting in only three places through a highly abstract interface. In short, it constitutes a software wrapper for turning any BDI agent into a C-Agents arbitrator that can interact with other, potentially different, agent implementations that have also been outfitted with the C-Kit.

Figure 4 illustrates the operation of such a composite agent. Here a generic agent process has been outfitted to serve as the arbitrator of a group of C-Agents. Mirroring the formal pairing of deliberator and executor, the C-Kit provides two principle services to the pre-existing implementation: the Knowledge Interface and the Execution Interface. Where an individual planning agent formerly consulted its own knowledge base through the course of deliberation, it now consults the C-Kit's knowledge base, which compiles individual beliefs from the constituents. A query first passes through the local knowledge cache to see if it can be answered by already compiled information. If not, it goes through the query manager and is communicated to subordinates. Likewise, the query manager is also responsible for responding to queries from superiors, potentially passing such requests on to its constituents. The specific methods for resolving conflicting reports or deciding what information to cache are determined by the domain and not specified by the C-Kit. In the test domain presented below we coded simple voting procedures and heuristics, but the C-Kit should be seen as a workbench for encoding more advanced methods.

Similarly, a C-Agent's actions are actually executed by actuators controlled by its constituents. Hence, when the group wishes to perform a certain series of actions, the commands actually go through the Execution Interface, which first checks to see if any can be performed by local actuators. The Instructions Manager dispatches any remaining orders to constituents. In the other direction, it receives instructions from superiors and either executes them locally or dispatches them to constituents. Representing a group entity's capabilities as a compilation of constituent capabilities and their possible interactions is again left to the domain expert. Where before such models were installed into each necessarily identical individual agent or emerged implicitly from hard-wired behavior, they can now be deployed once within the C-Kit and used with heterogeneous collections of agent implementations.

The third and final interface, depicted at the bottom of the figure, governs the messaging activities produced by the other two. Specifically, the Communications Interface sends queries and instructions to constituents, and replies to queries from superiors. Similarly, it queues query responses from constituents, and queries or instructions received from superiors. To do so, it interfaces with whatever communicative facility the existing agent implementation already uses. If there are none, then certainly one must be created in order to do multi-agent planning, and then it should be connected directly to the C-Kit through this interface.

The toolkit is implemented using the Java language "Remote Methods Interface" package, which allows an agent to perform operations on a remote host without relying on high-level message passing. This way, if an agent is already deployed it can interact with a C-Kit running on a different host almost transparently. In Figure 4 it is the gray connectors that signify this link. Later, including RMI in the design should allow for mobile code, whereby arbitrators will be able to transfer themselves to new hosts in search of extra processor time or faster access to local actuators. However, such functionality has yet to be implemented.

Before continuing a final observation should be made concerning the domain-specific components of the knowledge and action interfaces. In particular, the compilation of parent-agent beliefs and capabilities is left open in this particular implementation, but related research efforts toward similarly aligned goals may provide generalized procedures for doing so. Specifically, the area of structured theorem proving seeks to perform inference over multiple knowledge bases, while one main area of Semantic Web research is automated service compilation from multiple sources. The author is most familiar with work at the Stanford Knowledge Systems Laboratory (Amir & McIlraith 2001, McIlraith et al 2001).

6 Example Scenario

Suppose that in some future military scenario, two autonomous ground vehicles G1 and G2 patrol for enemies within a sensitive area, while a pair of autonomous air vehicles A1 and A2 patrol the perimeter. If either party detects an enemy, the first priority is for the detecting team to pursue it immediately, the second is to maintain the inner patrol, and the least important is to continue the perimeter patrol. Also, suppose that a single air vehicle can patrol an area or pursue a target, but such tasks require two ground vehicles working together.

Based on these mission criteria, the desired outcome depends on whether the target appears before the ground or air vehicles. If it appears within the area, both ground vehicles should immediately pursue the target, with an air vehicle taking over the inner patrol. On the other hand, if it appears on the perimeter, only one air vehicle should pursue, as the other continues the perimeter patrol and the ground vehicles continue their inner patrol.

In an individualistic agent system, such conditional behavior would have to be distributed across each agent. The initial patrol plan must specify each agent's new role given each contingency, or alternatively the agents must re-negotiate their tasks upon detecting the target. Both approaches elicit broad interest within the agent systems community (with Ortiz, Hsu et al. 2001 and Ortiz & Hsu 2002 representing each approach within this same type of scenario.) On the other hand, the CA framework would encapsulate such group-level decisions within new computational structures.

Specifically, each agent can be wrapped using the C-Kit and connected to one of two independent C-Agents representing the patrol teams, P1 and P2. Further, the two teams would in turn connect via a third new C-Agent C1 representing the overall coalition. It is easy to see that if these vehicles and their mission were part of an even larger mission, then C1 could connect to an even higher agent. Each of the non-leaf C-Agent consists of an unmodified individual agent implementation outfitted with the C-Kit so that it operates as an individual but serves as a group.

Thus, it is C1 that receives a reported detection over the chain of communications, and dispatches a pursuit instruction to the nearer of P1 and P2, while assigning the inner patrol task to the other. It is important to stress that the individual agent operating C1 does not "realize" that it is running a group. It thinks it has two sets of actuators (P1 and P2) and is using them both to patrol. The reported detection arrives in its knowledge base the same as if C1 had sensed it directly, and its subsequent instructions are translated by the C-Kit from calls to its supposed actuators.

One consequence is that C1 does not specify that the air vehicles should split up when they are closer to the target. Rather, the C-Agent representing their partnership (P2, without loss of generality), receives the instruction to pursue the target, and dispatches it to either A1 or A2. Again, the individual agent implementation has been initialized with two sets of actuators, by virtue of the C-Kit. At first P2 uses both for its single task of patrolling, and patrol instructions are dispatched to the C-Agents representing A1 and A2. On receiving the new directive, it confirms that either virtual actuator can pursue either directive on its own, and the tasks are split. If, on the other hand, P1 receives the same instruction to pursue, it reports that it cannot because patrolling requires both ground vehicles. C instructs P1 to pursue, which has higher priority, and registers P1's inability to patrol upon assuming the pursuit. Thus it tries to achieve the patrol task by calling on its other actuator, P2, with results analogous to the previous case. (In practice, the implementation shortcuts some of this interaction by tagging each initial instruction with a priority.)

Finally, the individual agents receive their instructions through P1 or P2. When they attempt to perform them, the C-Kit Execution Interface dispatches them via what turn out to be actual local actuators. Should they encounter a reportable event such as success, failure, or their arrival at some specified state, they report it to P1 or P2. There the C-Kit Knowledge Interface deposits the report in P1 or P2's knowledge base as though it was its own activities that provoked it.

In many cases such interactions are the same ones exhibited by individualistic systems. When a new task arrives, an elected or otherwise designated agent leader might query constituents for action capabilities before reaching a decision, just as C1 compiles its capabilities from P1 and P2's. That the individual agents might communicate amongst themselves to make a decision in unison is not exclusive to individual architectures. C1 may very well consult its knowledge base concerning P1s and P2's individual utilities for a given course of action, its query passing through the C-Kit Knowledge Interface and finally the agents themselves via the communications interface. Is this any worse than individual agents spawning some centralized process to compile their votes and issue the outcome? If the two approaches are functionally equivalent then CA may be operationally superior, making such centralization explicit and running it within a well-understood single-agent process. The alternative should be treated as an interesting, younger, and open research area in the best cases and an inefficient ad-hoc solution in the worst.

A second observation is that none of the C-Agents, at any level, need be autonomous. Not only can the individual agents within follow advisable architectures, they could even consist entirely of human elements. That is to say, the C-Kit can be viewed as a sort of ``command console" where human operators receive reports from constituents, arbitrate them with goals, and issue queries or commands. The usefulness of such approaches has already been successfully illustrated without the use of the C-Kit, and in fact helped inform its design (Myers & Morley 2001).

7 Results

Sample Domain During development the C-Kit has been continuously tested within an autonomous robots domain. In typical scenarios, the agents in question must form and then execute group plans for patrolling arbitrarily defined areas and pursuing any targets they might find, based on their varying capabilities. Though this controller is being developed to eventually run on hardware, at this point experimentation occurs in the simulator depicted below.

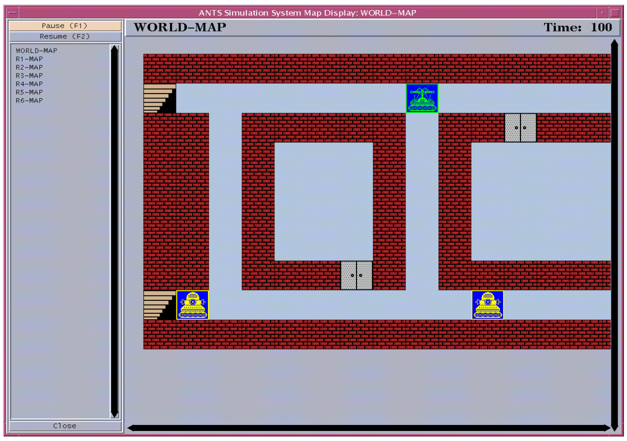
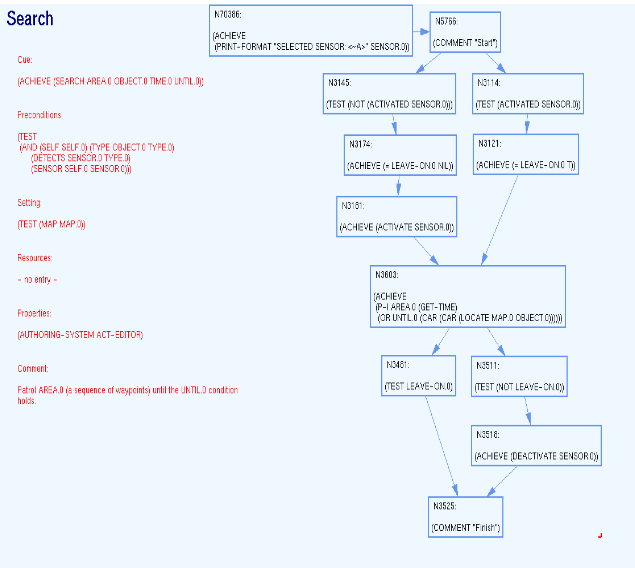


Figure 5: Software Domain Testbed.

Each robot has its own processor and memory, so in the experiments each agent runs on its own host, connected to all the others via a local-area network. Each C-Agent consists of an existing planning and control implementation coupled with a C-Kit encoded with domain knowledge in the form of plan templates, beliefs about each robot's capabilities, and rules for compiling information. During execution, the agents form a coalition in response to high-level directives to perform group tasks. Whenever this occurs, the system spawns a new C-Agent co-hosted with an arbitrarily selected team member.

The "PRS" Procedural Reasoning System, freely available from SRI International, provides the "existing" implementation for the C-Agents. It responds to mission orders or new perceptual information via the C-Kit's Knowledge Interface, and sends queries in the opposite direction. It then performs planning and scheduling, as well as execution monitoring, outputting requests for action to the C-Kit Execution Interface. Figure 6 depicts an example search plan represented in PRS's graphical control language. More detailed information on the PRS system can be found at http://www.ai.sri.com/~prs.



Search

Cue:

(ACHIEVE (SEARCH AREA.0 OBJECT.0 TIME.0 UNTIL.0))

Preconditions:

```
(TEST
 (AND (SELF SELF.0) (TYPE OBJECT.0 TYPE.0)
      (DETECTS SENSOR.0 TYPE.0)
      (SENSOR SELF.0 SENSOR.0)))
```

Setting:

(TEST (MAP MAP.0))

Resources:

- no entry -

Properties:

(AUTHORING-SYSTEM ACT-EDITOR)

Comment:

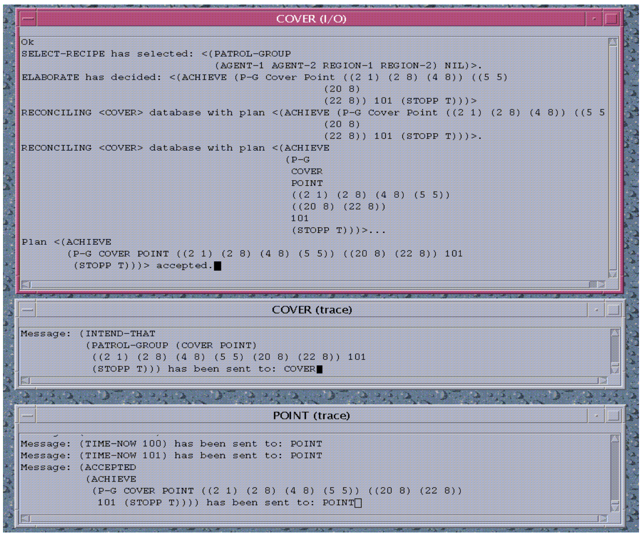
Patrol AREA.0 (a sequence of waypoints) until the UNTIL.0 condition holds.

N70386:
(ACHIEVE
(PRINT-FORMAT "SELECTED SENSOR: <~A>" SENSOR.0))
N5766:
(COMMENT "Start")
N3145:
(TEST (NOT (ACTIVATED SENSOR.0)))
N3114:
(TEST (ACTIVATED SENSOR.0))
N3174:
(ACHIEVE (= LEAVE-ON.0 NIL))
N3121:
(ACHIEVE (= LEAVE-ON.0 T))
N3181:
(ACHIEVE (ACTIVATE SENSOR.0))
N3603:
(ACHIEVE
(P-I AREA.0 (GET-TIME)
(OR UNTIL.0 (CAR (CAR (LOCATE MAP.0 OBJECT.0))))))
N3481:
(TEST LEAVE-ON.0)
N3511:
(TEST (NOT LEAVE-ON.0))
N3518:
(ACHIEVE (DEACTIVATE SENSOR.0))
N3525:
(COMMENT "Finish")

Figure 6: PRS Search Procedure

The completed system ran over 100 randomly generated scenarios in the sample domain, each requiring between two and five individual robots. The typical experiment called for three initial tasks for various combinations of agents to execute and plan concurrently. Then, during execution, two additional tasks would arrive, either as new mission directives from outside the system or via unpredictable events in the environment. To be more specific about the latter type, some initial tasks included instructions to initiate a new one should a certain condition come to pass. For instance, a group of agents might initially be instructed to patrol a particular area and then pursue any targets detected during the patrol.

Figure 7 presents some of the output generated during one such run. Here a higher-level C-Agent consists of two individual agents named "Cover" and "Point," and resides on the same host as Cover.



```
Ok
SELECT-RECIPE has selected: <(PATROL-GROUP
                              (AGENT-1 AGENT-2 REGION-1 REGION-2) NIL)>.
ELABORATE has decided: <(ACHIEVE (P-G Cover Point ((2 1) (2 8) (4 8)) ((5 5)
                                                   (20 8)
                                                   (22 8)) 101 (STOPP T)))>
RECONCILING <COVER> database with plan <(ACHIEVE (P-G Cover Point ((2 1) (2 8) (4 8)) ((5 5
                                                   (20 8)
                                                   (22 8)) 101 (STOPP T)))>.
RECONCILING <COVER> database with plan <(ACHIEVE
                                             (P-G
                                              COVER
                                              POINT
                                              ((2 1) (2 8) (4 8) (5 5))
                                              ((20 8) (22 8))
                                              101
                                              (STOPP T)))>...
Plan <(ACHIEVE
       (P-G COVER POINT ((2 1) (2 8) (4 8) (5 5)) ((20 8) (22 8)) 101
        (STOPP T)))> accepted.
```

```
Message: (INTEND-THAT
          (PATROL-GROUP (COVER POINT)
           ((2 1) (2 8) (4 8) (5 5) (20 8) (22 8)) 101
           (STOPP T))) has been sent to: COVER
```

```
Message: (TIME-NOW 100) has been sent to: POINT
Message: (TIME-NOW 101) has been sent to: POINT
Message: (ACCEPTED
          (ACHIEVE
           (P-G COVER POINT ((2 1) (2 8) (4 8) (5 5)) ((20 8) (22 8))
            101 (STOPP T)))) has been sent to: POINT
```

Figure 7: Example Execution Trace.

The top panel depicts the C-Agent superior generating of a group plan for patrolling a particular area, and the bottom two shows its individual instructions for action being sent to the two constituents.

Ignoring improved developmental efficiency, the test system should be evaluated for actual performance, which is here measured in terms of messages and execution speed. In the system, collective agents are allowed to deliberate for as long as it takes to receive responses to all their queries, so plan optimality is determined solely by the planning system operating separately in PRS.

For purposes of comparison, the scenarios were also run over nearly identical agents that were previously developed without the C-Kit. Such "individual" agents used the same straightforward knowledge compilation procedures, plan templates, and domain models as the collective ones, so they always arrived at and executed the same plans. Hence, the only difference was the organization of their communications and processing, thus isolating the characteristic consequences of the CA framework.

Table 2 compiles the results of such experiments. Each row specifies the performance of one of the systems over a certain class of scenarios, in terms of the total number of messages sent and the average amount of time elapsing between the arrival of a new task and its execution. In general, the CA system executed approximately 15% faster using almost 17% fewer messages when compared to the individualistic system over the course of all scenarios.

The most significant source of message conservation (and hence, execution speed) involved situations where the group arbitrators happened to reside on hosts that were centrally located on the network topology. Thus, they required few hops in to reach the constituents. In contrast, the individualistic agents sometimes had to send their messages across the entire network in order to coordinate their planning activities. This naturally suggests that high-level arbitrators should reside at network hubs.

	Messages	Time (ms)
C-Agents, All Scenarios	8,014	1.946
Individualistic, All Scenarios	9,580	2.285
C-Agents, Event-Driven only	4,586	2.140
Individualistic, Event-Driven only	4,736	2.221
C-Agents, Goal-Driven only	3,428	1.752
Individualistic, Goal-Driven only	4,844	2.344

Table 2: Comparison of Agent Types.

The data also illustrates that unfortunately, most gains arose within a specific class of scenarios, specifically those denoted as goal-driven scenarios. On the other hand, the two systems had nearly identical performance in scenarios that included event-driven goals.

Such directives require specific actions to take place once a particular condition holds, for instance "Stay still unless you detect a target" or "Patrol until fuel runs low." The problem for the C-Agents is that PRS performs such directives by checking the knowledge base each execution cycle in order to determine whether the condition has come true. With the individual agents, each agent performs such checks locally, and reports to the group only when new perceptual information affirms the targeted condition. On the other hand, when a C-Agent representing a group of robots pursues the same goal, the continuous stream of queries generated by its PRS cycle is dispatched as messages to constituents, through the Knowledge Interface.

Such goals only accounted for one fifth of the tasks in the category designated "Event-Drive" in the table. Otherwise, this problem would outweigh the other advantages enjoyed by the C-Agents and their performance would be far worse than comparable. This suggests that the formalism should be extended to allow notification hooks in constituents so that they can inform their superior arbitrators when a reportable event occurs. This would alleviate the need for constant querying and make the implemented system's behavior in such scenarios comparable to that of the individualistic agents.

Another complication deliberately excluded from the above examples is that sub-teams are not always disjoint from each other, and hence CA hierarchies need not always be trees. Referring once again to the example scenario, suppose that for some new task, each air vehicle must now work together with one of the ground vehicles. If this means that the patrols involving like vehicles have terminated, then P1 and P2 can be killed off and replaced by two new C-Agents representing the new working groups. If, however, both groups are to be maintained at once, then there will now be four upper-level C-Agents. This is not a problem for the system, as the formalism allows leaf C-Agents to work on multiple tasks, with contention to be resolved somewhere up the hierarchy where they have a mutual parent. That is, if the new patrols were under the control of a new coalition C2 that used some of C1's resources, a higher agent with C1 and C2 as children would resolve any contention.

The logistical problems caused by such complexity are not specific to CA, as individual agents would have to divide their attention and index their communications in accordance with the denser structure. However, the combinatorial blow-up arising as each possible grouping of agents comes into play is limited to increased processor time for individual agents. For C-Agents, though, there are memory considerations as well because a new C-Agent must begin operation for each new working group. Even if few domains lack enough natural structure to limit interaction hierarchically, the fact remains that C-Agents can run our of space in such applications.

Such results suggest that the Collective-Agents Framework is particularly well suited for large, widely distributed coalitions of operationally isolated teams. As individuals become increasingly numerous and far-flung, the conservation of communicative channels becomes increasingly important. At the same time, the number of extra arbitrators can never exceed the number of working groups; processing requirements are at most doubled in hierarchical cases. Further, agents working in distantly related groups produce more balanced tree structures, in turn limiting the number of extra hops between agents. Finally, developing such large networks exploits the uniform organizational structure of CA. Arbitrating processes can modularized and replicated across computational structures, and individuals need only report information and follow instructions.

On the other hand, smaller, more cohesive groups can perform better using individualistic methods. If every agent will at some point need to communicate with every other agent, and the group is compact enough that there are plenty of channels available, then CA loses its communicative advantage over traditional approaches and begins to exhibit space problems. Further, as agents take on memberships in multiple groups, such problems are exacerbated. Hence, as an example, C-Agents might perform better as clearly delineated military entities than as specialized task forces. It should be observed that C-Agents can be used for the upper reaches of a hierarchy, with leaf C-Agents interfacing with alternative individualistic systems of agents rather than physical individuals.

8 Comparison

Most existing formalisms for multi-agent systems take an individualistic stance toward collaboration (Sonenberg et al. 1994; Cohen and Levesque 1990; Jennings 1995; Tambe 1997; Shoham 1993.) Somewhat surprisingly, we believe that the Collective-Agents framework is compatible with such approaches on two grounds. First, it is an implementational framework as well as a theoretical formalism. Secondly, it is nevertheless a philosophically principled move; it does not provide a software-based "hack" to approximate real-world phenomena.

In presenting their individualistic SharedPlans formalism, the authors observe that their logical constructions are not meant to be fed through a theorem-prover or serve as a software design document (Grosz and Kraus 1996.) Rather, formalisms specify group processes at a high level, serving to inform implementations that might take on very different forms. The CA framework presents a particular method for organizing agent implementations so that such high-level processes can be specified in an efficient, group-oriented manner. Whether individualistic agents adopting a new task elect a leader who polls them for conflicts with existing plans, or a high-level C-Agent arbitrator consults its constituents' databases via the Knowledge Interface, the patterns of messages traveling across the network are remarkably similar. At worst this is a new framework for specifying such patterns.

This is not to say that the motivations behind CA do not match its operational semantics. While individualism has long reigned in philosophical studies of intentional attitudes (Bratman, 1992; Searle 1990), a new movement has begun to explore collective mental phenomena as basic entities (Baier 1997; Gilbert 1996; Stoutland 1997.) Indeed, the creation of new arbitrators for group activity is much like Baier's mental commons. Such constructions need not exist physically to serve as useful conceptual schemes. For AI to be a valid enterprise, an entity's operation cannot be bound to our conceptual labels for its activities. Otherwise a computer cannot even add; it is merely moving electrical current through registers.

Ongoing work currently focuses on more efficient methods for event-driven behavior and on more intelligent protocols for hosting newly generated arbitrator processes. Another task is to try using the C-Kit with a more sophisticated planning system, such as SRI's SIPE family of planners built on PRS.

9 References

Amir, E. and McIlraith, S. (2000) "Partition-Based Logical Reasoning", Proceedings KR2000, pages 289-400.

Baier, A. (1997) "Doing Things with Others: The Mental Commons", In Alanen, Heinämaa, and Wallgren, editors, Commonality and Particularity in Ethics. St. Martin's Press, New York, pages 15-44.

Bratman, M. (1992) "Shared Cooperative Activity", Philosophical Review, 101(2):327-41.

Cohen, P. and Levesque, H. (1990) "On Acting Together", In Proceedings of AAAI-90:94-9.

Gilbert, M. (1996) "Walking Together: A Paradigmatic Social Phenomenon", Living Together: Rationality, Sociality, and Obligation. Rowman and Littlefield, New York.

Grosz, B. and Kraus, S. (1996) "Collaborative Plans for Complex Group Action", Artificial Intelligence, 86(1):269-357.

Jennings, N. (1995) "Controlling Cooperative Problem Solving in Industrial Multi-agent Systems Using Joint Intentions", Artificial Intelligence, 75(1):195-240.

McIlraith, S., Son, T.C. and Zeng, H. (2001) "Semantic Web Services", IEEE Intelligent Systems, Special Issue on the Semantic Web, 16(2);46-53.

Ortiz, C. L. and Hsu, E. I. (2002) "Structured Negotiation", To appear, Autonomous Agents and Multi-Agent Systems.

Ortiz, C. L., Hsu, E. I., DesJardins, M., et al. (2001) "Incremental Negotiation and Coalition Formation for Resource

Bounded Agents", Proceedings of the AAAI Symposium on Negotiation.

Searle, J. (1990) "Collective Intentions and Actions", In Cohen, Morgan, and Pollack, editors, Intentions in Communication. The MIT Press, Cambridge, MA, pages 15-32.

Sellars, W. (1968) "Science and Metaphysics: Variations on Kantian Themes", The Humanities Press, New York.

Shoham, Y. (1993) "Agent-Oriented Programming", Artificial Intelligence, 60(1):51-92.

Sonenberg, E. et al. (1994) "Planned Team Activity", In Castelfranchi and Werner, editors, Artificial Social Systems, Lecture Notes in Artificial Intelligence. Springer Verlag, Amsterdam.

Stoutland, F. (1997) "Why are Philosophers of Action so Anti-Social?", In Alanen, Heinämaa, and Wallgren, editors, Commonality and Particularity in Ethics. St. Martin's Press, New York, pages 45-74.

Tambe, M. (1997) "Agent Architectures for Flexible, Practical Teamwork", In Proceedings of AAAI-97: 22-8.

Model Predictive Risk Control of Military Operations

Jan Jelinek

Honeywell Laboratories,
3660 Technology Drive
Minneapolis, MN 55418
U.S.A.
jelinek_jan@htc.honeywell.com

Abstract. Work reported in this paper was done as part of the DARPA Joint Force Air Component Commander (JFACC) project. Its objective was to investigate the possibility of improving the stability and agility of military operations using the concepts of modern control and game theories within the framework of state-of-the-art computer science and operations research.

Military operations are always defined and executed within the context of a command and control (C2) hierarchy. The questions we have studied at the Task level were twofold. What is the minimal effective task force size and composition for a given task, and how to guarantee successful task execution in the presence of uncertainties in combat, both due to random effects of weapons and an intelligent adversary? At the Task Group level, we asked how to optimally allocate and schedule available resources to satisfy the force size requirements for as many concurrent tasks as possible. We have developed probabilistic Markov models of combat dynamics, and then used them to build the Model Predictive Task Commander and Model Predictive Resource Allocator systems, which are briefly described in the paper along with experimental results showing their performance in simulated battles.

1 Introduction

War or, on a smaller scale, any battle can be viewed as a trade in which opponents mutually “trade” their resources until one side is forced to declare bankruptcy due to its bad trading decisions. The traditional mathematical approach to optimize this process is to view it as a resource allocation problem, whose objective is to match resources to targets in the most “profitable” way as defined by a suitable criterion. Obviously, the fundamental question is how to properly value resources to be traded. Dollars spent to procure them do not make much sense in war since their military values derive largely from expected opportunity costs and not from the numbers listed in accountants’ ledgers. The military value of a bomber is surely different at the outbreak of war than on the V day and on any day in between, and the difference depends strongly on the war strategy and many other factors.

In any model of war based on the notion of trade the issue of resource valuation cannot be eventually avoided. However, the responsibility for resource valuation should be assigned to those who can be expected to possess knowledge and information relevant for such decisions, which, in practice, means the higher rungs of the command and control (C2) hierarchy. If the resource value derives mostly from the opportunity costs, then it is neither fair nor reasonable to ask a field commander to decide, say, how many of his bombers are worth a given heavily defended enemy air base he was ordered to destroy. Yet he somehow has to compose his strike packages, devise their offensive and defensive capabilities, schedule their employment and then manage the battle to success once it gets underway.

In the project we are reporting on here, our objective was to investigate the potential of improving the stability and agility of military operations. We have focused mostly on the lower rungs of the C2 hierarchy. For the reasons outlined above, we have rejected the more traditional design concepts based on straightforward resource allocation. Instead, we proceed in two steps. For every task we first try to establish what it takes to get it done regardless of whether the resources are actually available. We call the answer the *minimal effective force* (MEF). Only then we attempt to allocate available resources to the set of given tasks in amounts sufficient for their successful completion. As it turns out, most tasks can be successfully completed in more than one way so that their MEF is actually a set of alternative solutions. This extra degree of freedom provides the planner with greater flexibility to reconcile competing demands. Moreover, the breakup of planning into the two phases improves the transparency of the planning and battle management algorithms, significantly simplifies their computer implementation and speeds up their execution. Below we explain the gist of our approach, show some experimental results and outline issues for further work.

2 Concept of operations

The technology underpinning our approach cannot be judged properly without some notion of concept of operations. The concept emerging from our current work is as follows.

Superficially, the proposed C2 structure remains hierarchical. In its functionality, however, there are substantial differences in the way information flows about the hierarchy. The traditionally strong top-down information flow is complemented by a comparably strong bottom-up flow, producing a structure, in which decisions are reached through a negotiation process, whose objective is to optimally match the war objectives at the top with the resources and operational constraints existing in the field at any given moment. Conceptually, we are trying to move away from directive- to more consensus-based command and control, in which different levels of the hierarchy are seen more like peers united by a shared interest in winning the war rather than disinterested subordinates asked to fulfill orders passed down to them.

The hierarchy, its levels and the kinds of information passed between them are shown in Figure 1. The Mission Execution Level has a place in our hierarchy, even though we have not addressed it in our work. We have focused on investigating the Task, Task Group and, to some degree, Operations Levels. We have not studied levels above the Operations Level.

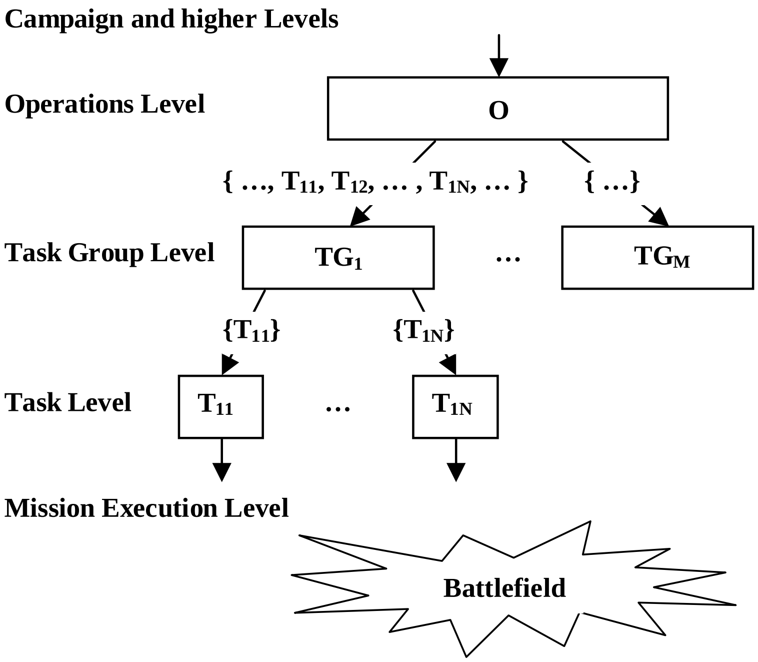


Figure 1. The Command and Control hierarchy

The basic notion in our approach is that of the *task*. Task is a relatively simple activity, whose objective is already stated in military terms (i.e., not strategic or political). It has a deadline, by which it has to be completed, and the issuer specifies the urgency of its successful completion by providing the desired probability of success. He also specifies what is the acceptable own loss to fulfill the task. This formulation reflects a realistic view of combat as always involving a significant random component, which is much better to be dealt with in the open than to pretend that it does not exist. The *battle* is then the process of executing a task. In this sense, task and battle are more or less synonymous words, one stressing more the objectives, the other the process of achieving them. Battles are generally viewed as sequences of simpler combat activities. The reader can interpret them as missions, sorties, etc.

The Task Level commander's role is in designing appropriate minimal effective forces (e.g., strike packages) for each mission of the given task and then making the necessary corrections to them prior to the next mission depending on the *battle damage assessment* (BDA) feedback he receives. Executing the missions is the mission commander’s responsibility and as we have pointed out above, is not the subject of our interest.

Any operation will likely consist of a large number of tasks running concurrently, with overlapping demands on resources. The Task Group Level commander, who is generally seen as the “owner” of resources, is responsible for apportioning his resources to the individual tasks that were assigned to his group for execution.

The Operations Level job is to translate the broader, strategic objectives (e.g., destroy enemy air base Alpha) into an interconnected web of individual tasks, which are then passed down to the Task Group Level commander for execution. The interface between the two levels is a queue of tasks $\{ \ldots, T_{11}, T_{12}, \ldots \}$, which is continuously filled by the former on one end and emptied by the latter on the other.

The reader may have noticed that in our concept of operations there is no notion of target and hence of target value. To be sure, targets are discussed, but only internally at the Operations Level, when tasks are being defined. In our C2 philosophy, the notions of targets and their values are not used in communication between levels below the Operations Level, because commanders working there lack the knowledge context to properly understand them.

A task exists since the moment the Operations Level pushes its statement into the task queue, well before any resources have been allocated to it. Indeed, once the Task Group commander retrieves it from the queue, his first job is not to allocate his resources to it, but to find out what is the minimal effective force to successfully prosecute it. To do so, he assigns the task a Task commander, effectively passing it down to the Task Level. This officer calculates the Minimal Effective Force needed to fulfill it. Since tasks can generally be fulfilled in more than one way, he compiles a list of alternative solutions and sends it back to the Task Group commander, who then attempts to choose the alternative that best utilizes his resources. If none of the alternatives can be reconciled with other already scheduled tasks, he can

1. either report back to the issuer, i.e., the Operations Level, or
2. can attempt to exchange resources with other Task Group commanders, or
3. can possibly define down or even drop some of the existing tasks, if the issuer has given him a specific permission to do so on his behalf.

Infeasibility to simultaneously provide for all the tasks currently in the task queue can trigger an extensive negotiating process running up and down the hierarchy, when the issuer can ask for sensitivity analysis of other tasks already in progress, their status and prospects. He can modify or drop some of them in order to make room for new ones. All those activities are conceptually supported in our system and, in fact, can happen automatically.

3 The nature of a task

As mentioned above, battle is the process of achieving a task's objective. Actually, we should use the plural, because the enemy has his objectives as well. This competitive process has its dynamics, which arise from the dynamic interaction of several components as depicted in Figure 2, and which each combatant tries to control in his favor. Prior to the first mission, each commander composes his package and sends it off to the battlefield. When the combat is over and survivors return to their bases, the commanders evaluate their battle damage assessment information and based on this feedback put together the second mission. Such iterations proceed until either one side succeeds in attaining its objectives or misses the deadline stated in its task specification and subsequently terminates the battle.

A control theorist readily recognizes in the description something that looks like a batch control problem. The Blue commander, who is the good guy whom we want to help, resembles a discrete controller which responds to the observed plant output by calculating a new control value to be applied next in order to bring the plant closer to meeting his objective. Admittedly, the control "value" is unusual and rather abstract, being represented by the composition, weapons, munitions and other attributes of the package sent into combat to drive the battle state in the desired direction. The BDA block represents sensors that map the state into the observable plant outputs, most likely with a lot of distortion, incompleteness, false readings and latency.

In addition to similarities, there are some important differences, too. As Figure 2 shows, Blue's links to the system go through the battlefield, which we view as a giant trading floor, where combatants trade their assets. In our approach, all deal-making is governed by probabilistic laws to reflect the random effects of weapons and other uncertainties of combat. Mathematically, we describe the trading which goes on on the battlefield as a Markov decision process, whose control variables are the package attributes. Such models addressing different kinds of combat have been developed and studied [Clark 1969], [Ancker 1982], [Ancker & Gafarian 1992], [Jelinek 2001a], [Jelinek 2001b]. Interpreting the battlefield as the trading floor also readily offers an important generalization of the notion of the package, namely viewing it as a set of all the assets the combatants bring for trade. While some of them will undoubtedly be weapons systems, others may be passive assets like bridges, airfields, power stations, etc.

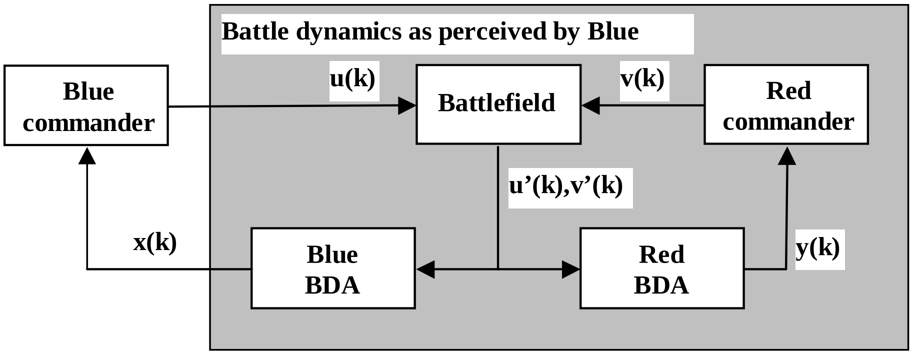


Figure 2. The internal structure of battle when viewed as a dynamic system

Unlike the mother Nature, which is an indifferent player in industrial applications, the Red commander has vested interest in the outcome of trading. Blue does not generally know what Red's interests exactly are, nor what strategy Red is going to pursue in advancing them. This brings in a different kind of uncertainty than the randomness of combat, one that must be addressed by the game-theoretic means. These problems have been intensively studied since the 1950's [Dresher 1961], [Basar & Olsder 1982].

Battle damage assessment poses yet another obstacle. In the world of smokescreens and deceit, even Blue's own BDA cannot be completely trusted. It is very difficult to find out, how much one actually does not know and somehow quantify this ignorance. For those reasons, we include the Blue's own BDA block into the battle dynamics model to stress the fact that the ground truth of the battlefield is not available to him for battle planning and management and that he can only see it through the lenses of his own BDA.

In order to rationally calculate the minimal effective force for a task, the Task Level commander has to model all the components enclosed in the gray box in Figure 2 along with their dynamic interaction. Compared to a control engineer facing a batch controller design problem, his plant-battle has not yet been "built" so his model cannot be, say, a neural network or some other regressor to be fitted to the plant using experimental data. Furthermore, in a typical operations center, hundreds of tasks are handled every day, so the ease and speed of model construction are paramount. Once built, the model will be run only once and then discarded.

The Task Group commander, whose job is to provide requested resources for tasks, deals only with their respective minimal effective forces. He is not interested in the details of how their they were calculated nor how the tasks' execution will be managed, and thus has no need to know the battle models. As it will become clear shortly, the minimal effective force specification is, in fact, the specification of a set of controlled random processes. It includes not only the immediate resource allocation request for the next mission, but, more importantly, provides a forecast of the expected battle evolution, including expected losses and resource demands for all future missions all the way to the task completion deadline. The challenge we are now facing is how to build a planner and scheduler that would take full advantage of this information to maximize the resource utilization and minimize disruptive plan and schedule modifications. Put in more mathematical terms, we are interested in the planning and scheduling of activities, which are characterized by sets of partially observable Markov decision processes operating over finite horizons [Sondik 1971], [Monahan 1982] or, in the most general form, partially observable competitive Markov processes [Filar & Vrieze 1997].

4 Minimal effective force

We have seen that each of the four components of the battle model in Figure 2 introduces uncertainties so fundamental that they cannot be ignored in the minimal effective force calculations. Uncertainty entails risk, which can be reduced by appropriate design, but never completely eliminated for very practical reasons: The cost of risk reduction tends to progressively escalate the faster, the closer we are approaching certainty, until at some point a low risk solution becomes unaffordable. We are thus caught between two conflicting interests. On one side, the task issuer wants to minimize the risk of task failure, because he will have to live with its consequences. On the other side, the task executor wants to lower the demand on his resources, which always seem in short supply, and is thus interested in relaxing the risk threshold. Furthermore, uncertainty in combat is not a constant, but varies in time as new information becomes available and effects of earlier decisions make an impact on the battlefield. This time variability requires continuous reevaluation of risk and making adjustments to the deployed forces, if the acceptable risk level is to be maintained with the minimal amount of resources. Having adopted this perspective as the central theme of our approach, we do not consider the notion of risk management to be just a metaphor. On the contrary, we view the C2 hierarchy as a hierarchical control system, which is to be explicitly designed to manage (control) risks arising from the uncertainties present at its different levels.

We define the risk ρ as the probability of task failure. Alternatively, we may be using the probability of success P_s, which is related to risk as $P_s = (1 - \rho)$. Let $u(k), v(k)$ be vectors, whose components are the attributes that characterize the packages employed in the k-th mission by the Blue and Red commanders, respectively. The attributes may be, for example, the numbers of bombers and escort fighters in a strike package, their weapon lethalities against the opponent's assets, combat tactics to be used, etc. Recall that not all assets brought for trade by the combatants are necessarily weapon systems. In the course of combat, some of the attributes are traded for others so that when the fighting is over, the status of the surviving packages will be $u'(k), v'(k)$. Due to the randomness inherent in combat, the particular values of $u'(k), v'(k)$ are impossible to predict. The best we can hope for is to find their probability distribution $P\{u'(k), v'(k)\}$. It can be computed providing that we know the conditional probability distribution $P\{u'(k), v'(k) \mid u(k), v(k)\}$ which we call the *combat model*. This model, in contrast to the full model of battle dynamics, describes only the block named Battlefield in Figure 2. Assuming that the vectors $u(k), v(k)$ take on only a finite number of discrete values, their sets $\mathcal{U}(k), \mathcal{V}(k)$ are also finite and the combat model $P\{u'(k), v'(k) \mid u(k), v(k)\}$ can be viewed as the set of transition probabilities of a Markov chain.

The vectors $u'(k), v'(k)$ generally are not directly available to the commanders for observation in their entirety. While the commanders usually get a good grasp of own losses, their information concerning their opponent's losses often is at best incomplete and at worst wrong. In our approach, we capture their less-than-perfect BDA capabilities by a pair of conditional distributions $P\{x(k) \mid u'(k), v'(k)\}$ and $P\{y(k) \mid u'(k), v'(k)\}$, which relate, although only in the probabilistic sense, the battle state $\{u'(k), v'(k)\}$ to the vectors $x(k)$ and $y(k)$, which represent the observables available to the Blue and Red commanders after the k-th mission, respectively.

Let $\mathcal{X}(k)$ be the sets of all possible vectors $x(k)$. Using Blue's task objective, we identify the subsets $\mathcal{X}_S(k) \subset \mathcal{X}(k)$, $\mathcal{X}_F(k) \subset \mathcal{X}(k)$ that Blue considers task success for himself and Red (i.e., Red's success is Blue's failure), respectively. If the battle reaches any one of those states, he terminates it. All other vectors are clearly indecisive situations, for which the battle continues with the $(k+1)$-st mission unless Blue has reached his task deadline, in which case he declares all such indecisive vectors $x(k)$ a success for Red (i.e., failure for Blue). Now it is easy to calculate the probability of Blue's success in the k-th mission

$$P\{x(k) \in \mathcal{X}_S(k) \mid u(k), v(k)) = \sum_{\mathcal{X}_S} \sum_{\mathcal{U} \times \mathcal{V}} P\{x(k) \mid u'(k), v'(k)\} P\{u'(k), v'(k) \mid u(k), v(k)\} \quad \textbf{(1)}$$

where we assume that the set $\mathcal{U} \times \mathcal{V}$ of all admissible values of the pairs $\{u'(k), v'(k)\}$ is finite. Let $J(n \cdots m)$ denote the probability of success in any of the missions between n and m

$$J(n \cdots m) = \sum_{i=n}^{m} P\{x(i) \in \mathcal{X}_S(i) \mid u(i), v(i)) \quad \textbf{(2)}$$

Then the minimum effective force $\{u^*(k), \ \ldots, u^*(K)\}$ calculated prior to the k-th mission is

$$\begin{aligned}&\{u^*(k),\ \ldots,\ u^*(K),\ v^*(k),\ \ldots,\ v^*(K)\} = \arg \min_{\mathcal{U}(k)} \max_{\mathcal{V}(k)} J(k \cdots K)\\&\text{so that}\\&J(1 \cdots (k-1)) + J(k \cdots K) \geq P_S\end{aligned} \tag{3}$$

where P_S is the desired probability of task success. For the sake of notational clarity, the obvious assumptions regarding the nonnegativity of attributes, etc. are omitted.

Note that the optimization is always done over the entire remaining task horizon. As a result, the minimum effective force specifies not only the package to be immediately employed in the upcoming k-th mission, but provides the estimates of all future packages all the way up to the task deadline. As will be shown in the next section, the estimates are continuously updated after each mission. Numerous experiments suggest that if Blue's battle model is reasonably accurate, total upsets of his expectations are rare. In most cases, the updates will be only modest corrections of the previous estimates, which enables the stable operation of forward-looking resource allocation planning and scheduling algorithms supporting the tasks at the Task Group Level. Second, as a side benefit of this game-theoretic optimization, Blue also develops his best estimate, $\{v^*(k),\ \ldots,\ v^*(K)\}$, of how Red is going to conduct the battle in the future given his current knowledge of Red's constraints. Although the problem statement (3) implies an assumption that the Red's sole objective is to prevent Blue from reaching his objective, which may be unrealistic in individual cases (and which turns our problem into a neat zero-sum game), other, more general game-theoretic formulations preserve this useful feature.

Since the problem statement (3) does not restrict what are the acceptable attribute values in the minimum effective force $\{u^*(k),\ \ldots,\ u^*(K)\}$, we call it the *Victory-at-Any-Cost* formulation. Most tasks, however, limit the amount of resources the Task Level commander is allowed to sacrifice. This extra constraint appears in the following *Victory-With-Acceptable-Loss* formulation [Jelinek & Godbole 2000].

$$\begin{aligned}&\{u^*(k),\ \ldots,\ u^*(K),\ v^*(k),\ \ldots,\ v^*(K)\} = \arg \min_{\mathcal{U}(k)} \max_{\mathcal{V}(k)} J(k \cdots K)\\&\text{so that}\\&J(1 \cdots (k-1)) + J(k \cdots K) \geq P_S\\&u_i^*(1) - u_i^*(K) \leq l_i, \qquad i \in \mathcal{I}\end{aligned} \tag{4}$$

where $\mathcal{I}$ is the set of attributes whose change between the first and last mission we want to limit so as not to exceed the given threshold l_i.

5 Model predictive risk control

Unless the battle terminated after the *(k-1)*-th mission, the commanders have to determine the package composition for the k-th mission. How the Red commander actually makes his decisions is rarely known to the Blue commander. Blue may know Red's doctrine and rules of engagement, and often has intelligence assessing Red's resources. If Blue is proactive and holds the initiative, he may reduce most of the Red's objectives to attempts at stopping him. Calling the games affords Blue an additional foresight into what to expect of his opponent. Furthermore, it is reasonable to expect the Red commander to be rational. All this constrains Red's decision space and enables Blue to produce a qualified estimate of Red's likely decisions when calculating the minimal effective force.

The above constraints are known a priori and are not tied to any particular task. In addition to them, however, Blue also receives real time BDA feedback from the battlefield in the form of the observation vector $x(k)$. This additional information allows him to either strengthen some of them or add new ones that reflect the customized, up-to-date knowledge about the task being prosecuted. When translated into the mathematics of the minimum effective force calculations (3) or (4), the feedback loop is closed by making the sets of admissible attribute values $\mathcal{U}(k)$, $\mathcal{V}(k)$ dependent on $x(k)$. Since combat results in asset attrition, these sets generally shrink with each mission, reducing the combatants' options in the process.

Using the model predictive control paradigm as a framework we have integrated the minimum effective force and real time feedback concepts into a system which we call the Model Predictive Task Commander (MPTC) and whose purpose is to assist the Task Level Commander in the planning and execution of individual tasks. The next section offers a simple illustration of how the MPTC works.

6 Example

The task specification may look as follows.

At 0600 the commander of the Blue Task Group TG_1 is given the order to destroy Red's surface-to-air missile (SAM) assets made up of L_R real sites and D_R decoy by 1800 tomorrow. Because the objective is needed to clear the way for an already planned subsequent offensive, the Operations Level requests the order be executed with a very high degree of certainty, say less than 1 in 20 chances that it will not be met in full. The Red's SAMs are known to have the lethality λ_R against the attacking aircraft that Blue is intending to use. They also have a good radar tracking capability to know the accurate numbers and positions of attackers in real time.

The MPTC helps answer the following questions:

- How many airplanes, L_B, does Blue need in his strike package, if his kill rate on the Red SAM's is known to be λ_B ?
- How many missions (sorties), K, he should divide his objective into, one, two, or perhaps ten?
- If he decides to fly more missions, how should he define their individual objectives, against which he could measure the task's progress once it gets underway? Without them, he would not be able to identify looming problems until it may be too late for any correction.
- If he decides to fly more missions, how should he optimally assemble the strike packages for each one? On one side, gradual enemy attrition will lower the threat, but he will have his losses as well. How big? What is the total number of aircraft he should ask to be allocated for the task?
- If, for whatever reason, the task execution does not proceed as planned, what corrective action to take?

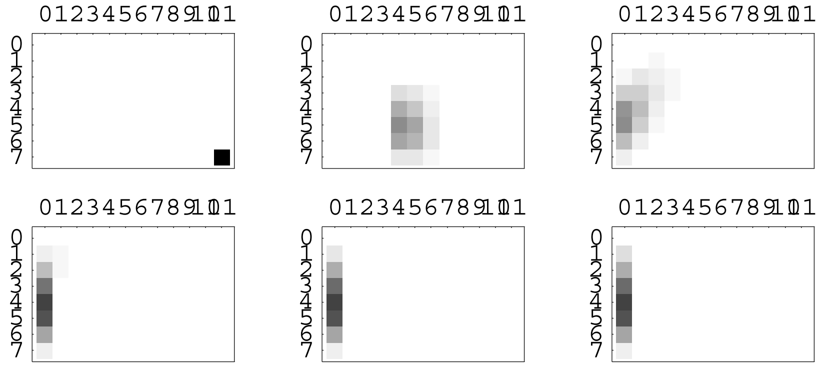


Figure 3. Probability distributions of the battle state $\{u'(k), v'(k)\}$ forecast how battle is going to evolve over the next 5 missions if Blue applies the minimum effective force calculated prior to the first mission. The first column are the win states for Blue.

To be specific, let us say that Blue's intel tells him that Red has $L_R = 11$ sites and $D_R = 1$ decoys. His SAMs are known to have the lethality against Blue aircraft $\lambda = 0.2$. On the other hand, Blue's weapons system data tell him that he can expect to kill this particular type of SAMs with the lethality $\lambda_B = 0.9$. Blue assumes that, when attacked, the Red commander will always employ all his surviving SAM sites to defend himself (which is actually the best policy for him in the game-theoretic sense). Because the mission turnover time is 6 hours due to the target distance, Blue can fly at most 6 missions before hitting the deadline. When Blue calculates his minimum effective force for $P_S = 0.95$ using the data, he is advised to employ 7 strikers for the first mission and then fight the remaining ones with survivors only (which is also the best policy for him). The MPTC also tells him that he can expect to lose slightly less than 3 aircraft on average in this job. The expected course of battle is shown in

Figure 3. The rows and columns in the frames correspond to the numbers of Blue and Red units that are alive, and plot densities are proportional to the state probabilities.

The top left frame is the initial state of battle, when Blue knows with probability 1 the initial numbers ($L_B = 7$, $L_R = 11$). The outcome of the first mission shown in the next frame is not that unequivocal anymore. The most likely number of survivors will be ($L_B = 5$, $L_R = 4$), but other outcomes still rather close to those numbers are possible. As the battle progresses (read Figure 3 row-wise), the cluster spreads more and more until it eventually splits into two, i.e., the distribution becomes bimodal. (This is not well visible in the figure.) The last frame in the bottom right corner says that Blue is most likely to win with 3 to 5 survivors, but battles with 2 or 6 survivors are a fairly likely outcome as well.

The MPTC repeats the same minimum force calculations after each mission upon receiving BDA feedback. The plots of their distributions would be similar except that the initial distribution, that is, the first frame in Figure 3, would be replaced by the latest battle state estimate and the planning horizon gets shorter with each new mission. The closed loop behavior is also a random process, whose distributions - of which we do not have their analytic forms to readily obtain their density plots - would not be much different from the open loop distributions shown in Figure 3.

If we performed multiple Monte Carlo simulations of the task execution under MPTC control, then each run would produce a new realization of the closed loop random process. A couple of runs shown in Figure 4 invokes the feeling for their variability. What matters, though, is that in spite of their vastly different appearance, Blue's MPTC will drive over 95% of all runs to success as requested by the task statement regardless of good or bad luck, which is always a factor in combat. As can be seen in the plot on the left, here the MPTC decided after the first and second missions that this battle was progressing better than expected and withdrew at first 3 and then 1 more survivors from action. In the battle on the right, Blue was down on luck and lost a lot more than expected in the first mission. Subsequently, the MPTC called for 2 additional airplanes to be added to the survivors to fly the second mission. Both battles shown happen to terminate after the third mission, but this is a coincidence. Battles can and do terminate anytime between 1 and 6 missions.

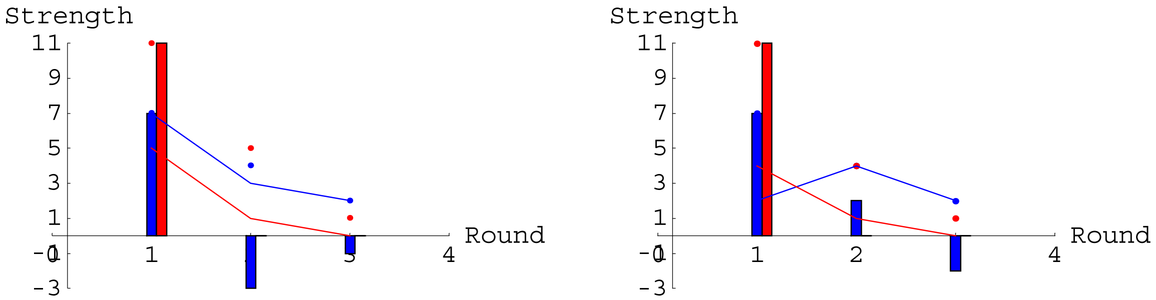


Figure 4. Monte Carlo simulation of two battles. Bars, dots and solid lines represent deployment increments/decrements, actually deployed and surviving units in each mission, respectively. Lacking color, the plots lose some of their explanatory power. The left and right bars in the first round (= mission) are Blue and Red, all bars in rounds ≥ 2 are Blue. The segmented line, which ends at the x axis is Red, because Red gets wiped out.

The true value of robust feedback control can be best appreciated when Blue does not get his battle model quite right. With so many unknowns to consider, such situations are very likely in the real world. Figure 5 shows on the left the average of 100 randomly generated battles managed by the MPTC, when Blue's model was right. On the right, the user underestimated the Red SAM's lethality by 50%, i.e., instead of entering $\lambda_R = 0.2$, he entered $\lambda_R = 0.1$. We see that the battles are generally longer and average withdrawals per mission are smaller. The simulation statistics for the perfect model example show that 99 battles were won, none lost in fight and 1 was lost by not being completed by the deadline. For the mismatched model, 93 battles were won, 1 lost on the battlefield and 6 were lost by not being completed in time. Although these numbers pertain only to the random samples of 100 battles, they are indicative of general results. The potential improvement the MPTC offers becomes clear once we calculate that in the model mismatched case 17 battles out of 100 would have been lost on average without it.

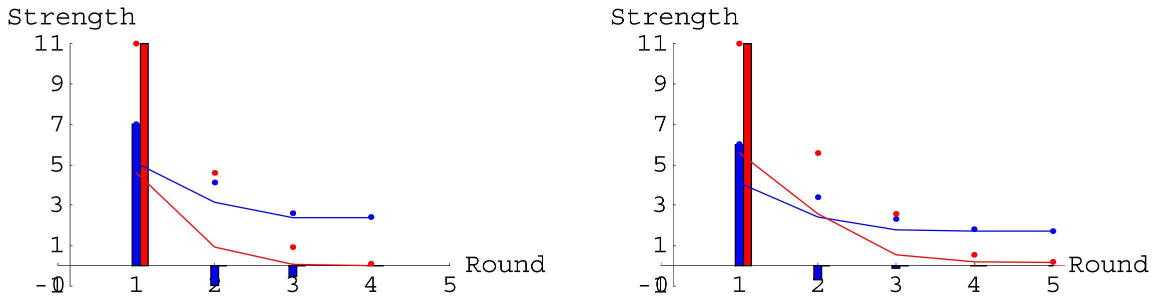


Figure 5. The expected evolution of battle computed as the average of randomly generated 100 battles for the perfect (left) and mismatched (right) models

7 Designing Package Defenses

A package can be viewed as an abstract weapon system that the Task Level commander custom designs and, with help from the Task Group Level commander, then "manufactures" for each mission to effect the desired change in the battlefield. As any other weapon, this one also has its offensive and defensive capabilities, which are characterized by the package attributes. While designing the offensive capabilities is, at least conceptually, straightforward, the design of defensive capabilities often is more difficult. As long as defensive components are an explicit part of a package, we use the formulation (4). However, many factors contributing to the package defense are intangibles that are very hard to model and quantify. The following concept offers a possible solution [Jelinek 2000].

Improving defense against enemy weapons, in effect, means lowering his weapons' lethality against own weapons. In other words, in this view Red's lethality λ_R should not be treated as a constant, but as a variable that Blue can actively manipulate in his favor. For example, flying a mission at night against the enemy whose fighters cannot fly by instruments would greatly lower his fighters' lethality against intruding Blue bombers. Likewise, adding a jammer aircraft to a package will lower the SAM's lethality by disabling their radar tracking. Just the threat of Wild Weasels leading a group of bombers might suffice to convince Red not to turn on his SAM radars at all, thus effectively lowering their lethality as well.

Let us assume that Blue can indeed manipulate the lethality of Red weapons and concern ourselves with the following question: How much down has Blue to drive the Red's lethality λ_R in order to keep his losses below a given threshold l_i? Once the MPTC suggests the needed lethality reduction, the Task Commander uses his military expertise to interpret it in terms of particular defensive options and suggest package defenses which are best suited for the task at hand. Mathematically, the solution is obtained by modifying the optimization problem (4) so that the minimization is carried out not only over the original set $\mathcal{U}(k)$, but also over the lethalities of Red weapons, which are now added to the Blue's decision variables.

Consider an example, in which Blue is tasked to destroy a bridge deep in the Red territory. Blue knows that on their way to the target, his bombers are likely to be intercepted by L_R Red fighters whose lethality $\lambda_R = 0.5$. The task specifies that his loss of bombers must not exceed 5 airplanes. How much must he lower λ_R to conform with the order? Figure 6 shows the results of the minimum effective force calculations for the acceptable maximum loss values ranging from 0 (the curve on the left closely following the y axis) to 9 (the last curve on the right, which is the lower envelope of the family). The thick curve provides the desired answer for maxLoss = 5. The example nicely demonstrates that our optimization problem does not have a unique solution as the needed number, nLB, of Blue bombers depends on their protection level that Blue is willing to provide. Any point of the thick curve meets the task objective, but offers a different offense-to-defense ratio for the package. Another fact worth pointing out is that trading off offense for defense has its limits. Even an unlimited number of bombers in the package cannot guarantee that their loss will not exceed 5 airplanes unless Blue defends them enough to drive the Red lethality (= pLR) down from 0.5 to about 0.38.

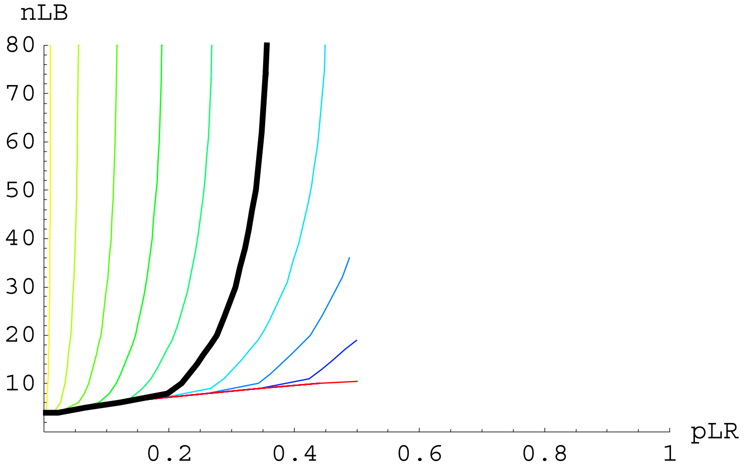


Figure 6. Many minimum effective force calculations offer multiple solutions. Any point of the thick curve meets the task objective, but offers a different offense-to-defense ratio.

8 Resource Allocation and Task Scheduling

Managing a set of tasks, which compete for shared resources, brings up new problems. The Model Predictive Resource Allocator (MPRA) proposed in [Tierno 2000] aims to achieve, whenever possible, the desired probability of success for all given tasks, while minimizing combat exposure of assets. It uses the market oriented programming to reconcile the competing demands. This is trivial as long as there is enough resources to satisfy all of the tasks' demands. It becomes a very difficult problem at the moment when the demands are irreconcilable and the MPRA is expected to provide some meaningful advice to the Task Group commander as to which tasks would be hurt the least if deprived of some of their resources, what effect such step would have on their probability of success, losses, etc. The MPRA may also be asked to advise which tasks can be redefined down without causing much harm or dropped altogether.

An appealing feature of this approach is the "currency" that bidders use in their attempts to acquire resources, which is based on the probability of success computed for each task by its MPTC. This injects some degree of objectivity into the way the algorithm works and makes its behavior easier to control and understand.

Mathematically, resource allocation is a packing problem. Such problems are notoriously hard to solve numerically. In real world applications, the Task Group commander has to handle tens or even hundreds of tasks simultaneously. This eliminates many potential algorithms, which simply cannot scale up to this problem size. For those reasons, work reported on in [Deshpande et al. 2001] investigated the use of greedy search and genetic algorithms to find only approximate solutions but in times more likely to be considered "real".

The above work concerned resource allocation done in a fixed time instant. There is no notion of the future in the resource allocation algorithms that were studied. Although the minimum effective force of each task offers an glimpse into its likely future, this information was ignored.

So far our team has not addressed the task scheduling problem at all. In their breakdown into individual, sequentially executed missions tasks do involve the notion of time, but this is only the "mission", not physical time. The job of scheduling is to map this mission time onto the physical time axis, and stack up the missions there so that they can be actually executed in the real world. We are addressing those issues in our current project.

Acknowledgements

This document is based upon work supported by the DARPA JFACC program through SPAWAR Systems Center San Diego under contract no. N66001-99-C-8510. Any opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of DARPA and SPAWAR Systems Center.

References

Ancker, C.J., Jr.: "One-on-One Stochastic Duels", Operations Research Society of America, 1982

Ancker, C.J., Jr. and Gafarian, A.V.: "Modern Combat Models", Operations Research Society of America, 1992

Basar, T. and Olsder, G.J.: *Dynamic Noncooperative Game Theory*, 2nd edition, SIAM 1999

Clark, G.M.: The Combat Analysis Model, Ph.D. Dissertation, Ohio State University 1969

Dresher, M.: *The Mathematics of Games of Strategy*, Dover Publications 1981. Originally published in 1961 by Prentice Hall, Inc.

Deshpande, R., Tierno, J., Parthasarathy, S.: Resource Assignment Across Multiple Tasks, DARPA JFACC Project Report, Honeywell Laboratories, September 2001

Filar, J. and Vrieze, K.: *Competitive Markov Decision Processes*, Springer 1997

Jelinek, J. and Godbole, D.: Model Predictive Control of Military Operations, Proc. 39th IEEE CDC Conf., Sydney, December 2000

Jelinek, J.: Winning Battles Without Losses, DARPA JFACC Project Report, Honeywell Laboratories, June 2000

Jelinek, J.: Predictive Models of Battle Dynamics, Proc. SPIE AeroSense Conf., Orlando, FL, April 2001

Jelinek, J.: Modeling Battles With Asynchronous Firing, DARPA JFACC Project Report, Honeywell Laboratories, September 2001 (a shorter version will appear in 2002 SPIE AeroSense Conf. proceedings)

Monahan, G.E.: A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms, Management Science, Vol. 28, No.1, 1982

Sondik, E.J.: The Optimal Control of Partially Observable Markov Processes, Ph.D. Dissertation, Dept. of Electrical Engineering, Stanford University, May 1971

Tierno, J.: Distributed Multi-Model Predictive Control, DARPA JFACC Project Report, Honeywell Laboratories, June 2000

Models of Defeat

Gary King Brent Heeringa David Westbrook Joe Catalano and Paul Cohen

Department of Computer Science
University of Massachusetts, Amherst
140 Governor's Drive
Amherst, MA 01003
{gwking,heeringa,westy,catalano,cohen}@cs.umass.edu

Abstract. Capture the Flag is a wargaming environment that includes intelligent, autonomous adversaries. In the past, the planner controlling the adversaries focused on two goals: occupying objectives and attrition. However, attrition is actually a means to an end, defeat of one's enemies, not an end in itself, and not necessarily desirable. Similarly, coalition operations plan for defeat by applying force for political and psychological effect. For Capture the Flag to plan for these effects, it needs a model of defeat. We model the "capacity for conflict" as a leaky bucket: when a unit's bucket is full, it has no more capacity for conflict and it capitulates. Flow into and out of the bucket is modulated by several factors including attrition and heroism. The model is inherently dynamical, so it exhibits the time-dependent behaviors one observes in real conflicts; for example, identical attacks will have different effects on capitulation as a function of their timing.

1 Introduction

Coalition operations are increasingly effects-based, which means they apply force only as necessary to achieve political and psychological effects. Actions that have relatively small effects in terms of conventional target-based or attrition-based planning can have large political and psychological effects, not only on adversaries but also on coalition members. AI planning technology has not kept up with the requirements of effects-based coalition planning, in part because we lack *models* of psychological and political effects. It is relatively easy to model the effects of attrition on conventional units — they get smaller, less mobile, less lethal, and so on — but what about their psychological state, their will to fight, their morale? Where are the models to predict the catastrophic collapse of the Iraqi regular units, or the differences between Taliban fighters from Saudi Arabia and those from Afghanistan?

This paper reports on our efforts to add models of defeat mechanisms to the Capture the Flag wargaming system. Defeat mechanisms are strategies that achieve capitulation. While attrition may achieve defeat, it is not necessarily the fastest or most desirable course of action. Although most planners are attrition-based, intelligent wargaming environments need agents that use defeat mechanisms to plan for the effects of their actions. For example, a smart agent might notice that an opposing unit has separated from its supply line and is ripe for an attack. It might also notice that attacking from a nearby forest is better than other routes because it will surprise the foe. While filling planners with rules like *always initiate attacks from hidden terrain* is possible, it is not necessarily desirable. Instead, we want agents that plan for effects: attacking an isolated unit from the forest is good because it is more easily defeated. That is, the effects of isolation and surprise make the foe more susceptible to defeat because its capacity for conflict is reduced.

If an agent is to plan for defeat it requires a model. This paper presents one such model that uses a metaphoric leaky bucket to represent an agent's capacity for battle. The bucket has inputs, outputs, and effects. While conceptually simple, the leaky bucket model paired with the Capture the Flag environment is flexible enough to account for many non-linear effects of battle. For example, differences in unit type, impact of friendly and opposing forces, and soft factors such as morale can all be represented with the leaky bucket and contribute, non-linearly at times, to defeat.

In the next section we review previous work in modeling defeat and discuss how our model differs from current research. Next we discuss the Capture the Flag wargaming environment and how our model naturally complements the simulator and planner. We follow this with details of the leaky bucket, specifically the mathematics of input and output flows and how they affect an agent's physical attributes. We demonstrate the flexibility and effectiveness of the model through a number of experiments and conclude with a discussion of future work.

2 Background

Historically, defeat models use hardcoded breakpoints of casualties to determine surrender or posture changes (Dupuy, 1990). Most researchers agree that such models are inaccurate and ill-suited for simulation. In particular, Dorothy Clark found that the breakpoints of casualty ratios in historical data fell uniformly between 10 and 70 percent (1954). She concludes that factors such as breakdown in leadership, support, reinforcement, and communication affect capitulation more than attrition. It was not until recently though, that such models were addressed for the purpose of simulation.

Janice Fain in (1990) provides two of the first non-attrition based breakpoint models for determining posture changes in computer simulation. Both models are essentially flow charts of conditional statements, but one flows with respect to time, the other by event. The variables in each condition are taken from historical data, interviews with veterans, and facts from the literature. The variable thresholds are calculated exclusively from battle data. This tends to model the historical data well, but may lead to overfitting. Fain's methodology also set the trend for future research: identify factors that influence defeat and model them directly.

In this spirit, a wealth of related literature exists both in the decision-making and human behavior modeling communities. For example Hudlicka and Billingsley (1999) develop a cognitive architecture for modeling individual characteristics such as personality and affective factors; McKenzie et al. (2001) investigate human personality models in decision-making; Gillis and Hursh (1999) integrate human performance models into simulation; and Angus and Heslegrave (1985) discuss cognitive abilities during command and control exercises in the event of sleep loss.

Recent work that deals directly with models of defeats include Alan Zimm's casual model of warfare (1999), and Brown and May's work in casting defeat as a breakdown in organizational structure within a complex adaptive system (2000). Zimm's work primarily identifies "stress factors" and their "cause and effect relationships" with a unit's behavior and decision-making skills. In contrast, Brown and May take a more biological slant on capitulation. When a unit can no longer adapt to the battle and its environment, it is primed for defeat. This argument is compelling and probably deserves more attention.

With the exception of Brown and May's research, most of the current and related work in defeat models deals with first-level characteristics of individuals. In Capture the Flag, this level of granularity is inappropriate since the agents are spring and blob masses and interact in an abstract world. In the next section we describes the Capture the Flag wargaming environment and how the bucket model works within and complements the system.

3 Capture the Flag

Capture the Flag is based in the Abstract Force Simulator (AFS) (Atkin et al., 2001; Atkin et al., 1999). AFS is a simulator of processes that operates with a small set of physical features including mass, velocity, friction, radius, attack strength and so on. The agents in AFS are abstract units called blobs; a blob can be an army, a soldier, or a political entity. Every blob has a small set of primitive actions it can perform: PRIMITIVE-MOVE, APPLY-FORCE (push), and CHANGE-SHAPE. All other actions are built from these. Blobs are modeled as a set of balls connected by springs where balls are point masses that can exert a force at some distance from their center. The ball and spring model means that blobs are amoeba-like: they can assume almost any two dimensional shape without holes.

We create simulations by changing the physics of AFS–how collisions affect mass and velocity, how terrain surfaces affect friction and so on. By tuning AFS, we have used it to simulate billiard balls, robots moving from room to room, rats scurrying about on a network of streets, and military battalions in division level combat.

AFS is tick-based, but the ticks are small enough to accurately model the physical interactions between blobs. Although blobs themselves move continuously in 2D space, for reasons of efficiency, the properties of this space, such as terrain attributes, are represented as a discrete grid of rectangular cells. Such a grid of cells is also used internally to bin spatially proximal blobs, making the time complexity of collision detection and blob sensor modeling no greater than linear in terms of the number of blobs in the simulator. AFS was designed from the outset to be able to simulate large numbers (on the order of hundreds or thousands) of blobs.

The physics of the simulation are presently defined by the following parameters:

Blob-specific parameters:

- shape
- density
- viscosity and elasticity: determine how blobs interact
- mass: the blob's ability to apply force
- position and velocity
- acceleration
- friction on different surfaces
- strength coefficient: a multiplier on mass to compute the force a blob can apply
- resilience coefficient: determines how much mass a blob loses when subjected to outside force

Global parameters:

- the different types of blobs present in the simulation (such as blobs that need sustenance or blobs than can apply force at a distance)
- the damage model: how blobs affect each others' masses by moving through each other or applying force
- sensor model: what information blobs can collect

AFS is an abstract simulator; blobs are abstract entities that may or may not have internal structure. AFS allows us to express a blob's internal structure by composing it from smaller blobs, much like an army is composed of smaller organizational units and ultimately individual soldiers. Because a blob is completely characterized by its physical attributes at every level of abstraction, we can ignore its internal structure while simulating if we choose to. Armies can move and apply force just as individual soldiers do. The physics of armies is different than the physics of soldiers, and the time and space scales are different, but the main idea behind AFS is that we can simulate at the "army" level if we so desire—if we believe it is unnecessary or inefficient to simulate in more detail.

In a similar fashion, we use abstract notions like *mass*, *strength* and *resilience* as stand-ins for the vast variety of actual unit attributes: weapon type, training, ammunition levels, supply lines, sickness, and so on. The mass of a blob agglomerates all of these and its strength and resilience account for the broad strokes of situation dependent factors. This loss of detail allows Capture the Flag simulations to be built and run in minutes rather than days. We can quickly assess multiple COA's in Monte Carlo trials and use our understanding for refinements in planning and strategy. Our simulation could be made much more detailed, but doing so runs the risk of arbitrary parameter choices and of pretending knowledge about what is best captured as noise and variance.

4 The Leaky Bucket Model

Innumerable factors influence whether or not an agent will cease to function. In Capture the Flag, we combine all of these factors into one abstract quantity we call fatigue. The fatigue of an agent rises and falls depending on its activities and interactions. Fatigue also alters these activities and interactions because an agent's fatigue changes its effectiveness. For example, as an agent's fatigue rises, it becomes less able to exert force and to protect itself, it moves more slowly, processes information less accurately, and so on. Finally, the agent has a breaking point. When an agent's fatigue becomes higher than this preset amount, the agent ceases to function.[1]

Using $\mathcal{F}_t$ and $\mathcal{E}_t$ to represent the fatigue and effectiveness (respectively) of an agent at time t, the following two general equations relate fatigue and effectiveness.

$$\begin{aligned} \mathcal{F}_t \quad = \quad & \mathcal{F}_{t-1} \\ & +f(\mathcal{F}_{t-1}) \quad \text{Loss} \\ & -g(\mathcal{F}_{t-1}) \quad \text{Recovery} \end{aligned} \tag{1}$$

$$\mathcal{E}_{t,i} \quad = \quad h_i(\mathcal{F}_{t-1}) \tag{2}$$

The new fatigue of an agent depends on its previous fatigue and two functions f and g which increase and decrease it. The effectiveness of an agent depends on another function h. In Capture the Flag, agent effectiveness is modeled as a multiplier on its strength, resilience, friction, turn rate, enemy intelligence abilities, sighting ability and so on. We call these altered agent properties the *effects* of fatigue. We subscript $\mathcal{E}$ and h to indicate that the change occurs for each altered agent property $\mathcal{P}$.

By varying the functions f, g and h, this model can become arbitrarily complex. We have chosen to keep these functions simple initially and to only add complexity when it seems necessary. We use the following functions:

$$f \quad = \quad \mathcal{M}_{self,t} + \frac{\mathcal{M}_{self,t}}{\mathcal{M}_{attacker,t}+\mathcal{M}_{self,t}} \quad \text{Differential mass loss} \tag{3}$$

$$g \quad = \quad \mathcal{R} \quad \text{a constant recovery factor} \tag{4}$$

$$h_i \quad = \quad \mathcal{P}_i(1 \pm \kappa\frac{\mathcal{F}_t}{\mathcal{B}}) \quad \text{for each effect} \tag{5}$$

Where we use the following notation:

$\mathcal{B}$	Breaking point
$\mathcal{P}_i$	Agent property effected by fatigue
$\mathcal{M}_{self,t}$	Agent's mass lost at time t
$\mathcal{M}_{attacker,t}$	Combined attacker's mass lost at time t
κ	The maximum percentage change in effectiveness due to fatigue.

We use the ± notation in equation 5 because some properties decrease as fatigue increases (e.g., strength and resilience multipliers) while others increase along with fatigue (e.g., friction). Each h_i will use the appropriate operation. We also make the following simplifying assumptions:

- Although the actual initial breaking point of an agent depends on its training, motivation and other intrinsic factors, we model it based solely on unit type.
- Each agent has a constant recovery rate that reduces its fatigue over time.
- The fatigue of an agent increases when it loses more mass that the units attacking it lose (i.e., the increase is based on the (perceived) differential mass loss).
- Rather than modeling each effect separately, we use the same percentage change in effectiveness for all of them.

[1] In the current implementation, agent's that have broken are removed from the game. We are considering providing agents with the ability to reconstitute and also with multiple breakpoints.

To summarize our model: an agent's fatigue rises when it is damaged and especially when its enemies damage it more than it damages them. The fatigue also has a natural constant recovery rate. The fatigue has a linear effect on the effectiveness of the agent where effectiveness is modeled by scaling the agent's key properties away from their nominal values. Though it is not explicit, this model is non-linear because as the fatigue rises, the effectiveness falls and as the effectiveness falls, the agent is liable to take more damage (and dole out less) which will cause the fatigue to rise more quickly.

One of the putative advantages of our model is that it has few parameters and all of them are reasonably intuitive:

$\mathcal{B}$	Breaking point or bucket size
κ	The maximum percentage change in effectiveness due to fatigue.
$\mathcal{R}$	a constant recovery factor

But these alone allow us to model different training levels (via increased bucket size or smaller κ); resilience to stress (by increasing R) and so on. By modifying f, g, and h we can complicate the model as necessary.

5 Experimental Results

As both Capture the Flag and our leaky bucket model operate in the abstract physics of AFS, it makes little sense to ask for quantitative results. Instead, we validate our model by seeing how well it matches the qualitative interactions of real military conflict. Any reasonably complex model can be tuned to fit almost any desired outcome. Our goal is to see if our simple model provides the right sort of behaviors without endless tuning. This section presents results from three different simulations

Two evenly matched blobs We ran 300-trials varying two independent variables: 1. Defeat Model: this could be on, off or on for only one blob; 2. Bucket size: this could be large or small. In each case, we randomly varied the initial mass and positions of the blobs.

A small blob against a much larger opponent We ran 50-trials in each of five conditions by setting the initial bucket level of the larger blob to 0-%, 30-%, 50-%, 70-% or 90-% of it maximum size.

Two blobs attacking a single, much larger blob We ran a total of 600-trials varying the total effect of the model (κ =10-% or κ =90-%) and the number of ticks between the attacks of the two blobs (0-, 15-, 30-, 45-, 60-, and 90-ticks). We also randomized the initial mass of each of the blobs. We did not randomize the positions because doing so added too much additional variance to the delay between the attacks.

In each simulation, we investigate if our model produces reasonable results.

5.1 Two evenly matched blobs

We might expect battles to last longer if blobs become less effective as they become fatigued–think of two drunken and weary boxers. On the other hand, if blobs can break and surrender, we might think that battles should end more quickly. We can observe both of these effects in this simulation. When the model is turned on, smaller bucket sizes lead to shorter battles (47-ticks as compared to 60-ticks for the larger bucket size). On the other hand, battles between blobs with high breaking points actually last longer (60-ticks as compared to 57-ticks) than the same blobs with no defeat model. Note that these battles may *last* longer but they actually do *less* total damage. As expected, battles with the defeat model turned on always produce less overall attrition that those with the model turned off.

5.2 A small blob against a single, much larger blob

If fatigue is not a factor, a small blob can never defeat a larger enemy in a head on assault. As the larger blob becomes fatigued, however, we would expect that it will suffer more damage and possibly even reach its breaking point. Furthermore, we would expect that the smaller blob would suffer correspondingly less damage. This simulation provides qualitative evidence of exactly these effects.

5.3 Two red blobs attacking a single, much larger blue one

All else being equal, it is always better to coordinate attacks. Adding fatigue to the simulation should greatly exacerbate the problems of uncoordinated attacks because blobs recover somewhat between attacks (at a rate determined by the outflow constant $\mathcal{R}$). Our simulations show how a coordinated attack succeeds where an uncoordinated one cannot and furthermore show significant differences when the blob effects from the defeat model are turned up high. Figure 1 shows the result of a coordinated attack. The x-axis shows time (in ticks) and the y-axis shows how full the blue blob's bucket is as a percentage of its total size. The stars on the graph show at what ticks the two red attacks occurred. As you can see, each attack causes an inflection in the graph. Because the attacks are coordinated, the blue blob has no time to recover and is overwhelmed. Figure 2 paints a completely different picture. The axes in this graph are the same but here the two red attacks are uncoordinated. The blue blob is able to defeat the first red blob and has time to recover before the second blob attacks. This recovery time allows it to defeat the second attacker and win the day.

In sharp contrast, figure 3 shows the difference in total mass lost when model effects are high and low. When fatigue effect is high, the blue blob cannot recoup its losses even with equal recovery time.

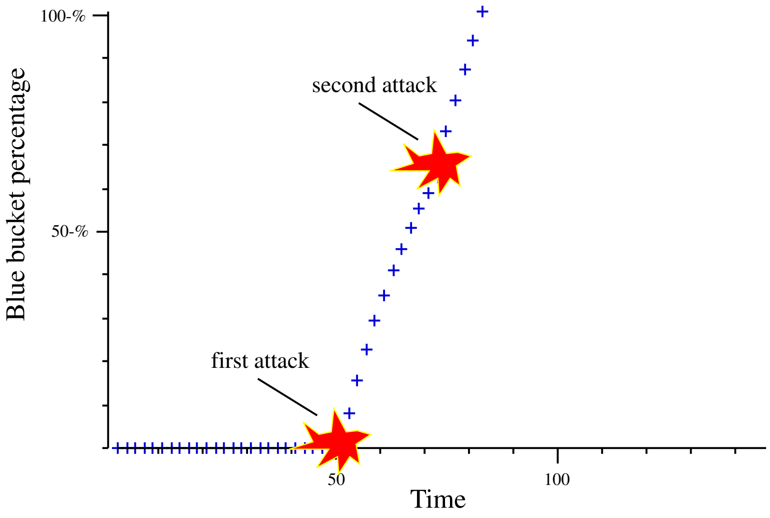


Figure 1: Blue bucket level over time, coordinated attack

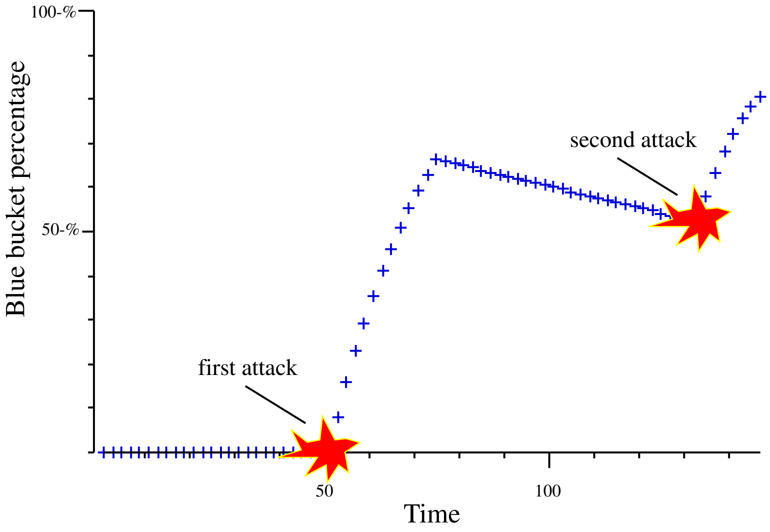


Figure 2: Blue bucket level over time, uncoordinated attack

6 Future Work

Our current model provides a simple, parameterized defeat model implemented within the Capture the Flag wargaming simulation. Our qualitative experiments show that the leaky bucket matches our expectations and increases the fidelity and range of Capture the Flag. There remains, however, much work to be done. For example, there are several plausible additions we can make to the individual agent model. Furthermore, although the Leaky Bucket model extends the behavior repertoire of single agents within the simulation, it does not capture the interactions between agents. Finally, we need to complete the circle and use our model to create plans which lead to capitulation by their effects rather than by brute force and attrition.

6.1 Model Extensions

The current model is deterministic whereas real battles are always characterized by the unexpected bravery or cowardice of individuals. We can capture the flavor of these events by adding a stochastic element to the bucket inflow and outflow functions (f and g). This would occasionally cause large decreases or increases in an agent's bucket level leading to renewed vigor or sudden defeat.

The current model also seems impoverished in its overreliance on blob combat as the only means of bucket level increase. We intend to investigate isolation, perceived vulnerability and terrain unfamiliarity as possible new sources of inflow. Some of these relate to the group dynamics of agents operating together to achieve their goals.

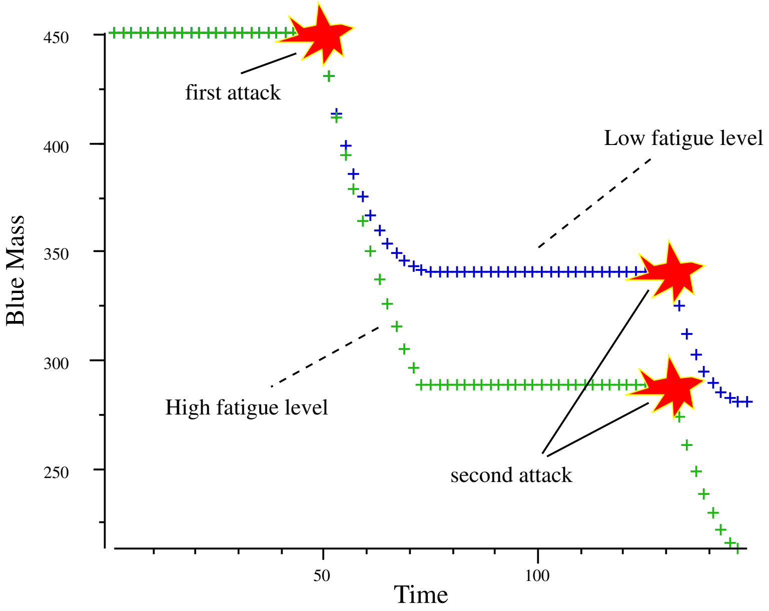


Figure 3: Blue mass level over time in uncoordinated attack comparing high and low effects of fatigue

6.2 Group Dynamics

The fatigue and morale of agents cannot really be viewed in a vacuum. Agents interact to uplift and poison one another; they respond to events all over the battlefield and in the world beyond; and they respond differently depending on their current situation and location. We will model all of these interactions by extending Capture the Flag with a layered network model of interconnections modeling Command and Communication, Supply, Infrastructure and so on (Cohen et al., 1996). Events will pass over this network and act as inflows and outflows on each agent's bucket.

7 Conclusion

Models of defeat are an integral component of intelligent wargaming environments for two reasons. First, models of defeat make simulation more realistic and agent behavior more accurate. Second, they provide a means for agents to execute defeat mechanisms – courses of action that achieve capitulation in military engagements. We presented a conceptually simple leaky bucket model that interacts with our Capture the Flag wargaming environment to capture many non-linear effects of defeat mechanisms. Qualitatively, our model behaves realistically and reasonably under a variety of different scenarios. In particular, we showed that the model is sensitive to the timing of attacks: coordinated attacks succeed whereas uncoordinated attacks fail. In the future we will experiment with agents that plan for the effects of defeat mechanisms. Such planning combined with our leaky bucket defeat model should result in a robust and realistic wargaming environment.

Acknowledgements

This research is supported by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Labs (AFRL) under contracts 30602-99-C-0061 and F30602-01-1-0589. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the DARPA, AFRL or the U.S. Government. Many thanks to General Charlie Otstott for his valuable comments and observations.

References

Angus, R. G., & Heslegrave, R. J. (1985). Effects of sleep loss on sustained congnitive performance during a command and control simulation. *Behavior Research Methods, Instruments, and Computers*, *17*, 55–67.

Atkin, M. S., King, G., Westbrook, D., Heeringa, B., Hannon, A., & Cohen, P. R. (2001). Hierarchical agent control: A framework for defining agent behavior. *Proceedings of the Fifth International Conference on Autonomous Agents* (pp. 425–432). ACM Press.

Atkin, M. S., Westbrook, D. L., & Cohen, P. R. (1999). Capture the Flag: Military simulation meets computer games. *Proceedings of AAAI Spring Symposium Series on AI and Computer Games* (pp. 1–5).

Brown, M., & May, A. (2000). Defeat mechanisms: Military organizations as complex, adaptive, nonlinear systems. Prepared for the Office of Net Assessment, Office of the Secreatary of Defense, under Contract No. DASW01-95-D-0060.

Clark, D. K. (1954). *Casualties as a measure of the loss of combat effectiveness of an infantry battalion*. Washington D.C.: Operations Research Officer.

Cohen, P. R., Anderson, S. D., & Westbrook, D. L. (1996). Simulation for ARPI and the air campaign simulator. *Advanced Planning Technology* (pp. 113–118). Menlo Park, CA: AAAI Press.

Dupuy, T. N. (1990). *Understanding defeat*. New York, New York: Paragon House.

Fain, J. B. (1990). Simulating defeat with computers: Forced changes of combat posture. *Understanding Defeat* (pp. 261–290).

Gillis, P. D., & Hursh, S. R. (1999). Using behavior moderators to influence cgf command entity effectiveness and performance. *8th Conference on Computer Generated Forces and Behavioral Representation*.

Hudlicka, E., & Billingsley, J. (1999). Representing behavior moderators in military human performace models. *8th Conference on Computer Generated Forces and Behavioral Representation*.

McKenzie, F. D., Catanzaro, J., & Petty, M. D. . (2001). A personality-based command decision-maker: Results and recommendations. *10th Conference on Computer Generated Forces and Behavioral Representation*.

Zimm, A. D. (1999). Modeling maneuver warfare: Incorporating human factors and decisionmaking in combat modeling. *Proceedings of the 8th Conference on Computer Generated Forces and Behavioral Representation*.

Issues of Dynamic Coalition Formation Among Rational Agents

Matthias Klusch and Andreas Gerber

German Research Centre for Artificial Intelligence
Multiagent Systems Group
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
{klusch,agerber}@dfki.de

Abstract. We introduce the notion, issues, and challenges of dynamic coalition formation (DCF) among rational software agents in open, heterogeneous and world widely distributed environments such as the Internet and Web. Selected relevant approaches coping with only parts of the DCF problem domain in different disciplines such as decision theory, social reasoning, and machine learning are briefly discussed. Finally, we sketch one novel DCF scheme, and highlight some future research work towards a general framework of dynamic coalition formation.

1 Introduction

Self-interested, autonomous software agents on the Internet may negotiate rationally to gain and share benefits in stable (temporary) coalitions. This is to save costs by co-ordinating activities with other agents. For this purpose, each agent determines the utility of its actions and productions in a given environment by an individual utility function. The value of a coalition among agents is computed by a commonly known characteristic function which determines the guaranteed utility the coalition is able to obtain in any case. In a characteristic function game the agents may use imposed individual strategies to achieve a desired type of economically rational behaviour such as altruistic, bounded rational, or group rational. In any case, the distribution of the coalition's profit to its members is de-coupled from its obtainment but is supposed to ensure individual rational payoffs to provide a minimum of incentive to the agents to collaborate.

Rational agents should also be able to form beneficial coalitions in open, distributed and heterogeneous environments at any and in reasonable time. That includes scenarios in which dynamically occurring events may interfere with the running coalition processes such as continuous change of tasks to be accomplished, information and computing resources available to the agents, as well as temporary disconnection of coalition partners in the network, and changes in their reputation and trust.

Due to its nature dynamic coalition formation methods promise to be particularly well suited for applications of ubiquitous and mobile computing, including mobile commerce. M-commerce as it may be supported by personalised, rational information agents residing, for example, on WAP-enabled access devices such as pagers, organisers, (sub)notebooks, or UMTS cell phones, currently still remains to be an appealing vision for the common Internet user. However, the development and application of DCF methods enabling potential business partners to form temporary coalitions on demand, on the fly, at any time may inherently enable and even advance the development of effective mobile commerce and collaborative work. This includes, for example, the challenge of quickly forming time-constrained, profit-oriented customer coalitions for optimally negotiating, purchasing and sharing appropriately partitioned sets of items at multiple electronic market places world wide in reasonable time. First approaches into this direction include, for example, (Tsvetovat & Sycara, 2000; Lerman & Shehory, 2000; Preist, Byde & Bartolini, 2001; Yamamoto & Sycara, 2001; Shehory, 2001).

The remainder of this paper is structured as follows. Section 2 summarises some static approaches of forming stable coalitions among rational agents. Issues and problems of dynamic coalition environments are discussed in section 3 while selected relevant approaches to cope with parts of these problems are surveyed in section 4. We sketch a novel DCF scheme in section 5, and conclude the paper with a brief outlook on future work.

2 Static Formation of Stable Coalitions

According to (Conte and Sichman, 1995) models of coalition formation may be classified into two main approaches: utility-based and complementary-based models dividing the societies of actors into ones following either the principle of 'bellum omnium contra omnes' as it is largely favoured, for example, by game theory (Luce and Raiffa 1957, Axelrod 1984), or ones which rely on the collaborative use of complementary individual skills to enhance the power of each agent to accomplish its goals, respectively.

Up to now, most classic methods and protocols for a formation of stable coalitions among rational agents follow the utility-based approach. They rely on derived concepts from co-operative game theory, economics, and operations research. Utilitarian coalition formation covers two main activities: (1) the generation of coalition structures, that is partitioning or covering the set of agents into coalitions, so as to maximise the monetary value depending on the benefit of accomplishing tasks regarding used resources and time spent; (2) the distribution of gained benefit among the participants of each of the coalitions. These activities may be interleaved and are not independent. A comprehensive discussion and classification of relevant work on coalition formation is given, for example, in (Kraus, 1997; Vauvert & El Fallah-Segrouhni, 2001).

2.1 Prerequisites

We briefly summarise the basic concepts and notions of co-operative game theory which are necessary to follow the discussion of coalition formation methods and the problems of dynamic coalition formation in subsequent sections. For a more comprehensive introduction to co-operative game theory we refer the reader to (Kahan & Rapoport, 1984; Osborne & Rubinstein, 1994; Holler & Illing, 2001).

2.1.1 Co-operative Games, Coalition Configurations

A **co-operative game** (*A, v*) is determined by a set *A* of agents wherein each subset of *A* is called a coalition, and a real-valued characteristic function *v: P(A)→R*, assigning each coalition its maximum gain, the expected total income of the coalition (the so-called coalition value). It is commonly assumed that (a) the value of any coalition *C* is in money, (b) the value *v(C)* does not depend on the actions of agents outside the coalition, (c) any coalition *C* forms by binding agreement on the distribution of its coalition value *v(C)* among its members, in particular no side-payments are allowed from *C* to any agents outside *C* within the game, and (d) the characteristic function *v* is known to all agents in *A*.

The solution of a co-operative game with side payments is a **coalition configuration** *(S,u)* which consists of

- a partition *S* of *A*, the so-called coalition structure, and
- an efficient payoff distribution $u : A \rightarrow \Re, (u(a_i))_{i\in\{1,..n\}} \in \Re^n$, $|A| = n$

The payoff distribution assigns each agent in *A* its utility out of the value of the coalition it is member of in a given coalition structure. It is commonly assumed that every coalition may form, including singletons or the grand coalition *A*. However, the number or size of coalitions to be formed using a coalition formation method is often restricted to ensure, for example, polynomial complexity of the formation process.

Individually rational distributions are assigning each agent at least the gain it may get without collaborating within any coalition, i.e.,$\forall a \in A : u(a) \geq v(\{a\})$, it is assumed to hold for any coalition configuration. For group rational distributions it holds that $\forall C \subseteq A : \sum_{a\in C} u(a) \geq v(C)$,i.e., the group of all agents is assumed to maximise its joint payoff.

In coalition configurations with so-called Pareto-optimal payoff distributions no agent is better off in any other valid payoff distribution for the given game and coalition structure. A coalition configuration *(S,u)* is called stable if no agent has an incentive to leave its coalition in *S* due to its assigned payoff *u(a)*. Each notion of stability defines a particular solution space for co-operative games. Concepts of stability applied to coalition configurations are discussed in the context of coalition formation methods in the following section 2.2.

2.1.2 Coalition Algorithm, Coalition Formation Environment and Model

Rational agents which are involved in a co-operative game *(A,v)* are supposed to negotiate a stable payment configuration *(S,u)* as a solution of the game by the use of an appropriate **coalition algorithm** CA which should have the following desirable properties.

- *Local execution*. Each agent is able to execute the CA locally. Negotiation according to the CA is completely decentralised.
- *Anytime*. After any regular termination of an arbitrary co-operative game in the considered environment the CA outputs a stable configuration as a solution of that game.

A **coalition formation environment** *CE* for a given set of agents *A* is the set of assumptions and constraints which are valid for any kind of coalition forming activity between agents in *A* including propositions on

- The functionality of each of the agents in *A*, including, for example, the sets of tasks, actions, and utilities of its task-related productions,
- Valid methods for computing the values of coalitions, for example, by the sum of production utilities of all agents in a coalition,
- Valid methods for determining coalition configurations, including methods for searching coalition structures, negotiation and payoff distribution schemes.
- Commitments, obligations of and agreements between agents in *A* concerning the type of collaboration and interaction.

In a given coalition formation environment the agents particularly agree on (a) what kind of stable coalitions shall be negotiated (the considered notion of stability), and (b) what particular coalition algorithm CA shall be used for the negotiation. Please note that agents may, for example, use different utility functions to evaluate the utilities of task execution and corresponding productions.

A coalition environment is called *super-additive* or *sub-additive* depending on the type of all co-operative games it allows, and *general* if it allows for both, sub-additive and super-additive games. In non-super-additive environments at least one (all) pair(s) of potential coalitions is not better off by merging into one which could be caused by, for example, communication and co-ordination overhead costs, decrease of coalition value as a result of restricting utility constraints posed by agents joining a coalition, or anti-trust penalties for specific coalitions (Kraus & Shehory, 1999).

A **coalition formation model** CM = (CE, CA) is defined by both, the considered environment CE and given coalition algorithm CA for this environment. Interesting models are those where coalition formation is concerned with general and sub-additive environments. In environments where published interests and utilities used for negotiation to form coalitions cannot be verified, most current coalition algorithms allow for fraud by different types of lies. Arbitration schemes for competing agents with conflicting interests may help to circumvent such situations (Tesch and Fankhauser, 1999).

2.2 Selected Coalition Formation Methods

As mentioned above, current coalition formation methods aim at building stable coalitions. The meaning of stability of coalitions varies dependent from the considered application domain and discipline. Many if not most of the coalition formation algorithms today rely on chosen game-theoretic concepts for pay-off division within coalitions according to, for example, the Shapley-value, the Core, the Bargaining Set, or the Kernel (Kahan & Rapoport, 1984). We briefly discuss selected main approaches to (static) coalition formation based on co-operative game theory in subsequent sections[1].

2.2.1 Core-stable coalitions

One approach to form stable coalition configurations as proposed in (Sandholm, 1999) comprises the following two steps: Searching for a social welfare maximising coalition structure in a corresponding coalition structure graph for the given game (A, v), and then compute its payoff division according to the stability concept of the core (Wu, 1977). The core of a game with respect to a given coalition structure is the set of coalition configurations with not necessarily unique payoff distributions such that no subgroup of agents is motivated to depart from the given structure. Only coalition structures that maximise the social welfare, i.e., the sum of all coalition values of coalitions in the considered structure, are Core-stable. However, searching for an optimal coalition structure (given a set *A* of agents) among the exponential number of $|A|^{|A|/2}$ possible coalition structures is computationally hard since one has to try at least $2^{|A|-1}$ coalition structures (Sandholm et al., 1998). Another well-known problem with core-stable configurations is that the core may be empty for certain co-operative games, and is exponentially hard to compute. This hardly suits the needs of solution approaches for dynamic coalition formation.

2.2.2 Shapley-value stable coalitions

Any pay-off division scheme according to the so-called Shapley-value (Shapley, 1953) provides an agent the added value (marginal contribution) that it brings to the given coalition structure, averaged over all possible joining orders. Obviously, the Shapley-value is exponentially hard to compute. In contrast to the core the Shapley-value is proven to uniquely exist, to be Pareto-optimal, and individual and group rational for super-additive games.

Algorithms for forming stable coalitions which rely on the stability concept of the Shapley-value and a variation of it, the so-called bilateral Shapley-value (Ketchpel, 1994) applied to arbitrary n-agent co-operative games, are proposed in (Klusch, 1997; Klusch & Shehory, 1996b; Contreras et al., 1997). It is shown in (Klusch, 1997) that the computation of proposed payoff division according to the bilateral Shapley-value with equal or history-based recursive share among coalition members is of polynomial complexity, and is guaranteed to be efficient and individual rational for super-additive games. However, since it is also shown that the latter fact does not necessarily hold for sub-additive games, these algorithms are not suitable to dynamic environments in their current form. Ongoing research is performed to devise novel methods for adapting these algorithms to such environments.

2.2.3 Kernel-stable coalitions

The Kernel of a co-operative game *(A,v)* with respect to a given coalition structure is the set of so-called K-stable configurations *(S,u)* in which all coalitions in *S* are in equilibrium. Coalition *C* is in such an equilibrium if each pair of agents in *C* is in equilibrium, i.e., any pair of agents in *C* is balanced, that is, none of both agents can outweigh the other in (S,u) by having the option to get a better payoff in coalition(s) not in *S* excluding the opponent agent. In other words, agents argument each other like "Since I could obtain more without you in alternative coalitions than you without me, I deserve more, but without going to harm you." For this purpose each agent has to compare its surplus with those of other agents; the calculation of the surpluses bases on that of the excesses of all alternative coalitions. Obviously, the kernel of a game is exponentially hard to compute unless, for example, the size of the coalition is limited by a constant. The kernel appears to be attractive due to the following features: The kernel K is

[1] One publicly available simulation environment for coalition formation among rational information agents based on selected classic coalition theories is, for example, COALA (Klusch & Vielhak, 1997).

unique for any 3-agent game (A,v), assigns symmetric agents of some coalition in a given coalition structure for (A,v) equal payoff, and is locally Pareto-optimal in K.

Polynomial coalition algorithms for polynomial K-stable coalition configurations have been developed and applied to the domain of co-operative information agents in (Klusch & Shehory, 1996b; Shehory & Kraus, 1996b; Klusch, 1997).

2.2.4 Fuzzy coalitions

Negotiation during the coalition forming process may be connected with various forms of uncertainty. Such uncertainties could be induced by the possibility of dynamically occurring events which, for example, may hamper the negotiation process and produce vague or incomplete knowledge on expected profits or the share of the income of coalitions in which they intend to participate. This in turn implies so-called fuzzy co-operative games with vague profits and has been dealt with in numerous works, for example, in (Mares, 2001; Aubin, 1981). A fuzzy co-operative game with side payments is consisting of a set of agents and a fuzzy characteristic function *v*, and the membership function *m* of the fuzzy quantities *v(C)* which may be interpreted as vague expectation of the common coalition profit that is to be distributed among its members. That is, the worth *v(C)* of a coalition *C* is a fuzzy set of its (possible) real-valued coalitional profits. This set of fuzzy quantity *v(C)* has at least one modal value, i.e., *m(v(C))=1,* determined by the membership function *m*. If for a given fuzzy co-operative game the coalition value *v(C)* is equal to one modal value of C for all possible coalitions *C*, it is equivalent to a (deterministic) co-operative game. The vagueness of the distributed profit *v(C)* means that particular payoff distributions can be realised with certain possibility only, which in turn is derived from the membership function *m*. Concepts of fuzzy super-additive co-operative games and "stable" fuzzy payoff distribution according to the fuzzy extension of the core and the Shapley-value are introduced and investigated in detail in (Mares, 2001). However, additional basic research on, for example, fuzzy sub-additive games and other concepts of "vague" stability remains to be performed, in particular appropriate coalition algorithms for fuzzy co-operative games have to be developed. This is topic of current research, for example, at DFKI.

2.2.5 Stochastic coalitions

Another class of co-operative games arises from co-operative decision making problems in stochastic environments. The notion of so-called stochastic co-operative games or co-operative games with stochastic payoffs, is introduced and investigated in (Suijs 1998; Suijs et al., 1999). A game with stochastic payoffs is defined by a set of agents, a set of possible actions coalitions may take, and a function assigning to each action of a coalition a real valued stochastic variable with finite expectation, representing the payoff to a coalition when this particular action is taken. Thus, in contrast to a deterministic co-operative game, the payoffs can be random variables, and the actions a coalition can choose from are explicitly modelled since the payoffs are not uniquely determined. It has been proven in (Suijs & Borm, 1999) that convex stochastic co-operative games are super-additive and have a non-empty core. Efficient coalition algorithms using these concepts are currently under development at DFKI.

However, all of the above mentioned as well as the vast majority of known other mechanisms for building utilitarian coalitions among agents remain static in the sense that they do not allow for any type of dynamic interference of running coalition formation processes. We will discuss types of dynamic events, corresponding problems and relevant approaches in the following sections.

3 Towards Dynamic Coalition Forming

The domain of *dynamic coalition formation* (DCF) among rational agents can be defined by the set of co-operation methods, schemes, and key enabling technologies to cope with the problem of dynamically building beneficial coalitions among agents in open, distributed, and heterogeneous environments such as the Internet.

3.1 The DCF Problem

The DCF problem rises in any collaboration environment and scenario in which at any time

(1) agents may enter or leave coalition formation processes,
(2) the set of tasks to be accomplished and the (computational) resources used, as well as
(3) the information, network, and user environment of each of the agents and the system as a whole may dynamically change.

Classical game-theoretic notions of coalition stability and respective negotiation algorithms are not applicable to such dynamic settings. Scenarios inducing uncertain, time-limited, context-based utilities and coalition values exacerbate the DCF problem. For example, an agent may determine the degree of membership to potential coalitions based on bargaining and the possible level of its commitment indicating the degree of collaboration that it desires.

3.2 Dynamic Coalition Formation Environments

As mentioned above, environments and settings in which rational agents have to be able to dynamically build coalitions can be characterised by the following classes of events and induced problems.

- *Tasks*: The set of tasks, goals and corresponding plans to accomplish may change for each individual agent at any time. Such changes concern, for example, the volume of tasks, utilities and costs of task execution as well as the frequency of such changes. This requires an agent to be able to perform, for example, fast dynamic re-planning of task execution to achieve its individual and/or common goals of the coalition. Re-planning concerns the granularity, re-usability and partiality or completeness of each of the considered plans. General task allocation problems are known as at least NP-hard problems. Real-time issues and requirements to perform planning under time-dependent uncertainty (Wellman, Ford & Larson, 1995) may even exacerbate these kinds of problems.
- *Agents*: Agents may leave or enter the agent society at any time, some agents may even temporarily hide their existence to parts of the society for different reasons.
-
- Optimisation:
- Negotiation:

We may distinguish between external and internal dynamic events. External events include, for example, a change in the specification of the problem to be solved by the agents, or any other change in the environment which are not caused by and cannot be influenced by the agents per se. Whereas internal events may be caused by the agents itself such as, for example, the entering or leaving of a coalition.

In dynamically changing environments rational agents may have to compute their individual utilities based on a pure sequence of local decisions. The problem of calculating an optimal complete mapping from states to actions (a so-called policy) in an accessible, stochastic environment with a known transition model is called a Markov decision problem. A transition model refers to a set of probabilities associated with the possible transition between states after any given action. Thus the agent is concerned with computing a sequence of values of stochastic variables X_t each of them is determined solely by the previous one. The resulting chain of probabilities $P(X_t|X_{t-1})$ yields a so-called Markov chain, a state evolution model. However, Markov chains and underlying decision support policies appear to be hardly feasible in open and dynamic environments for coalition formation. (Choi & Liu, 2001) propose one approach to mitigate the problem of prior knowledge on probabilities by using additional statistical information for the agents including the probability distributions of specific events to maximise their expected utilities without the need to of speculating others' actions. It remains to be investigated to what extent this approach can be generalised to coalition formation environments.

4 Selected Relevant Work

Relevant work on fuzzy coalition forming and co-operative games with stochastic payoffs (section 2.2), as well as rational revision of preferences, and other qualitative approaches to decision making based on partial, uncertain, and tentative information hold promise to be useful for coping with some of the issues of the DCF problem. We briefly discuss only some of the most relevant approaches and systems which are relevant for coping with parts of this problem. Other relevant work includes, for example, utility-based schemes for dynamically re-organising organisational structures (Barber & Martin, 2001), and exception tolerant reasoning and multi-criteria decision making under uncertainty (Benferhat et al., 2001; Dubois et al., 2000). These works may be properly extended for application to different dynamic coalition formation settings. The same hold with applying work on dynamic constraint satisfaction problems (Schiex & Verfaillie, 1993) since many of the above mentioned problems can be viewed naturally as CSPs (Eaton, Freuder & Wallace, 1998).

4.1 Game-Theory Based Approaches

4.1.1 Fuzzy and Stochastic Coalitions

Work on fuzzy and stochastic co-operative games as briefly described sections 2.2.4 and 2.2.5, respectively, is assumed to play an important role for the development of DCF schemes. Reasonable solutions for such types of games may lied to co-operation schemes which enable the agents to cope with issues of uncertainty, including, for example, vagueness of expected coalition values and corresponding payoffs. Such uncertainties may be induced by dynamic events such as network faults, changes of trust or reputation ratings of possible coalition partners, and receiving vague or even incomplete information and data during task execution or negotiation.

Both, the field of fuzzy and stochastic co-operative games still are in its very infancies and require further basic research efforts. This is even more valid for the application of principles and methods for such non-classical but still static coalition forming to dynamic settings. The development of algorithms for dynamic fuzzy or probabilistic coalition forming appears to be most promising and challenging at the same time. We are currently working on the development of such DCF algorithms.

4.1.2 Overlapping Coalitions

A method for building overlapping coalitions for precedence-ordered task-execution has been proposed in (Shehory & Kraus, 1996c). The suggested any-time algorithm is of polynomial complexity and yields sub-optimal results. Goal satisfaction by agents is approached as a problem of assigning goals to coalitions of agents. Thus the distributed algorithm tries to compute appropriate partitions of the considered set of agents adopting solution methods (Chvatal, 1979) for the similar set covering problem which is known to be NP-complete (Cormen, Leierson & Rivest, 1990). The algorithm is relevant for dynamic environments, wherein the time period for negotiation and coalition formation may be changed during the process.

4.2 Social Reasoning

Social reasoning mechanisms are considered as essential building blocks suitable to situations where agents may dynamically enter or leave the society, without any global control. Such mechanisms are often based on the notion of social dependence (Castelfranchi et al., 1992), or aim at reputation and trust management.

4.2.1 Social Dependence Networks

In order to acquire and use dependence knowledge on the considered agent society each agent has to (a) explicitly represent some properties of the other agents, which may change dynamically, (b) exploit this representation thereby optimising its behaviour according to the evolution of the society, and (c) to monitor and revise its representation to avoid inconsistencies to an acceptable degree, without any pre-established global control.

For example, the multi-agent system DEPINT (Sichman, 1995) illustrates some essential aspects of an agent's social reasoning mechanism in particular concerning the (a) adaptation of an agent to changes in goals and plans, (b) formation of coalitions for plan achievement, and (c) revision of inconsistent belief. Each DEPINT agent dynamically builds and maintains its individual network of dependency relations with respect to the accomplishment of goals based on the skills of its own and that of other agents in the agent society[2]. It may adapt to changes in goals to pursue and corresponding feasibility of plans to perform by using this dependency knowledge to select at any moment the goals and plans which it actually is able to execute by itself and/or with the help of the society. The agent evaluates the susceptibility of other agents to adopt its goals which in turn enables it to dynamically form respective coalitions for accomplishing its tasks.

However, DEPINT agents are assumed (a) to show benevolent behaviour in the sense that they do not try to exploit each other, never offer erroneous information deliberately and always communicate information in which they believe; (b) posses complete and correct knowledge of their own goals, expertise, etc., and (c) to perform belief revision once inconsistent or contradictory belief about others is detected. These assumptions appear unrealistic in open, dynamic coalition environments as described above.

4.2.2 Reputation and Trust Management

Social mechanisms of reputation management aim at avoiding interaction with undesirable participants and may complement other security technologies for authentication and authorisation. Mechanisms for building, propagating, measuring and maintaining reputation and trust (Yu & Singh, 2000; Manchala, 2000) are useful to apply, for example, to settings for coalition formation among self-interested agents in e-commerce applications where trusted third parties are required but not available. Negotiation schemes for uncertain games with trusted third party are proposed, for example, in (Wu & Soo, 1999; Soo, 2000). The merging of several individual trust matrices which are commonly used as a means for assessing trust relationships is not necessarily transitive and certainly requires further research.

In general, mechanisms which allow agents to efficiently react on frequent changes of reputation ratings and assessment of trustworthiness of potential coalition partners with respect to, for example, the expected share of profits, reliability of membership, and benevolence are, to our knowledge, more than rare up to date. First approaches into this direction include, for example, fuzzy models of reputation in multi-agent systems (Rubiera, Lopez & Muro, 2001).

4.2.3 Time-Constrained Reasoning

Rational agents may face many potentially beneficial choices related to the timing of events which may occur during (a) the individual decision process, and/or (b) the negotiation process with other potential coalition partners.

[2] A DEPINT agent is said to be dependent on another if the latter may facilitate or prevent it from achieving one of its goals. Both agents are mutually or reciprocally dependent on each other with respect to the same or different goals, respectively

Regarding the use of social reasoning mechanisms in continuously changing environments temporal dependence networks and adequate temporal social reasoning mechanisms are proposed, for example, in (Allouche, Boissier & Sayettat, 2000). These mechanisms may be applied to DCF schemes which rely in part on social reasoning.
Relevant work on real-time issues in the context of agent-based online auctions (on a single auction server) suggesting a design for maximal asychrony and robustness to network delay includes, for example, (Wellman & Wurman, 2000). (Choi & Liu, 2001) propose a dynamic mechanism for simple but time-constrained trading. The preliminary results and experiences reported in these and other relevant work may be taken into account for a design of more complex dynamic customer coalition formation schemes.

5 One DCF Scheme: DCF-A

In this section we propose a DCF scheme, called DCF-A, to enable rational agents to react on events which occur dynamically during the coalition forming process. In this paper we do not focus on the details of the coalition forming according to some given coalition model but on the simulation of the
Due to the dynamic nature of the environment in which the agents are situated in their behaviour may change over time. We include appropriate learning components into the DCF scheme DCF-A to adapt the individual pay-off matrix of each agent to the current situation using reinforcement learning (Sutton & Barto, 1998), especially Q-learning (Mitchell, 1997). The main idea is to approximate the function assigning each state-action pair the highest possible pay-off. Regarding the adaptation of each agent's world model to frequent changes in the agent society we adopt the concept of levelled reasoning on the behaviour of other agents as it is described in (Weiss, 1999).
In the DCF-A scheme each coalition built is represented by one distinguished agent acting as the so-called coalition leader. The coalition leader continuously attempts to improve the value of its coalition. In order to prevent the implied communication overhead between the leader and other members of the coalition, the leader simulates possible adjustments of the actual coalition configuration by building hypothetical re-configurations and rating them based on the members' capabilities, resources, desirability, communication stability, task description, and suggestibility from the current environment. As soon as the coalition leader achieves a significant improvement of the coalition value by simulation, it informs all its coalition members about proper alternatives. In turn, the agents have to send their estimation about the quality of relevant services and agents in regular time periods to the coalition leader or some so-called world utility agents. This is quite similar to the co-ordination and collaboration within so-called holonic multi-agent systems (Gerber, Siekmann & Vierke, 1999).

The coalition leader is assumed to be able to obtain up to date information about the agent society, for example, by request from some distinguished so-called 'world utility agent'. Such world utility information include public rankings about the quality of services offered by individual agents. Each agent may get a vague idea of the utilities and estimated payoffs of other agents, services, etc. When a new agent initialises itself and has no or less information on the world's entities, a global world utility function can give him a first hint while deciding what is a good choice to do next. The world utility on the one hand (in a benevolent agent society) can be used to give a global guideline for later evolution of the society. On the other hand (in a non-benevolent society) a group of agents may try to manipulation the world utility of some items for their own interests. But as more agents report their own estimation about entities listed at the world utility agent, the harder it will be to manipulate these utilities. Therefore we extend the world utility function by collecting the number of remarks from different agents for one ranked entity. Only the newest remark from an agent about an entity is stored. In addition, to avoid the world utility value from jumping from low to high, we extend the world utility function with proper learning mechanism. The world utility function provides a median of the incoming remarks and may provide common utility estimations of relevant items, entities and relationships of the society.

The DCF-A Scheme (Dynamic Coalition Formation Based on Simulation)

Variables and functions used by the DCF-A:

C configuration of a coalition (members, payoffs)
CPL list containing the changes (new partners) in of the coalition structure in relation to the current structure
AAL list containing the agents' abilities (capabilities, capacity, desirability, communication stability, stability of task description, suggestibility from the environment)
tp trust penalty for removing an agent from coalition *C*
cv current value of coalition *C* based on the Shapley-value
rvf () function to determine the risk value when adding an agent a_i to coalition *C* (Linsmeier & Pearson, 1996; Alexander, 1998)

Individual agent's preferences characterising its behaviour:

wr worst acceptable risk to remove a single agent a_i from C and getting punished from the agent society by loosing reputation *wtp* worst acceptable trust penalty for which the coalition head is willing to change the coalition structure with regards to all agents of the current *CPL*

k number of simulation cycles as an upper bound for the number of agents that have to be requested during the negotiation phase ($|C| \leq k$). A higher *k*-value denotes a higher risk in not getting all the changes of the coalition structure realised, but the chance to obtain a higher performance of the coalition is also higher.

Coalition formation and adjusting protocol used by each of the coalition leaders:

1. Initialisation Phase

```
    CPL = null
    halt = false
```

2. Simulation Phase

To prevent to get stuck in a local maximum and to avoid cyclic changes of the coalition structure, we use a randomised version of the algorithm for the simulation phase. The algorithm for the simulation phase is intended to run as long as it is not necessary to make changes of the coalition structure. In case of the occurrence of dynamic events it stops and presents a valid configuration which does not decrease the coalition value compared to that in the previous configuration. Therefore the agent does not change the current configuration, instead it builds hypothetical coalition structures and configurations, and simulates possible changes of them. During these iterations the actually best solution is stored in *BestCPL* such that the algorithm can be halted at any time and outputs a valid solution. The solution is not a degeneration of a previous solution since the simulation phase is stopped if and only if the value of the hypothetical configuration appears to be much better then that of the current configuration. The argument ‘much better’ is necessary to prevent too many changes in the coalition structure. The simulation phase is an any time algorithm.

```
    while not (halt) do
        requesting newAAL from distinguished world utility agent
        merging newAAL with local AAL: For this purpose we adopt learning mechanisms (Watkins, 1989; Sutton &
        Barto, 1998) and stochastic methods for agent ratings;
        CPL := null
        for (c=1 to k)
            choose randomly one operation for cycle c (noop, add_member, remove_member)
            if add_member then
                choose agent $a_i$ from AAL with [$min_{1 \leq i \leq |AAL|} rvf(a_i)$ and $max_{1 \leq i \leq |AAL|} value\{C+a_i\}$]
                insert tupel [$a_i$ , add] to CPL
            if remove_member then
                choose agent $a_i$ from AAL with [$max_{1 \leq i \leq |C|} rvf(a_i)$ and $max_{1 \leq i \leq |AAL|} value\{C-a_i\}$]
                if $rvf(a_i) > wr$ then
                    insert tupel [$a_i$ , remove] to CPL
                    $tp := tp + 1/ rvf( a_i)$
        next
        if value(CPL) > value (LastCPL) then
            // following types of dynamic events are considered: changes of the current coalition configuration, or
            changes in the environment or task requirements.
            BestCPL=CPL
        If value(BestCPL) >> cv and tp<wtp then
            // if a new coalition structure is found that is much better then the old one, then the simulation is stopped
            and the negotiation phase for realising the hypothetical coalition re-configuration begins
            halt = true
    while end
```

3. Negotiation Phase

Concerning the fact of a dynamic environment the term of stability of a coalition has to be properly modified. In our case of a dynamic scenario it is not possible to build stable coalitions in the classical game-theoretic sense. This is because at any time dynamic events may happen and the coalition configuration has to be adjusted in real-time. However, in situation where no dynamic events occur, the rankings of the agents are stable, the simulated coalition protocol finds the approximately best configuration (if it exists) and hold it until a change in the environment happens. After the simulation phase has stopped the *BestCPL* is used in the following negotiation phase, where the coalition leader tries to realise the corresponding hypothetically “best” configuration. It sequentially gets into a negotiation process with each agent of the *BestCPL* list based on a mechanism for 'multi-attribute negotiations' (Jonker & Treur, 2001). The agents have to negotiate about multiple attribute values, for example, the remaining time to fulfil a particular service, the costs of the service, etc. It is not guaranteed that all negotiations will end successfully. Thus, we adopt a 'levelled commitment protocol' (Andersson & Sandholm, 2001).

```
halt := false
for (i=1 to |BestCPL|)
    [$a_i$, $operation_i$ ]:= i-th tupel of BestCPL
    try
        if $operation_i$ = add_member then
            bilateral negotiation with agent $a_i$ based on protocols for multi-attribute negotiation and 'levelled
            commitment contracts' [1] (if not all agents of the BestCPL can be added to this coalition).
            if negotiation was successful then
                add $a_i$ to C
        else
            remove $a_i$ from C
    catch (if any dynamic event occurs during the execution of the negotiation phase)
        stop Negotiation Phase
next
```

4. Evaluation Phase

Send *AAL* to the known world utility agent, which merges this list with its local *AAL* (using learning mechanisms and stochastic methods for the agent rankings). Restart the simulation phase (Go to 2.)

6 Conclusions

We introduced the notion, selected issues, and challenges of dynamic coalition formation (DCF) among rational software agents. In addition, we briefly discussed selected relevant work in different disciplines and proposed a novel DCF scheme. It has to be emphasised that one of the main challenges of the domain of dynamic coalition formation is the development of efficient DCF algorithms which enable rational agents to efficiently cope with different hard issues and problems they are facing in continuously changing, open, distributed and heterogeneous environments such as the Internet and Web. This is one focus of ongoing and future research, for example, at DFKI. For this purpose, many relevant approaches and theoretical work stemming from different disciplines are available to date including work on temporal social reasoning, and fuzzy and stochastic co-operative games.

References

Alexander, C. (1998) "Risk Management and Analysis, Risk Measurement and Management" ISBN: 0-471-97957-0, 1998.

Andersson, M. and Sandholm, T. (2001) "Leveled Commitment Contracts with Myopic and Strategic Agents" Journal of Economic Dynamics and Control, Vol. 25, pp. 615-640, Special issue on Agent-Based Computational Economics, 2001.

Aubin, J.P. (1981) "Cooperative fuzzy games" Journal Mathematics of Operations Research, Vol. 6(1), February 1981

Axelrod, R. (1984) "The Evolution of Cooperation" Basic Books, New York, 1984.

Barber, K.S. and Martin, C.E. (2001) "Dynamic re-organisation of decision-making groups" Proceedings 5th International Conference on Autonomous Agents, Montreal, Canada, ACM Press, May/June, 2001.

Benferhat,S., Dubois, D., Prade,H., and Sabbadin,R (2001) "Exception –tolerant reasoning and decision uner uncertainty using qualitative possibility theory" Journal Künstliche Intelligenz, Vol. 3, arendtap Verlag, Beremen, 2001.

Castelfranchi, C., and Micelli, M., and Cesta, A. (1992) "Dependence Relations Among Autonomous Agents" W Eric and Y Demazeau (eds.), Decentralized A. I. 3, Elsevier Science Publishers, pp 215-227, 1992.

Choi, S.P.M. and Liu, J. (2001) “A dynamic mechanism for time-constrained trading” Proceedings 5th International Conference on Autonomous Agents, Montreal, Canada, ACM Press, May/June, 2001.

Chvatal, V. (1979) “A greedy heuristic for the set-covering problem” Journal of Mathematics of Operations Research, Vol. 4(3), pp. 233-235, 1979.

Conte, R., and Sichman, J.S. (1995) “DEPNET: How to benefit from social dependence” Journal of Mathematical Sociology, 20(2-3), 161-177, 1995.

Contreras, J., Wu, F., Klusch, M. and Shehory, O. (1997) “Coalition formation in a power transmission planning environment”, Proceedings of International Conference on Practical Applications of autonomous Agents and Multi-Agent Systems (PAAM-07), London, UK, April 1997.

Cormen, T.H., Leierson, C.E., and Rivest, R.L. (1990) “Introduction to algorithms” MIT Press, CA, USA, 1990.

Dubois, D., Grabisch, Modave, F., and Prade, H. (2000) “Relating decision under uncertainty and multicriteria decision making models” Journal of Intelligent Systems, Vol. 15, pp. 967-979, 2000.

Eaton, P.S., Freuder, E.C., and Wallace, R.J. (1998) “Constraints and agents – Confronting ignorance” AI Magazine, Vol. 19(2), 1998.

Gerber, C., Siekmann, J., and Vierke, G. (1999) „Holonic multi-agent systems”. Technical Report Research Report RR-99-03, DFKI, 1999.

Holler, M.J. and Illing, G. (2000) “Einführung in die Spieltheorie” Springer Heidelberg, 4th edition, 2000

Jonker, C.M. and Treur, J. (2001) “An Agent Architecture for Multi-Attribute Negotiation” Proceeding of 17th International Joint Conference on Artificial Intelligence (IJCAI’01), Vol.2, pp. 1195-1201, 2001.

Kahan, J.P., and Rapoport, A. (1984) “Theories of coalition formation” Lawrence Erlbaum Associates, New Jersey.

Ketchpel, S. (1993) “Coalition formation among autonomous agents” Proceedings of European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-93), 1993.

Ketchpel, S. (1994) “Forming coalitions in the face of uncertain rewards” Proceedings of AAAI Conference, AAAI-94, pp. 414-419

Klusch, M. (1997) “Kooperative Informationsagenten im Internet”, Dissertation, Schriftenreihe Forschungsergebnisse zur Informatik, Dr. Kovac Verlag, Hamburg, ISBN 3-86064-746-6, 1997.

Klusch, M. and Shehory, O. (1996) “Coalition formation among rational information agents” Proceedings of European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-96), W Van de Velde and JW Perram (eds.), Lecture Notes in Artificial Intelligence, Vol. 1038, pp. 204-217, 1996.

Klusch, M. and Shehory, O. (1996b) “A polynomial kernel-oriented coalition formation algorithm for rational information agents” Proceedings of International Conference on Multi-Agent Systems (ICMAS-96), Kyoto, Japan, December 1996, pp. 157-164.

Klusch, M. and Vielhak, T. (1997) COALA - Simulation Environment for Coalition Formation Among Autonomous Information Agents. http://www.dfki.de/~klusch/COALA/

Kraus, S. (1997) “Negotiation and cooperation in multi-agent environments”, Journal of Artificial Intelligence, Vol. 94(1-2), pp. 79-98.

Larson, K. and Sandholm, T. (2000) “Anytime Coalition Structure Generation: An Average Case Study” Journal of Experimental and Theoretical AI, Vol. 12, pp. 23-42.

Linsmeier,T.J. and Pearson, N.D. (1996) “Risk measurement: an introduction to value at risk” ACE OFOR from University of Illinois at Urbana-Champaign. http://agecon.lib.umn.edu/uiuc/aceo9604.pdf

Luce, R. D. and Raiffa, H. (1957) “Games and Decisions: Introduction and Critical Survey”, John Wiley & Sons Ltd., 1957.

Manchala, D.W. (2000) “E-commerce trust metrics and models” IEEE Internet Computing, Vol.4(2), March/April, 2000.

Mares, M. (2001) “Fuzzy cooperative games – Cooperation with vague expectations” Physica Verlag, 2001.

Rubiera, J.C., Lopez, J.M.M., and Muro, J.D. (2001) “A fuzzy model of reputation in multi-agent systems” Proceedings 5th International Conference on Autonomous Agents, Montreal, Canada, ACM Press, May/June, 2001.

Sandholm, T. (1999) “Distributed Rational Decision Making” G. Weiss (ed.), “Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence”, MIT Press, pp. 201-258, 1999.

Sandholm, T. and Lesser, V. (1997) “Coalitions among Computationally Bounded Agents” Journal of Artificial Intelligence, 94(1), Special issue on Economic Principles of Multiagent Systems, pp. 99-137.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., and Tohmé, F. (1999) “Coalition Structure Generation with Worst Case Guarantees” Artificial Intelligence, 111(1-2), pp. 209-238.

Schiex,T. and Verfaillie,G. (1993) “No-good recording for static and dynamic constraint-satisfaction problems” Proceedings of International Conference on Tools with AI, pp. 48-55, Washington DC, IEEE Computer Society Press, 1993.

Shehory, O. and Kraus, S. (1996a) “Cooperative goal-satisfaction without communication in large-scale agent-systems” Proceedings of European Conference on AI (ECAI-96), pp. 544-548, Budapest, Hungary, August 1996.

Shehory, O. and Kraus, S. (1996b) “A Kernel-oriented model for autonomous-agent coalition-formation in general environments: implementation and results” Proceedings of AAAI Conference, pp. 134-140, Portland, Oregon, August 1996.

Shehory, O. and Kraus, S. (1996c) “Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents” Proceedings of International Conference on Multiagent Systems (ICMAS-96), pp 330-337, Kyoto, Japan, December 1996.

Shehory, O. and Kraus, S. (1998) “Methods for task allocation via agent coalition formation”, Journal of Artificial Intelligence, Vol. 101 (1-2), pp. 165-200, May 1998.

Shehory, O. (2001) “Optimality and Risk in Purchase at Multiple Auctions“ Proceedings of 5th International Workshop on Cooperative Information Agents (CIA 2001), M Klusch, F Zambonelli (eds.), Lecture Notes in Artificial Intelligence, Vol. 2182, Springer, 2001.

Sichman, J.S. (1998) “DEPINT: Dependence-Based Coalition Formation in an Open Multi-Agent Scenario” Journal of Artificial Societies and Social Simulation vol. 1, no. 2, March 1998.

Soo, V.W. (2000) “Agent negotiation in trusted third party mediated uncertain games” Proceedings of 4th International Conference on Autonomous Agents, Barcelona, June, ACM Press, 2000.

Suijs, J. (1998) “Cooperative Decision Making in a Stochastic Environment” CentER Dissertation Series #36, CentER. Tilburg, 1998.

Suijs, J. and Borm, P. (1999) “Stochastic cooperative games: superadditivity, convexity and certainty equivalents” Journal of Games and Economic Behavior, Vol. 27, pp. 331-345, 1999.

Suijs, J., Borm, P., and De Waegenaere, A., and Tijs, S. (1999) “Cooperative games with stochastic payoffs” European Journal of Operational Research, Vol. 113, pp. 193-205, 1999.

Sutton, R.S., and Barto, A.G. (1998) “Reinforcement Learning: An Introduction”, MIT Press, Cambridge, MA, Bradford Book, 1998.

Tohmé, F. and Sandholm, T. (1999) “Coalition Formation Processes with Belief Revision among Bounded Rational Self-Interested Agents” Journal of Logic and Computation, Vol. 9(6), pp. 793-815.

Tsvetovat, M. and Sycara, K. (2000) “Customer coalitions in the electronic marketplace” Proceedings of 4th International Conference on Autonomous Agents, Barcelona, June, ACM Press, 2000.

Vauvert, G., and El Fallah-Segrouhni, A. (2001) “Coalition formation among strong autonomous and weak rational agents”, Proceedings of the 10th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW), Annecy, France, May 2001.

Vauvert, G., and El Fallah-Seghrouchni , A. (2000) “Coalition formation among egoistic agents”. Proceedings of International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce (MAMA'2000.), Wollongong, Australie, December 2000.

Watkins, C.J.C.H. (1989) “Learning from Delayed Rewards”. PhD thesis, King’s Collage. Cambridge University, 1989.

Wellman, M.P. and Wurman, P.R. (1998) “Real time issues for Internet auctions” Proceedings 1st Workshop on Dependable and Real-Time E-Commerce Systems (DARE-98), Denver, CO, USA, June 1998.

Wellman,M.P., Ford, M., and Larson, K. (1995) “Path planning under time-dependent uncertainty” Proceedings of 11th Conference on Uncertainty in AI (UAI-95), Montreal, Canada, August, 1995

Wu, L.S. (1977) “A dynamic theory for the classes of games with non-empty cores” SIAM Journal of Applied Mathematics, Vol. 32, pp. 328-338, 1977.

Wu, S.H. and Soo,V.W. (1999) “Risk control in multi-agent coordination by negotiation with a trusted third-party” Proceedings of International Joint Conference on AI (IJCAI-99), Stockholm, Sweden, 1999.

Yamamoto, J. and Sycara, K. (2001) “A stable and efficient buyer coalition formation scheme for e-marketplaces” Proceedings 5th International Conference on Autonomous Agents, Montreal, Canada, ACM Press, May/June, 2001.

Yu, B., Singh, M.P. (2000) “A social mechanism of reputation management in electronic communities” Proceedings of International Workshop on Cooperative Information Agents IV, M Klusch & L Kerschberg (eds.), Lecture Notes in Artificial Intelligence, Vol. 1860, Springer, July 2000

Coalition application of the Joint Battlespace Infosphere (JBI)

Mark H. Linderman and Vaughn Combs
Air Force Research Laboratory/IFTC
{mark.linderman, vaughn.combs}@rl.af.mil

Publish/subscribe information dissemination mechanisms are well suited to coalition engagements because they are inherently flexible and able to adapt to many of the challenges of standing up a coalition information infrastructure. However, as one moves beyond the abstract notion of publish and subscribe (pub/sub) to the realities of current commercial capabilities and requirements of military deployments in coalition environments, it becomes apparent that there are several challenges to achieving the promise of pub/sub information exchange architectures.

Publish/Subscribe architectures are used to disseminate information from those who have it (publishers) to those who need it (subscribers). Subscribers typically express their information needs with a predicate to limit the amount of extraneous information they receive. In a coalition deployment, however, the right to publish and subscribe to specific information is subject to policy. In particular, publishers of information and overarching command authority may impose releasability restrictions (e.g. NATO-ONLY) on information that must be satisfied by the pub/sub infrastructure.

This paper considers the current state of commercial publish/subscribe technology and its relevence and limitations for coalition military deployment. Specifically, we will address how information is represented, filtered and controlled. Finally, we will describe the Joint Battlespace Infosphere, a US Air Force Research Laboratory project that seeks to harness the power of publish/subscribe architectures in a military context.

Software Agents for Coalition Forces

Zakaria Maamar
Software Agents Research Group$_{@ZU}$
College of Information Systems
Zayed University
PO Box 19282, Dubai, U.A.E
zakaria.maamar@zu.ac.ae

Paul Labbé
System of Systems Section
Defence Research Establishment Valcartier
2459 Pie-XI Blvd. North Val-Bélair
(Québec), Canada, G3J 1X5
paul.labbe@drev.dnd.ca

Wathiq Mansoor
Software Agents Research Group$_{@ZU}$
College of Information Systems
Zayed University
PO Box 19282, Dubai, U.A.E
wathiq.mansoor@zu.ac.ae

Abstract. The distributed, heterogeneity, and dynamic nature of the coalition context has raised the need for new advanced technologies. These technologies aim at managing the coalition informational infrastructure, in terms of autonomy, adaptability, and scalability. To achieve this support, Software Agents (SAs) seem to be a promising approach. To develop this approach, different aspects of a coalition has to be identified. These aspects include the coalition structure; the roles and responsibilities held by people within the coalition; the flow of information within the coalition; the capabilities required or available within the coalition; and the context in which the coalition operates. For many of these aspects, SAs can be used; . For instance, the coalition structure can be associated with several SAs of different types and with different roles.

1. Introduction

We discuss our research work on the design and development of collaborative environments for distributed and heterogeneous military applications. These applications, called Command & Control Information Systems (CCISs), are increasingly important for land, naval, and air operations. Moreover, CCISs have civilian applications in multiple areas such as air traffic control, search & rescue, and emergency services. In a military context, a commander makes decisions concerning his troops deployment using the information supplied by the CCIS. It may occur that this commander aims at involving other friendly CCISs before taking his decisions. For example, a Canadian commander has to take into account the positions of the enemy and friendly troops. Therefore, he has to involve other CCISs that may possess such an information. It would be more appropriate if this commander could perform this operation without being aware of each CCIS's characteristics. Take for instance a situation where different countries decide to set up a coalition for an international humanitarian assistance. In fact, the CCIS of each country has its own functional and structural characteristics. It is impossible for a commander to be aware of all the CCISs' locations, languages, and information semantics. Therefore, it becomes urgent to propose new support technologies that will free military users from worrying about the distributed, heterogeneous, and dynamic nature of the coalition, in general and CCISs in particular. In this paper, we describe the IC2MAS (Interoperable Command & Control based on MultiAgent Systems) project that aims at managing the coalition infrastructure at the following levels (adapted from [Babin et al., 1994]):

- Autonomy: in the coalition environment, CCISs should have the flexibility to be designed, developed, and managed independently, without having to comply with this environment's standards.
- Flexibility: CCISs, that use either standard or non-standard technologies, as well as new and legacy CCISs, should be incorporated into the coalition environment in a "seamless" way without causing any disruption to this environment.

- Scalability: the total coalition environment should be expandable by allowing this coalition to start with a number of countries and gradually extend over time, without loosing integrity.

Taking into account these three levels and the requirements of a coalition (discussed in Section 3.1), the IC2MAS project has established an interoperability approach to provide effective support to a coalition.

The motivation behind the support of a coalition is to provide an integrated view of all the aspects that are relevant to this coalition. These aspects are multiple and include: the coalition structure; the roles played by people and responsibilities held by them within the coalition; the flow of information within the coalition and with the external world; the resources required by and available within the coalition; and the context (war, peace-making/keeping, from war to peace-making, and from peace-keeping to war) in which the coalition takes place. MASs could handle a number of these aspects. For instance, the coalition structure could be viewed as a collection of collaborative MASs; each MAS could correspond to a CCIS and each MAS could contain different types of agents, fulfilling different roles, and carrying out different responsibilities.

In the IC2MAS interoperability approach, MASs are the CCIS's front-ends to the coalition network and hence, have the capability to act on their behalf. Moreover, MASs encompass different Software Agents (SAs) [Green et al., 1997] that handle and perform the functionalities required to coalition support, for example managing the CCISs' autonomy and invoking CCISs. However, given the distributed nature of a coalition and the network features in terms of reliability and bandwidth capacity (e.g. the coalition could occur in a country in which the network infrastructure is not well developed), some of the SAs in the IC2MAS approach are able to create Slave-Agents [Buschmann et al., 1996] and enhance them with mobility mechanisms [Lange and Oshima, 1999]. A mobile agent can move from one system to another to perform specific operations, instead of continuously keeping the network "busy". Moreover, it often happens that SAs have to work together to execute common operations. For instance, in a coalition, the Canadian forces have to interact with non-government organizations as well as with armed forces of other countries. Therefore, SAs have to rely on communication [Labrou et al. 1999a] and coordination [Hamada, 1997] mechanisms to avoid conflicts and collaborate efficiently. When diverse SAs communicate, they have to understand each other. By establishing an ontology [Jones et al., 1998], a common terminology and semantic basis for the various SAs is offered. Hence, the risk of getting inconsistent information is reduced.

The paper is organized as follows. Section *1* proposes an overview of our theoretical, i.e. MASs, and practical, i.e. CCISs coalition, research project. Section *2* presents the degrees of interoperability in a coalition and an overview of the CCISs field. Section *3* describes the characteristics of the IC2MAS interoperability approach. Section *4*, briefly, reviews the related work. Section *5* gives insights on topics that are currently, tackled. Finally, Section *6* consists of concluding remarks.

2. Background

This section is divided into two parts. The first part identifies the degrees of interoperability in a coalition while the second part provides an overview of CCISs.

2.1 Interoperability Degrees in a Coalition

In a coalition, three degrees of interoperability are identified (adapted from [Au et al., 1999]). Basic interoperability, called interconnectivity, allows simple data transfer (with no semantic), whereas application-level integration enables applications (for example, CCISs) running in any environment to exchange services and perform computing, even if these applications were designed at different times by different persons. In a coalition, working at the application-level is not enough, particularly if the military forces aim at merging their operational processes. Therefore, a collaboration at the commandment level is required. In what follows, the three degrees of interoperability are summarized (cf. Figure 1):

- Physical interconnectivity: to guarantee basic communication, computing resources are first interconnected to exchange messages. This interconnectivity occurs at the physical level.
- Application integration: its main purpose is to carry out operations among different computing resources. Generally, these resources are distributed across networks and heterogeneous at different levels (hardware, software, and terminology).
- Commandment collaboration: it goes beyond application integration, by expanding military

operational processes to other structures. To this end, a collection of components, such as software agents gathered into multiagent systems, could be set up. These components collaborate more than just interoperate.

In Figure 1, the commandment level relies on the application level to achieve the coalition mission, for example humanitarian assistance. Furthermore, the commandment level interacts regularly with its headquarter. The purpose is to keep the headquarter informed about the progress of the mission. In order to assist the commandment level in its daily operations, the application level offers different types of services, such as data fusion and logistic. In fact, the application level is built on top of the physical level and hence, uses its computing resources. When the coalition's military forces have to collaborate, they go through a coordination process. Such a process could be entrusted to their respective MASs. In order to collaborate efficiently, military forces have to agree on how to invoke mutually their services. To this end, their respective applications have to be integrated.

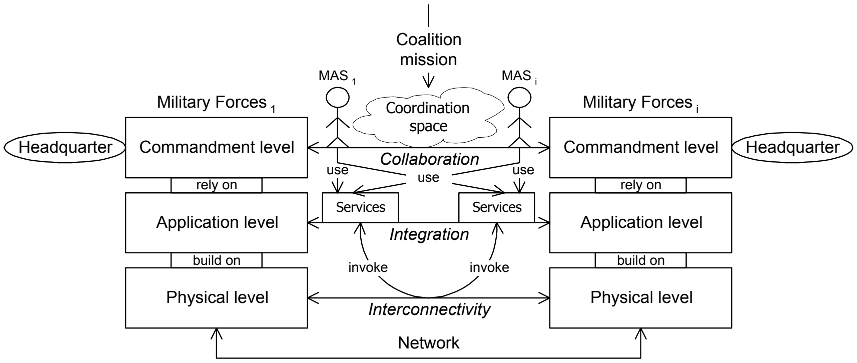


Figure 1 From interconnectivity to collaboration, through integration

2.2 An overview of CCISs

Nowadays, information technologies are an inherent part of the commanders' decision-making process. Particularly, CCISs help commanders to obtain a view of the tactical situation in which they are involved. In fact, a CCIS is used to gather information from different sensors, process this information, and suggest actions to be taken by the commander. Hence, CCISs are crucial and should meet demanding criteria in terms of reliability, efficiency, and fault-tolerance.

According to [Malerud et al., xxxx], a CCIS consists of a structure, functions, and tasks. The CCIS structure represents an assembly of facilities, arranged to meet the CCIS's objectives. To reach these objectives, the CCIS's functions are initiated in order to carry out the needed tasks. Tasks require the structure's facilities, in terms of personal, technical equipment, computing time, and so on. Figure 2 presents a simplified architecture of a CCIS. Several types of functions exist within the CCIS, ranging from planning and weather forecast to data fusion. These functions are offered to users and are built on top of a support structure in terms of hardware and software resources. Furthermore, some of these functions receive messages from the external environment, e.g. remote sensors, through a communication module. Currently, multiple definition languages of messages, e.g. USMTF, are available. These languages allow formatting messages in order to be automatically parsed by appropriate engines of the different functions. Unfortunately, such languages cannot be used in the achievement of interoperable CCISs. These languages' structures are too rigid and do not have semantics.

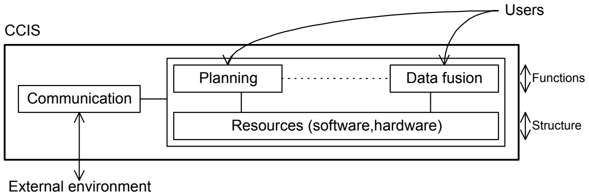


Figure 2 CCIS Simplified Architecture

As CCISs are getting larger and more complex, their interoperability and hence collaboration, in a coalition context for example, are becoming a central concern for military users and CCISs' designers. Therefore, the IC2MAS project aims at handling this concern, by providing 1) users with services that will free them from worrying about the characteristics of the interconnected CCISs; and 2) designers with approaches based on advanced technologies, such as MASs.

3. Presentation of the IC2MAS Approach

This section presents the IC2MAS approach. First, major requirements of a coalition are described. Next, IC2MAS architecture and types of SAs are presented. Finally, IC2MAS operating is detailed.

3.1 Coalition's Requirements

In the IC2MAS project, the running scenario corresponds to a coalition that is set up by different countries for different purposes: international humanitarian-assistance, peace support operations, etc. The coalition scenario is appropriate for several reasons:

- People from different countries, at different locations, and at different moments contribute to the definition of the same operations, for instance deploying troops in a combat zone. However, these people do not use the same communication language and do not manage the same types of resources that vary from high to low technologies. It happens that certain countries are well equipped than others.
- At diverse hierarchical levels, different people take decisions during the performance of operations. It happens that a decision is based on an information that is not well understood by all people. Moreover, it happens that a decision requires the interaction of diverse CCISs that could be distributed and heterogeneous.
- At the theater of operations, it is complex to provide and maintain a high level of assistance to military users. For example, it is not possible to afford to each combat unit an expert in PC software, an expert in Unix software, etc. Moreover, it is not possible for a military user to be aware of the characteristics of the different CCISs of the coalition.

Major requirements to coalition support constitute a framework that identifies what types of information could be exchanged, what types of operations could be delegated, and what types of communication approaches could be used. In what follows, a research avenue is associated with each requirement.

- Requirement: What types of information could be exchanged?
  Research Avenue: Ontology.
  Definition: an ontology is a means to express and exchange information that is understood by all the participants of the coalition. Moreover, to be used efficiently an ontology requires a language to be represented, e.g. KIF, and a language to be communicated, e.g. KQML.
- Requirement: What types of operations could be delegated?
  Research Avenue: SAs integrated into MASs.
  Definition: a SA is an autonomous, goal-oriented entity that has the ability to assist users in performing their tasks, to collaborate with other agents (software or human) to jointly solve problems, and to answer users' needs. Furthermore, a collection of SAs can be gathered into a MAS architecture. As stated in [Labrou et al. 1999ba], communities of agents are much powerful than any individual agent.
- Requirement: What communication approaches could be used?
  Research Avenue: Remote/Local communication.
  Definition: Communication between distributed components, for example SAs, could occur either remotely or locally. In the latter case, the components have to move to a common workplace.

3.2 IC2MAS Architecture

In the literature, different approaches that deal with the problem of interoperable systems can be found, among them Infosleuth [Bayardo et al., 1997], TSIMMIS [Chawathe et al. 199], SIMS [Knoblock et al., 1997], and SIGAL [Maamar et al., 1999]. All these approaches agree on the use of SAs, as a means to develop such systems and have several elements in common, such as all the SAs are static. Therefore, these SAs do not have the opportunity to move to distant systems. Moreover, all these approaches assume that the network infrastructure is fully reliable and has unlimited bandwidth for information transmission.

Based on these different approaches and the coalition's requirements, we proposed an IC2MAS architecture to the coalition (cf. Figure 3). Multiple MASs form the backbone of this architecture and interact remotely as well as locally. In addition, these MASs collaborate through a facility called Advertisement Infrastructure. It is managed by an agent and contains a Bulletin Board and a Repository of Active-Agents. Currently, we are aware that the Advertisement Infrastructure could be considered as a bottleneck. However and in the mid-term, this infrastructure could be duplicated and spread across networks.

In the IC2MAS architecture, MASs integrate different types of SAs: Interface-Agents assisting users, CCIS-Agents invoking CCISs' functions and satisfying users' needs, Resolution-Agents, also, satisfying users' needs, Control-Agents managing MASs, and finally, a Supervisor-Agent managing the Advertisement Infrastructure. In the IC2MAS environment, the Resolution-Agent is able to create Slave-Agents and transmit them either to the Advertisement Infrastructure or to other distant MASs. Slave-Agents carry out operations on behalf of Resolution-Agents. Slave-Agents' creation process complies with the Supervisor-Worker pattern as defined in [Fischmeister and Lugmayr, 1999]. In the next sections, agents' functionalities are depicted.

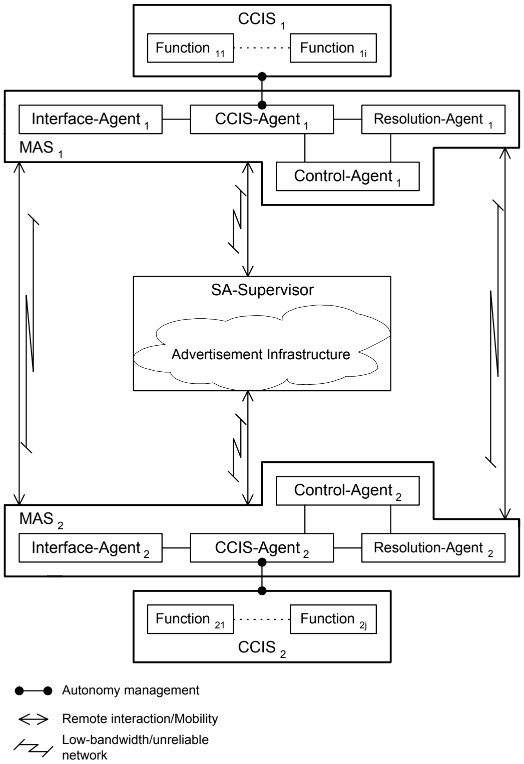


Figure 3 IC2MAS Architecture for coalition support

3.3 Software Agents and Advertisement Infrastructure

Different types of SAs exist in the IC2MAS architecture. These SAs belong to different MASs and collaborate through the Advertisement-Infrastructure facility. In what follows, certain agents’ internal-modules are detailed.

Interface-Agent - By analogy to Interface-Agents of [Maamar et al., 1999, Sycara et al., 1996], the IC2MAS's Interface-Agent assists users in formulating their needs, maps these needs into requests, forwards these requests to the CCIS-Agent in order to be processed, and provides users with answers obtained from the CCIS-Agent.

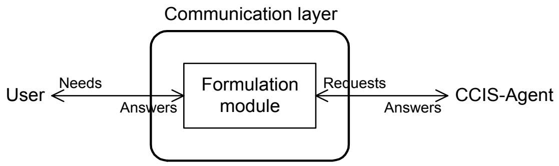


Figure 4 Interface-Agent modules

The Interface-Agent consists of one module, called formulation that is encapsulated into a communication layer (cf. Figure 4). The formulation module takes as inputs users' needs and CCIS-Agent's answers and provides as outputs requests to CCIS-Agents and answers to users. In the IC2MAS environment, users describe their needs according to the concepts that are understood by Interface-Agents (cf. Section 5.1).

CCIS-Agent - By analogy to Resolution-Agents of [Maamar et al., 1999] and Task-Agents of [Sycara et al., 1996], the IC2MAS's CCIS-Agent processes users' requests, only if these requests need the involvement of the CCIS of this particular CCIS-Agent. These requests are transmitted by the Interface-Agent. In addition, and by analogy to Knowledge-Agents of [Maamar et al., 1999], the IC2MAS's CCIS-Agent acts on CCIS behalf and hence, maintains its autonomy towards the coalition. To achieve this autonomy, the CCIS-Agent advertises, through its services (currently, the services do not have constraints, e.g. cost), the functions its CCIS performs. Here, the term service denotes a computing procedure, for example requesting the CCIS's weather-forecast function. In the IC2MAS environment, a CCIS-Agent advertises its services, by posting notes on the Bulletin Board of the Advertisement Infrastructure. In fact, the CCIS-Agent sends remote requests to the Supervisor-Agent. Before posting notes, the Supervisor-Agent checks the CCIS-Agent's security level to authenticate this CCIS-Agent's requests and identify the services it is authorized to advertise.

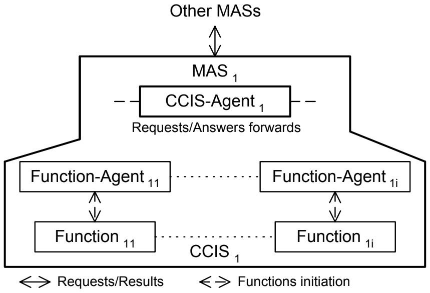


Figure 5 Function-Agents at the MAS level

A CCIS offers different functions that vary from data fusion and weather forecast to planning (cf. Section 2.2). Based on these functions and the complex nature of CCISs, for instance a planning function could be a distributed-object client/server application running on top of an Object Request Broker middleware, new types of SAs, called Function-Agents, are introduced in the IC2MAS architecture, and particularly at the MAS level. Each Function-Agent is associated with a CCIS's function. As a result, a CCIS-Agent manages a group of Function-Agents that evolves under its supervision (cf. Figure 5). For instance, a request to the planning function of a CCIS is initially, sent to the CCIS-Agent that forwards this request to the appropriate Function-Agent. Hence, a Function-Agent knows the protocols through which a function of a CCIS accepts requests and provides back results. IC2MAS's Function-Agents are similar to Information-Agents of [Sycara et al., 1996].

Figure 6 presents CCIS-Agents' and Function-Agents' modules. As the Interface-Agent, a communication layer encapsulates both agents' modules. The CCIS-Agent consists of two modules: definition and pre-processing. The IC2MAS administrator uses the definition module. He specifies the services to be advertised by the CCIS-Agent. The pre-processing module identifies whether or not the CCIS of a CCIS-Agent could satisfy users' requests. If not, these requests are transmitted to the Resolution-Agent. The pre-processing module relies on an information source, called CCIS capabilities. Moreover, the administrator updates this information source with the services it has specified. The Function-Agent consists of two modules: processing and monitoring. The processing module receives requests from the CCIS-Agent and performs them against the CCIS's function. The monitoring module monitors the modifications that could occur at the CCIS's functions level. These modifications have to be notified to the CCIS-Agent's definition-module.

Resolution-Agent - By analogy to Resolution-Agents of [Maamar et al., 1999] and Task-Agents of [Sycara et al., 1996], the IC2MAS's Resolution-Agent processes users' requests, only if these requests are transmitted by the CCIS-Agent and need the involvement of several CCISs to be completed. In fact, the resolution process requires that the Resolution-Agent collaborates with the CCIS-Agents of other MASs, including or not the CCIS-Agent of this Resolution-Agent's MAS.

At IC2MAS start-up time, the Resolution-Agent creates a Slave-Agent, called Help-Agent, and sends it to the Advertisement Infrastructure. As soon as the Help-Agent arrives, the Supervisor-Agent checks it. Next, the Help-Agent waits for the Resolution-Agent's queries about the services to look for the Bulletin Board[1].

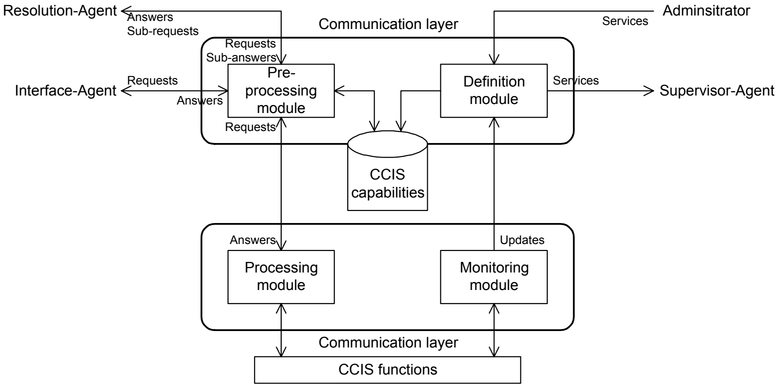


Figure 6 CCIS-Agent and Function-Agent modules

In order to identify the CCIS-Agents that are required to satisfy users' requests, the Resolution-Agent sends remote queries to the Help-Agent. This agent browses the Bulletin Board, identifies appropriate CCIS-Agents through their offered services, and finally informs remotely its Resolution-Agent parent. Next, the Resolution-Agent designs the procedure needed to the performance of the user's request. Generally, such a procedure is called a route or an itinerary. Then, the Resolution-Agent creates another Slave-Agent, called Route-Agent, and assigns to it this procedure. The Route-Agent may require either interacting remotely with the CCIS-Agents of the other MASs or migrating to the MASs and meet locally their CCIS-Agents. A decision about a remote request or mobility is based on the network status and the number of the CCISs required satisfying users' requests[2]. As CCIS-Agents, a security level is also associated with Slave-Agents. This security level is used to check Slave-Agents entering the Advertisement-Infrastructure as well as the different MASs.

The Resolution-Agent consists of two modules, called slave and pre-processing (cf. Figure 7). Both of them are encapsulated into a communication layer. The slave module creates Slave-Agents, namely Help-Agent and Route-Agent. The pre-processing module designs the procedure that is used to perform users' requests. This procedure is forwarded to the Route-Agent's performance module. This agent carries out these requests, according to the CCISs that have been identified by the Help-Agent's browsing module.

[1] A Help-Agent could regularly consult the Bulletin Board in order to inform its Resolution-Agent about the notes that could interest it.

[2] It is stated in [Bredin et al., 1999] that the value of mobile-agent system depends on both the number of host sites that an agent might migrate to as well as the number of other agents with which an agent may interact.

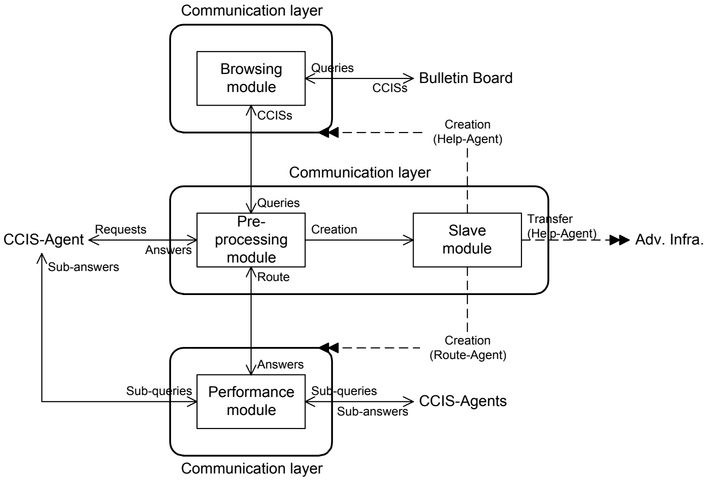


Figure 7 Resolution-Agent modules (including Help-Agent and Route-Agent)

Control-Agent - In an environment consisting of mobile agents, mobility operations consist of shipping the agents through the net to other distant systems, authenticating these agents as soon as they arrive, and finally installing these agents to resume their operations. In the IC2MAS environment, the Control-Agent of the MAS is in charge of all these operations. For instance, when a Help-Agent moves, it first interacts with the Control-Agent in order to be shipped to the desired MAS. Furthermore, Control-Agents maintain the coherence of their MASs by keeping track of the Route-Agents entering and leaving these MASs.

Supervisor-Agent - A Supervisor-Agent is in charge of several operations. It manages the Advertisement Infrastructure by receiving CCIS-Agents' advertisements, sets up a security policy in order to monitor the Help-Agents accessing this infrastructure, and finally, installs Help-Agents to resume their operations in this infrastructure.

In the IC2MAS environment, the Supervisor-Agent uses the Repository of Active-Agents to register all the Help-Agents and CCIS-Agents that have respectively got an agreement to enter the Advertisement Infrastructure and advertise their services. The Repository of Active-Agents is, also, updated when Resolution-Agents decide to remove their Help-Agents from the Advertisement Infrastructure.

Advertisement Infrastructure - In a coalition context, CCISs are spread across networks and generally rely on low-bandwidth and/or unreliable channels for communications. Moreover, a military user may use his VHF Combat Net Radio to send and request information. This military usually relies on mobile devices, such as portable computers, that are only intermittently connected to networks. In the IC2MAS environment, instead of overloading the network, Help-Agents migrate to the Advertisement Infrastructure and browse locally the Bulletin Board, looking for appropriate CCISs.

3.4 IC2MAS operating

Based on the characteristics of the IC2MAS architecture and the types of SAs this architecture integrates, we proposed four stages to handle the IC2MAS operating (cf. Figure 8): Initialization, Advertisement, Operation, and Maintenance. In what follows, the features of each stage are described. Note that Initialization and Advertisement stages are transparent to users.

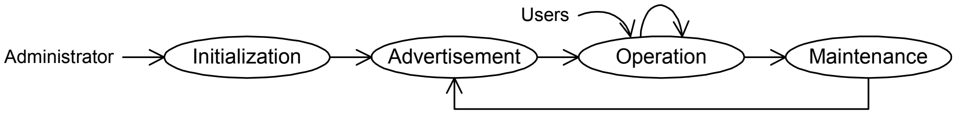


Figure 8 Stages of the IC2MAS operating

Initialization Stage - This stage is characterized by the following operations:

- The Advertisement Infrastructure and its components, i.e. Supervisor-Agent, Bulletin Board, and Repository of Active-Agents, are set up and started-up. Thus, the Supervisor-Agent initializes the Bulletin Board and the Repository. Further, this agent initiates the security policy that manages agents' accesses to the Advertisement Infrastructure.
- MASs are set up and associated with their respective CCISs. For instance, the Resolution-Agent creates its Help-Agent and sends it to the Advertisement Infrastructure (cf. Figure 9). As soon as it arrives, the Help-Agent is checked, registered, and finally, installed.

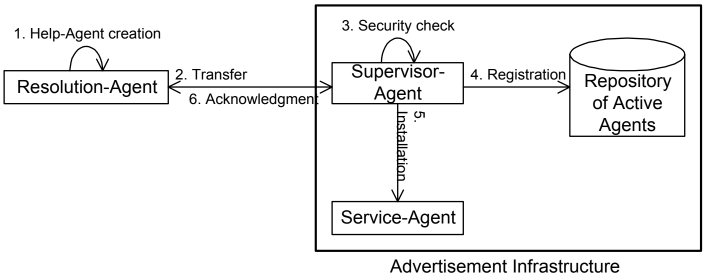


Figure 9 Help-Agent in the Advertisement Infrastructure

In what follows, we assume that, before leaving and entering MASs, Slave-Agents, namely Help-Agents and Route-Agents, interact with Control-Agents for security, shipping, and installation purposes.

Advertisement Stage - Once the initialization stage is done, CCIS-Agents have to advertise their services at the Advertisement-Infrastructure level. As stated in Section 3.3, CCIS-Agents send remote requests to the Supervisor-Agent of the Advertisement Infrastructure (cf. Figure 10).

According to the security level of this CCIS-Agent and the security policy of the Advertisement Infrastructure, the Supervisor-Agent decides if this CCIS-Agent is authorized to advertise and what types of services. In the positive case, the Supervisor-Agent processes the CCIS-Agent's request by posting the services it offers on the Bulletin Board. Furthermore, the Supervisor-Agent registers the fact that this CCIS-Agent has notes on the Bulletin Board. At the end, the Supervisor-Agent acknowledges the CCIS-Agent about the success (or failure) of the operation. We assume that CCIS-Agents send only one request in order to advertise all the services they offer. Moreover, we assume that other requests will follow that either update or withdraw the advertised services.

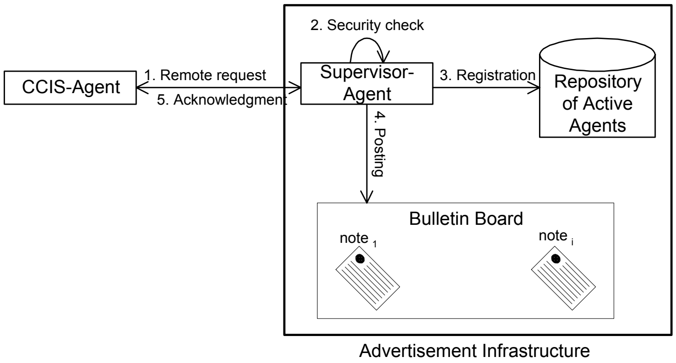


Figure 10 Services advertisement in the Bulletin Board

Operation Stage - Once the advertisement stage is done, the IC2MAS environment is ready to be operated. The operation stage of IC2MAS is summarized by two situations (cf. Figure 11):

- Only the user's CCIS is required: the CCIS-Agent is in charge of handling this situation (cf. Figure 11-a).
- Several CCISs, including or not the user's CCIS, are required: the Resolution-Agent is in charge of handling this situation (cf. Figure 11-b).

In what follows, numbers in parenthesis correspond to numbers in Figure 11 and illustrate operations chronology.

When a user wants to satisfy his needs (0), he interacts with the Interface-Agent of his MAS. Next, his needs are mapped into a request transmitted to the CCIS-Agent (1). This agent is in charge of deciding whether this user's CCIS contains the appropriate functions to process its request (cf. Figure 6, pre-processing module). Once such a decision is obtained (2), two situations exist and are identified in Figure 11 with letters a and b.

In Situation a, the CCIS-Agent forwards the user's request to the appropriate Function-Agent (3.a) of the user's CCIS. This Function-Agent initiates the CCIS's function and provides the results it obtained to the CCIS-Agent (4.a). Finally, results are sent to the user through the Interface-Agent (5.a, 6.a).

In Situation b, other CCISs, including or not the user's CCIS, are required to satisfy the user's request. These CCISs are identified using the notes of the Bulletin Board of the Advertisement Infrastructure. First, the CCIS-Agent forwards the user's request to the Resolution-Agent (3.b). Next, the Resolution-Agent interacts remotely with its Help-Agent, about the CCISs to identify (4.b). Once the Help-Agent has completed its operations (5.b), it sends to the Resolution-Agent the CCIS-Agents with whom it is going to interact (6.b). Once this information arrives, the Resolution-Agent starts to design its itinerary according to the number of the pertinent CCISs and the network status (7.b). To perform this itinerary, the Resolution-Agent creates a Route-Agent and assigns to this agent the designed itinerary (8.b). To clarify things, hereafter is an example illustrating this itinerary. In Figure 11, the itinerary indicates that the Route-Agent first has to move to a MAS (9.b), for instance MAS_2. Next, the Route-Agent interacts locally with the CCIS-Agent of this MAS (10.b). Furthermore, to complete its operations, the itinerary mentions that the Route-Agent has to remotely interact with other CCIS-Agents, for instance CCIS-$Agent_3$ of MAS_3. Then, the Route-Agent sends a request (11.b) and waits for the results from CCIS-$Agent_3$ (12.b). At the end, the Route-Agent goes back to its original MAS (13.b) and interacts with Resolution-Agent parent. Finally, the Resolution-Agent sends the results obtained from its Route-Agent to the user through the CCIS-Agent and the Interface-Agent (14.b, 15.b, 16.b).

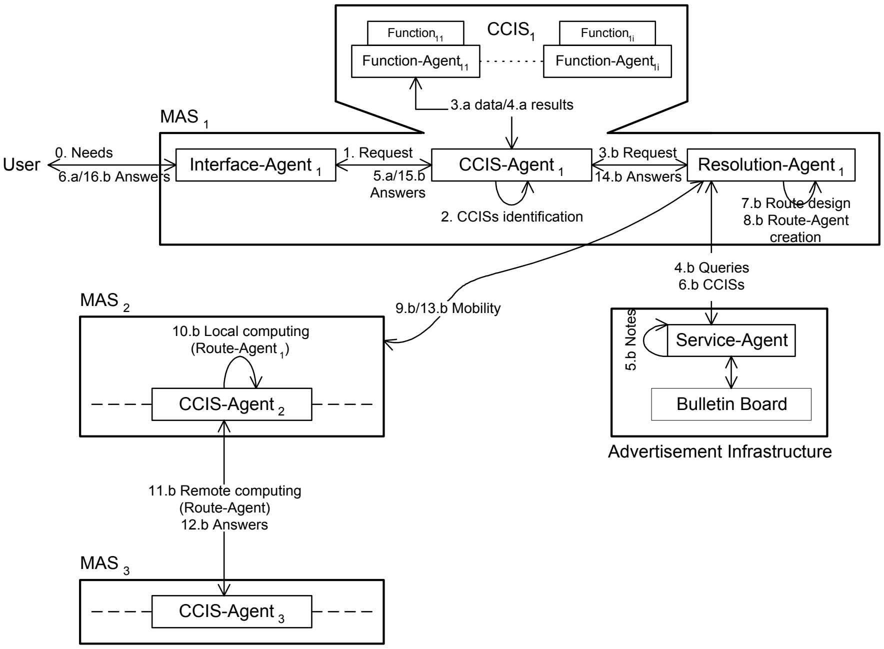


Figure 11 User's request satisfaction

Maintenance Stage - The IC2MAS environment is an open system. Indeed, a new CCIS could be integrated, another CCIS could be removed, etc. Therefore, the purpose of the maintenance stage is to take into account the situations that may have an impact on the architecture of the IC2MAS environment as well as on its operating. Several situations have been identified. In this paper, we briefly present two of them:

- It happens that a CCIS adapts its structural as well as functional characteristics, for example by adding a new function or by upgrading the version of a function's database management

system. Therefore, the CCIS-Agent has to be adapted either by adding new services to its capabilities or by updating its services. Further, the CCIS-Agent has to interact with the Advertisement Infrastructure.
- It happens that the Supervisor-Agent cleans up the Bulletin Board of the Advertisement Infrastructure, because of for example a new security policy. Hence, CCIS-Agents have to advertise their services from the beginning.

4. Related Work

This section summarizes the main characteristics of the IC2MAS environment with respect to other similar works. There exist different research projects in the field of systems interoperability [Bayardo et al. 1997] [Chawathe et al. 1994] [Genesereth and Ketchpel, 1994] [Hsu, 1996] [Knoblock et al., 1997] [Maamar et al., 1999] [Papazoglou et al., 1992]. All these projects have the same concerns, namely:

- Maintain the autonomy and independence of the systems to be integrated in an interoperable environment. In the IC2MAS environment, each CCIS has been associated with a CCIS-Agent that acts on its behalf.
- Reduce the informational disparities between the integrated systems. In the IC2MAS environment, the definition of an ontology is, currently, tackled (cf. Section 5.1).
- Help users satisfy their needs. In the IC2MAS environment, each MAS integrates an Interface-Agent that assists users.

However, all the projects cited above assume that the network infrastructure is fully reliable and has unlimited bandwidth for information transmission. In a coalition, this is not the case. In the IC2MAS environment, network concern has been considered, for instance by enhancing certain agents with mobility mechanisms and giving these agents the ability to decide whether local computing after a move is preferable than remote computing. Furthermore, security issues have been considered in the IC2MAS environment, by suggesting a security policy to manage the Advertisement Infrastructure and a security level to identify agents. Additional security elements could be suggested, for instance identifying services with authorization levels and users with use levels.

5. IC2MAS's Current Efforts

This section gives insights on topics that are currently, tackled, in the IC2MAS environment. Among these topics, we describe, briefly, the ontological disparities. Ontology is one of the main issues to be addressed in the design of an interoperable environment for heterogeneous systems. We consider an ontology as a means to represent and exchange information that are understood by all participants.

In a coalition context, each country has its own standards. Therefore, each military user specifies his needs, in term of information requests, and his CCIS's capabilities, in term of functions}, using these standards. Therefore, the need to define two types of specification languages is raised in the IC2MAS interoperability approach. The first type is a specification language for users' needs while the second type is a specification language for CCISs' functions. Both of these languages have to be based on two different ontologies: a user-oriented ontology and a CCIS-oriented ontology. Furthermore, because of the coalition context, the user-oriented ontology has to be adapted in order to take into account the individual differences, for example diversity of cultures that exist between the coalition's participants. To handle these characteristics, we intend to propose a user-oriented ontology that is "versioned" (certain authors talk about ontology sharing). Hence, only one user-oriented ontology is defined at the conceptual level but different versions of this ontology are defined at users level.

6. Conclusion

In this paper, we presented the major characteristics of the IC2MAS interoperability approach that uses MASs in the design of collaborative environments for distributed and heterogeneous CCISs. The coalition context is the running scenario. In this approach, MASs and their SAs are able to fulfill different operations, from users' needs specification to CCISs' functions initiation. Eight types of SAs exist in the architecture proposed for coalition support (Interface-Agent, CCIS-Agent, Resolution-Agent, Control-Agent, Function-Agent, Supervisor-Agent, Help-Agent, Route-Agent) while four stages describe this architecture operating (Initialization, Advertisement, Operation, Maintenance). Whereas MASs appear to offer benefits to coalition support, we must be aware of their limitations. MASs must allow a large degree of human interaction. Therefore, it is unrealistic to expect to be able to provide a "fully" automated coalition support. A whole set of negotiations, dialogues, coordination and communication between participants, groups of participants, and systems are involved.

References

[Au, 1999] T.A. Au. A primer of CORBA: A framework for distribution applications in defence. Technical report, DSTO-GD-0192, DSTO Electronics and Surveillance Research Lab, Salisbury, Australia, 1999.

[Bayardo et al., 1997] R. Bayardo, W. Bohrer, R. Brice, A. Cichocki, G. Fowler, A. Helai, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, and D. Woelk. Infosleuth: Semantic integration of information in open and dynamic environments. In *Proc. of the 1997 ACM International Conference on the Management of Data (SIGMOD)*, Tucson, Arizona, 1997.

[Babin et al., 1994] G. Babin, C. Hsu, and Z. Maamar. A distributed metadatabase architecture for an enterprise scalable, adaptive integration environment. Technical Report 37-94-424, DSES, Rensselaer Polytechnic Institute, Troy, N.Y., USA, 1994.

[Bredin et al., 1999] J. Bredin, D. Kotz, and D. Rus. Mobile-agent planning in a market-oriented environment. Technical report, Dartmouth Technical Report PCS-TR99-345, 1999.

[Buschmann et al., 1996] F. Buschmann, R. Meunier, H. Rohnet, P. Sommerland, and M. Stal. *Pattern-Oriented Software Architecture*. John Wiley and Sons Ltd., 1996.

[Chawathe et al. 199] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The TSIMMIS project: Integration of heterogeneous information sources. In *Proc. of the 10th IPSJ*, Tokyo, Japan, 1994.

[Fischmeister and Lugmayr, 1999] S. Fischmeister and W. Lugmayr. The supervisor-worker pattern. In *Proc. of Pattern Languages of Programs (PLoP'99)*, Kubova Hut, Czech Republic, 1999.

[Ghenniwa, 1997] H. Ghenniwa. *Coordination in Cooperative Distributed Systems*. PhD thesis, Systems Design Engineering Department, University of Waterloo, Ontario, Canada, 1997.

[Green et al., 1997] S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Somers, and R. Evans. Software agents: A review. Technical report, Trinity College Dublin and Broadcom Éireann Research Ltd., May 1997.

[Genesereth and Ketchpel, 1994] M.R. Genesereth and A.P. Ketchpel. Software agents. *Communication of the ACM*, 37(7):48--53, July 1994.

[Hsu, 1996] C. Hsu. *Enterprise Integration and Modeling - the Metadatabase Approach*. Kluwer Academic Publisher, Boston, Mass., USA, 1996.

[Jones et al., 1998] D. Jones, T. Bench-Capon, and P. Visser. Methodologies for ontology development. In *the proceedings of IT&KNOWS - Information Technology and Knowledge Systems, 15th IFIP World Computer Congress*. Vienna (Austria) and Budapest (Bulgaria), 1998.

[Knoblock et al., 1997] C.A. Knoblock, Y. Arens, and C.N. Hsu. Cooperating agents for information retrieval. In *Second International Conference on Cooperative Information Systems*, Toronto, Canada, 1994.

[Labrou et al. 1999a] Y. Labrou, T. Finin, and Y. Peng. Agent communication languages: The current landscape. *IEEE Intelligent Systems*, 14(2):45--52, March/April 1999.

[Labrou et al. 1999b] Y. Labrou, T. Finin, and Y. Peng. The interoperability problem: Bringing together mobile agents and agent communication languages. In *Proceedings of the Hawaii International Conference On System Sciences*, Maui, Hawaii, January 1999.

[Lange and Oshima, 1999] D. Lange and M. Oshima. Dispatch your agents; shut off your machine. *Communication of the ACM*, 42(3):88--89, March 1999.

[Malerud et al., xxxx] S. Malerud, Feet E.H., and U. Thorsen. A method for analysing command and control systems. Norwegian Defence Research Establishment (FFI) N-220 Kjeller, Norway.

[Maamar et al., 1999] Z. Maamar, B. Moulin, and Y. Bédard. Software agent-oriented frameworks for the interoperability of georeferenced digital libraries on the world wide web: The SIGAL project. In R. Fegeas M.F. Goodchild, M.J. Egenhofer and C.A. Kottman, editors, *Interoperating Geographic Information Systems*, pages 335--354. Boston: Kluwer Academic Publishers, 1999.

[Papazoglou et al., 1992] M.P. Papazoglou, Laufmann S.C., and T.K. Sellis. An organizational framework for cooperating intelligent information systems. *International Journal of Intelligent and Cooperative Information Systems*, 1(1):169--202, 1992.

[Sycara et al. 1996] K. Sycara, K. Decker, A. Pannu, M. Williamson, and D. Zeng. Distributed intelligent agents. *IEEE Expert-Intelligent Systems and Their Applications*, 11(6):36--45, July 1996.

Influence in MultiAgent Systems – Application to Coalitions

Zakaria Maamar
Software Agents Research Group$_{@ZU}$
College of Information Systems
Zayed University
PO Box 19282, Dubai, U.A.E
zakaria.maamar@zu.ac.ae

Paul Labbé
System of Systems Section
Defence Research Establishment Valcartier
2459 Pie-XI Blvd. North Val-Bélair
(Québec), Canada, G3J 1X5
paul.labbe@drev.dnd.ca

Wathiq Mansoor
Software Agents Research Group$_{@ZU}$
College of Information Systems
Zayed University
PO Box 19282, Dubai, U.A.E
wathiq.mansoor@zu.ac.ae

Emad Bataineh
College of Information Systems
Zayed University
PO Box 19282, Dubai, U.A.E
emad.bataineh@zu.ac.ae

Abstract. This paper presents a framework to deal with influence in multiagent systems. Influence is defined as the impact that a participant could have on another participant, known as target. Influence could be either positive or negative, according to how this target assesses the outcomes of the operations this participant has carried out. The presented framework could be viewed from two different perspectives: knowledge perspective with goal and belief as main components and organization perspective with task and resource as main components.

0. Paper structure

This paper is structured as follows. Section 1 presents an overview of influence and why it is relevant to study it in multiagent systems. Section 2 suggests definitions related to influence. Section 3 analyses influence at four levels, namely goal, belief, task, and resource. Section 4 illustrates the use of influence in the military domain. Finally, Section 5 consists of concluding remarks.

1. Overview

The purpose of this paper is to discuss the role that influence could play in understanding and predicting software-agents' behavior. Influence investigates the causes of human modification – whether that modification is a behavior, an attitude, or a belief [1]. Usually, influence is employed by a participant upon a target and relies on the social interactions that exist between them [3]. The participant is the one who influences whereas the target is the one who is influenced. Considering influence in MultiAgent Systems (MASs) has driven us to work at four levels, namely goal, belief, task, and resource. Goal and belief levels are seen from a knowledge perspective while task and resource levels are seen from an organization perspective. These four levels could be part of the agents' mental-model; this model is subject to modifications by the agent that is influenced (cf. Figure 1). These modifications depend on the outcomes of the operations undertaken by the agent that influences. We recall that goal, belief, task, and resource are connected to each other. An agent exhibits a goal-oriented behavior. Often, plans implement such a behavior. To achieve a goal, the agent selects the appropriate tasks on the basis of the beliefs it has in its mental model. Finally, tasks require resources in order to be completed. In this paper, **G**oal, **B**elief, **T**ask, and **R**esource constitute the GBTR framework.

In real life, the environment in which we live influences our behaviors in different ways. For example, we adapt our attitudes, reactions, and expectations. Influence could be either positive, i.e. “good”, or negative, i.e. “bad”. In the rest of this paper, we discuss how Software Agents (SAs) could be used for simulating influence. SAs are autonomous entities having the abilities to collaborate with other SAs to jointly solve different problems [2]. Usually, these problems are inherently distributed and heterogeneous. We intend to apply SAs as well as influence to military scenarios, where for instance several combat units have to cooperate regardless of the fact they are spread across a battlefield. These units influence each other, particularly if they are committed to the same operation. If a combat unit is defeated, a friendly unit should assess the consequences of this defeat. In fact, this friendly unit should assess how it will be influenced. For instance, this unit could expect attacks from the hostile troops. Interesting are situations in which a combat unit does not communicate with other units to avoid messages interception, i.e. “intelligence” surveillance. Therefore, such units are not able to assess how they will be influenced. Decisions, regarding the following operations to undertake, should be made under uncertainty. Uncertainty is defined as the difference between the knowledge that is required to accomplish a mission and the decision a decision-maker has at that time. Hence, uncertainty is inversely proportional to the decision-maker’s belief of understanding of the current situation [[5].

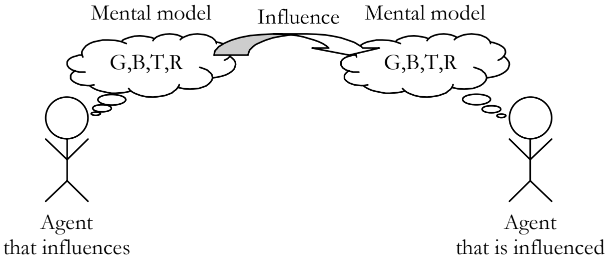


Figure 1 Influence impact on the mental model

[4] views influence as a cognitive process by which an agent acquires new knowledge. This process, known as social learning, takes place between an agent that is exposed to another. Both agents are located in a common environment; this means that they are aware of each other, for instance via observation. According to the same author, social learning could happen either by facilitation or by imitation. In the first situation, a learning agent updates its knowledge by perceiving the relationship between another agent and the physical or social environment that interests this learning agent. In the second situation, imitation is defined as a process in which a learning agent is ruled by the knowledge it has on the agent it is currently observing.

2. Definitions

In the GBTR framework, influence occurs at goal, belief, task, and resource levels. In what follows, a short definition is proposed for each level.

- **What does goal influence stand for?** Here, the agent’s goal-hierarchy is adapted, after the insertion of a new goal in this hierarchy. Insertion involves dealing with this new goal, by identifying who is going to achieve it? How to achieve it in term of planning? When to achieve it? And what does it require in term of resources?
- **What does belief influence stand for?** Here, the agent’s belief-repository is updated, after the insertion of a new belief in this repository. Consistency between the different beliefs should be ensured; an agent cannot manipulate contradictory beliefs.
- **What does task influence stand for?** Here, the agent’s task-repository is updated, after either the insertion of a new task in this repository or the modification of the characteristics of a specific task of this repository. In the insertion situation, the agent should find who is going to perform this new task? How to perform it? When to perform it? And what does it need in term of resources? In addition, the execution chronology of tasks should be dealt with since a new task has been introduced. In the modification situation, a task could be changed regarding for example who is going to perform it or when it is going to be performed.
- **What does resource influence stand for?** Here, the agent’s resource repository is updated. This agent could either receive additional resources or lose some of its resources momentarily. In the first situation, the agent uses the resources it gets in order to carry out its goals. In the second

situation, the agent outsources its resources.

These four types of influence require from the agent that is influenced to possess two modules, known as awareness and assessment. The awareness module is a means to identify the agents that are part of the agent's environment and that could influence this agent. The assessment module is a means to identify how the agent is influenced either positively or negatively and at which level, i.e. goal, belief, task, or resource. The assessment module relies on the awareness module. In what follows, we describe how both modules should work from the perspective of the agent that is influenced (cf. Figure 2).

- The awareness module has the following working cycle:
    a. The agent identifies who is located within its environment.
    b. After knowing its acquaintances, the agent establishes what kind of relationships it has with these acquaintances. Examples of relationships could be friendly and hostile.
    c. Finally, the agent makes out the operations its acquaintances have performed.
- The assessment has the following working cycle:
    a. The agent needs to know if the agents it has identified in Step a. of the awareness cycle are either new or it has already encountered them.
    b. Then, the agent investigates if the relationships it established in Step b. of the awareness cycle are valid.
    c. Finally, the agent analyses the outcomes of the operations these agents have undertaken. This analysis permits this agent to adapt its behavior on the basis of how it is influenced, either positively or negatively.

We recall that the awareness and assessment modules work in an interleaved arrangement. In fact, each step of the awareness cycle is followed by a step of the assessment cycle and *vice-versa*.

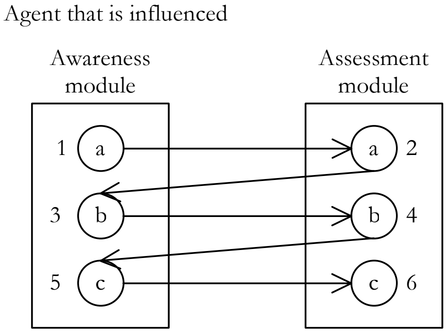


Figure 2 Awareness and assessment interleaving arrangement

We view the GBTR framework as a means to represent the agent adaptability in an open environment. As this environment changes, agents are affected and consequently must be ready to act. For illustration purposes, assigning a new goal to an agent requires either designing new plans or repairing the previous plans. In an open environment, classical long plans are not always successfully executed due to unpredictable changes in the world. A change in the world can make plans invalid.

3. Analysis

Influence depends on the relationships that exist between agents. Such relationships could be of type "supervise", "supervised-by", or "peer-to-peer". Both "supervise" and "supervised-by" define who reports to whom? "Who does what" question is also important when dealing with influence. This question defines the origin of influence, i.e. the operations that are the cause of this influence.

In the GBTR framework, an agent could influence another agent at goal, belief, task, and resource levels. Since influence could be either positive or negative, the following combinations are obtained (cf. Table 1). We assume that agent$_1$ influences agent$_2$. In a negative influence, the agent that is influenced should proceed as follows: suspend its operations that are in progress, carry out the operations of the agent that influences, and finally resume its operations.

Table 1 Types of influences between agents

Influence	Type		Description
Goal	Positive	(+)	Agent$_1$ generates a new goal that will support agent$_2$ in achieving its goals. Agent$_1$ will be in charge of satisfying this new goal for the benefit of agent$_2$. *Facilitate relationship between goals.*
	Negative	(-)	Agent$_1$ generates a new goal that will delay agent$_2$ in achieving its goals. In fact, agent$_2$ will be in charge of satisfying this goal for the benefit of agent$_1$. *Hinder relationship between goals.*
Belief	Positive	(+)	Agent$_1$ produces a new belief that will affirm some of agent$_2$'s beliefs. *Affirm relationship between beliefs.*
	Negative	(-)	Agent$_1$ produces a new belief that will contradict some of agent$_2$'s beliefs. Agent$_2$ should amend its beliefs. *Contradict relationship between beliefs.*
Task	Positive	(+)	Agent$_1$ carries out some of agent$_2$'s tasks on its behalf. *Conduct relationship between agents and tasks.*
	Negative	(-)	Agent$_1$ entrusts some of its tasks to agent$_2$, in addition to the tasks agent$_2$ is already in charge. *Work for relationship between agents and tasks.*
Resource	Positive	(+)	Agent$_1$ offers some of its resources to agent$_2$. This helps agent$_2$ to carry out its tasks and in the same time to achieve its goals. *Offer relationship between agents and resources.*
	Negative	(-)	Agent$_1$ takes over some of agent$_2$'s resources. Agent$_2$ could lack resources to carry out its tasks and thus, to achieve its goals. *Take over relationship between agents and resources.*

The symbols representing the different types of influences are in Figure 3. Filled symbols correspond to positive influence whereas dashed symbols correspond to negative influence. In what follows, T stands for time.

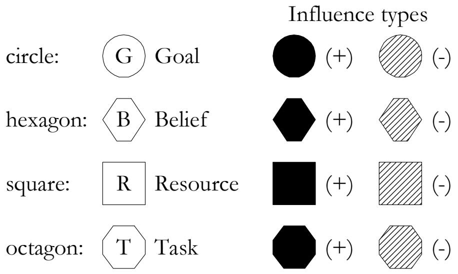
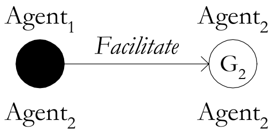
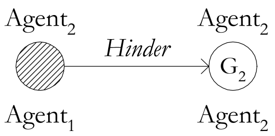


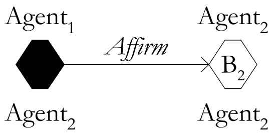
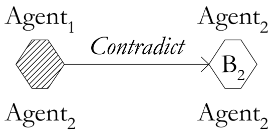
Figure 3 Symbols for influence representation

Goal influence

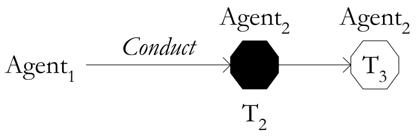
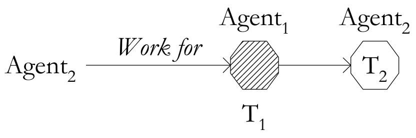
T	Agent$_2$ works towards achieving G$_2$ goal.	
T+1	Agent$_1$ influences agent$_2$	**(+)**  Facilitate(new_goal,G$_2$) Agent$_1$ generates a new goal, filled circle, for the benefit of agent$_2$.

		(-)  Agent$_2$ Agent$_2$ Hinder G$_2$ Agent$_1$ Agent$_2$ Hinder(new_goal,G_2) Agent$_2$ carries out the new goal, dashed circle, for the benefit of agent$_1$.

Belief influence

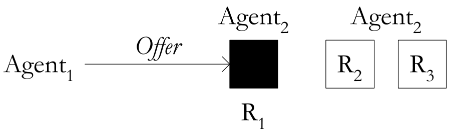
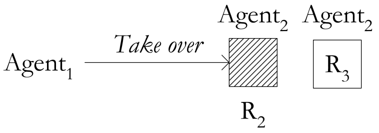
T	Agent$_2$ has B_2 belief.	
T+1	Agent$_1$ influences agent$_2$	**(+)**  Agent$_1$ Agent$_2$ Affirm B$_2$ Agent$_2$ Agent$_2$ Affirm(new_belief,B_2) Agent$_1$ generates a new belief, filled hexagon, for the benefit of agent$_2$.
		(-)  Agent$_1$ Agent$_2$ Contradict B$_2$ Agent$_2$ Agent$_2$ Contradict(new_belief,B_2) Agent$_1$ generates a new belief, dashed hexagon, contradicting agent$_2$'s belief.

Task influence

T	Agent$_2$ carries out T_2 and T_3 tasks.	
T+1	Agent$_1$ influences agent$_2$	**(+)**  Agent$_2$ Agent$_2$ Agent$_1$ Conduct T$_3$ T$_2$ Conduct(Agent$_1$,T_2,Agent$_2$) Agent$_1$ conducts T_2 task, filled octagon, for the benefit of agent$_2$.
		(-)  Agent$_1$ Agent$_2$ Agent$_2$ Work for T$_2$ T$_1$ Work-for(Agent$_2$,T_1,Agent$_1$) Agent$_2$ works for agent$_1$ regarding T_1 task, dashed octagon. T_1 task will precede T_2 and T_3 tasks.

Resource influence

T:	Agent$_2$ manages R_2 and R_3 resources.

T+1	Agent$_1$ influences agent$_2$	**(+)**  Agent$_2$ Agent$_2$ Offer Agent$_1$ R$_2$ R$_3$ R$_1$ Offer(Agent$_1$,R$_1$,Agent$_2$) Agent$_1$ offers R$_1$ resource, filled square, to agent$_2$.
		(-)  Agent$_2$ Agent$_2$ Take over Agent$_1$ R$_3$ R$_2$ Take-over(Agent$_1$,R$_2$,Agent$_2$) Agent$_1$ takes over R$_2$ resource, dashed square, of agent$_2$.

According to the type of influence, either positive or negative, the flow of work between the influencing agent and the influenced agent represents a delegation. For instance, in a positive-goal influence, the influenced agent is supported by a new goal that the influencing agent will be in charge. In a negative-goal influence, the influencing agent assigns a new goal to the influenced agent.

4. Running scenarios

In this section, we discuss how we are applying the influence concept and the GBTR framework as well to military scenarios. These situations could be decomposed into four types: maritime-oriented, land-oriented, air-oriented, and mix-oriented. According to the situation type, we expect that influence should take a different form. In fact, each situation has its structural and functional requirements in terms of doctrines, combat strategies, means, and missions. Therefore, influence should be dealt with differently. Let us recall that the equipments that will be committed to military scenarios should be associated with SAs that will act on their behalf. Simulations that implement proper national doctrines and operational procedures are more likely to be accepted and fostered by computer literate military users and decision-makers.

Each oriented-situation, i.e. maritime, air, and land, requires unique staff skills and training tune to their environment and type of operations, and requires specific infrastructures and equipments. Maritime-oriented situations involve for example vessels and submarines. The nature of the environment, namely sea, has an impact on the operations these vessels will undertake and the interactions these vessels will have together. Air-oriented situations involve for example airports, aircrafts, and helicopters. Land-oriented situations involve for example tanks, armored personnel carriers, and assault vehicles. Finally, mix-oriented situations are a combination of different operations, environments, and equipments[1], e.g. planes and vessels in support of land forces in a littoral area. As with Maritime-oriented situations, each oriented-situation, either air, land, or mix has its requirements that can be very complex and difficult to manage and satisfy.

Figure 4 is an example of the participants that could take part to a maritime-oriented situation. Two vessels and a submarine are used. In military situations, influence between participants is usually bi-directional. For understanding purposes, we assume that influence is unidirectional: vessel$_1$ influences vessel$_2$ and both vessels influence submarine$_1$. Regarding submarine$_1$, receiving contradicting information from vessel$_1$ and vessel$_2$ would occur.

[1] Interesting to consider the equipments that could be simultaneously used in different situations, for example from maritime to land and *vice-versa*. Amphibious vehicles are among these equipments.

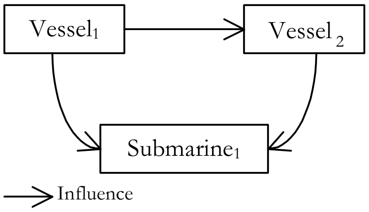


Figure 4 Example of a maritime-oriented situation

In what follows, we provide examples on how influence could occur according to the GBTR framework.

1. Goal influence:
   - Positive influence between vessel$_1$ and vessel$_2$: vessel$_1$ will transport a part of the troops that vessel$_2$ has been tasked to perform. Therefore, vessel$_1$ will pursue a new goal, e.g. carry_troops_for_vessel$_2$.
   - Negative influence between vessel$_1$ and submarine$_1$: because vessel$_1$ could lose a battle in progress, submarine$_1$ has been asked to join the combat as a support to vessel$_1$. Despite that submarine$_1$ is already in charge of securing a specific region, it has to pursue a new goal, e.g. provide_support_to_vessel$_1$.
2. Belief influence:
   - Positive influence between vessel$_1$ and submarine$_1$: submarine$_1$ believes that vessel$_2$ is friendly. Vessel$_1$ confirms to submarine$_1$ that vessel$_2$ is friendly. This permits to reinforce submarine$_1$'s beliefs.
   - Negative influence between vessel$_1$ and vessel$_2$: vessel$_2$ believes that submarine$_1$ is committed to a surveillance operation. However, vessel$_1$ informs vessel$_2$ that submarine$_1$ has been withdrawn from this operation. This new statement contradicts what vessel$_2$ assumed about submarine$_1$'s responsibilities.
3. Task influence: it is a consequence of goal influence.
   - Positive influence between vessel$_1$ and vessel$_2$: according to the positive goal-influence case (see above), vessel$_1$ has been ordered to transport equipments on behalf of vessel$_2$. Therefore, vessel$_1$ needs to perform as tasks: load equipments from the original destination, convey these equipments, and finally unload these equipments at the final destination.
   - Negative influence between vessel$_1$ and submarine$_1$: according to the negative goal-influence case (see above), submarine$_1$ will fulfill new tasks for vessel$_1$, such as attacking the enemy float. In fact, these tasks have not been planned in submarine$_1$'s initial schedule.
4. Resource influence: it is a consequence of goal influence
   - Positive influence between vessel$_1$ and vessel$_2$: according to the positive goal-influence case (see above), vessel$_1$ has to transport equipments on behalf of vessel$_2$. The new tasks that vessel$_1$ will carry out requires the use of its resources, such as a crane.
   - Negative influence between vessel$_1$ and submarine$_1$: according to the negative goal-influence case (see above), submarine$_1$ will fulfill new tasks for vessel$_1$. To this end, submarine$_1$ will use its resources.

5. Conclusion

In this paper, we discussed influence role in modeling and understanding software agents' behavior. To this end, we suggested the GBTR framework that views influence from two inter-related perspectives: knowledge and organization. The knowledge perspective consists of goal and belief components while the organization perspective consists of task and resource components. Influence could be either positive or negative. This requires enhancing the agent that will be influenced with appropriate mechanisms, such as assessment. Finally, we illustrated the use of the GBTR framework on different situations from the military domain. More work is needed. For instance, how to define the origin of influence is among our concerns. We just started considering Bayesian Networks to deal with this concern.

References

[1] http://www.influenceatwork.com/. Visited December 2000.

[2] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1) pp. 7-38, 1998.

[3] D. Foryth. *Group Dynamics*. Brooks/Cole, Pacific Grove, CA 2nd edition, 1990.

[4] R. Conte Intelligent social learning. in *the symposium on starting from society – the application of social analogies to computational systems* – held as part of AISB'2000, April 2000.

[5] L. Falzon, L. Zhang, and M. Davies. A Policy Analysis Approach to Operational Level Course of Action Analysis. in *The 5th International Command and Control Research and Technology Symposium, Modeling & Simulation Track*. http://www.dodccrp.org/ visited January 2001.

Force Templates –
A Blueprint for Coalition Interaction within an Infosphere

Robert E. Marmelstein, Ph.D.

Air Force Research Laboratory (AFRL) - Information Directorate
Information Systems Division (AFRL/IFS)
525 Brooks Road
Rome, NY 13441, USA
Robert.Marmelstein@rl.af.mil

Abstract. In this paper, we present the emerging force template model for the Joint Battlespace Infosphere (JBI) and discuss how it supports successful coalition operations. Infosphere architectures, such as the JBI, represent the way ahead for leveraging web and e-commerce technologies to streamline command, control, and intelligence (C2I) operations. We introduce the force template concept as the principal mechanism to quickly integrate battlespace entities (and their clients) into the JBI. Additionally, we show how force templates can ensure proper information dissemination within the JBI. With its emphasis on resource exchange and control, force templates provide the flexibility needed to seamlessly share information among members of ad-hoc coalitions.

1. **Introduction.**

There are many areas where technology has not caught up to military strategy and doctrine--coalition warfare is one of these. Future military operations will require close coordination and information sharing among heterogeneous units, coalition forces, and other civil and non-governmental (NGO) organizations. While United States increasingly relies on coalitions to achieve its military objectives, the technological infrastructure necessary to support this strategy has been lacking. The gulf between the desired and the possible is especially glaring in the area of C2I. For example, in the Joint Force Expeditionary Experiment (JEFX) '99, the effort to integrate coalition members into the Combined Air Operations Center (CAOC) was deemed a failure. This result was due to three factors: US-only applications within Theater Battle Management Core Systems (TBMCS), use of SIPRNET as the CAOC backbone, and the population of CAOC databases with US-only information [3]. The changes required to remedy this situation were sufficiently difficult as to result in the cancellation of the planned coalition operations in JEFX '00 [4].

One of the key recommendations from JEFX '99 was to develop a CAOC backbone accessible by all coalition users [3]. While some approaches include explicitly tagging database elements for releasibility, a cleaner solution requires a new paradigm that manages information in terms of standardized, discrete objects. Such an approach would enable the following positive developments:

- The segregation of information objects from their source applications and databases.
- Making publish, subscribe, query, and transformation capabilities available to producers and consumers of these information objects.
- The specification of policy governing how the published object types can be disseminated within the infosphere.

Currently, information potentially releasable to coalition partners is often combined with other, sensitive data within client applications and databases. The unfortunate result is a denial of useful information to coalition partners since the aggregated data is at a system high level. Segregating information into packages that are small, coherent, and discrete makes it easier to control and, therefore, distribute to other coalition members.

It is also possible to convert some sensitive data into a releasable form. In many cases, lightweight programs (referred to as fuselets) could be employed to accomplish these transformations. Policy associated with information objects (nominally defined by the publishers) will determine to whom, and in what form, specific objects would be disseminated. The combination of an infosphere, better information packaging, and fuselets would facilitate the controlled, secure sharing of information within a coalition.

2. The Joint Battlespace Infosphere.

The JBI is a system of systems that integrates, aggregates, and distributes information to users at all echelons, from the command center to the battlefield. Infospheres are a critical stepping stone to solving the problems of coalition C2I integration because they inherently provides many of the capabilities described in the previous section. The conceptual framework for JBI was outlined in two consecutive Air Force Scientific Advisory Board (SAB) reports, Information Management to Support the Warrior (1998) [5] and Building the Joint Battlespace Infosphere (1999) [6]. The SAB vision for the JBI encompasses the four key concepts described below and in Figure 1.

Information exchange through publish, subscribe, and query. This capability enables the user to locate and subscribe to information resources available within the JBI. Each publisher is responsible for tracking users that have subscribed to its resources. When an information resource is published, a tailored version of that resource is forwarded to the subscriber.

Transforming data to knowledge via fuselets. Fuselets are lightweight programs or scripts that process incoming information objects received from established subscriptions. When these objects arrive, fuselets can then aggregate, correlate, and/or transform them into information of interest to a given subscriber.

Distributed collaboration through shared, updateable knowledge objects. This concept refers to the ability of the JBI to facilitate collaborative problem solving among multiple, diverse users.

Assigned unit incorporation via force templates. A force template is an electronic description of an entity that enables its integration into the JBI (including all its subcomponents).

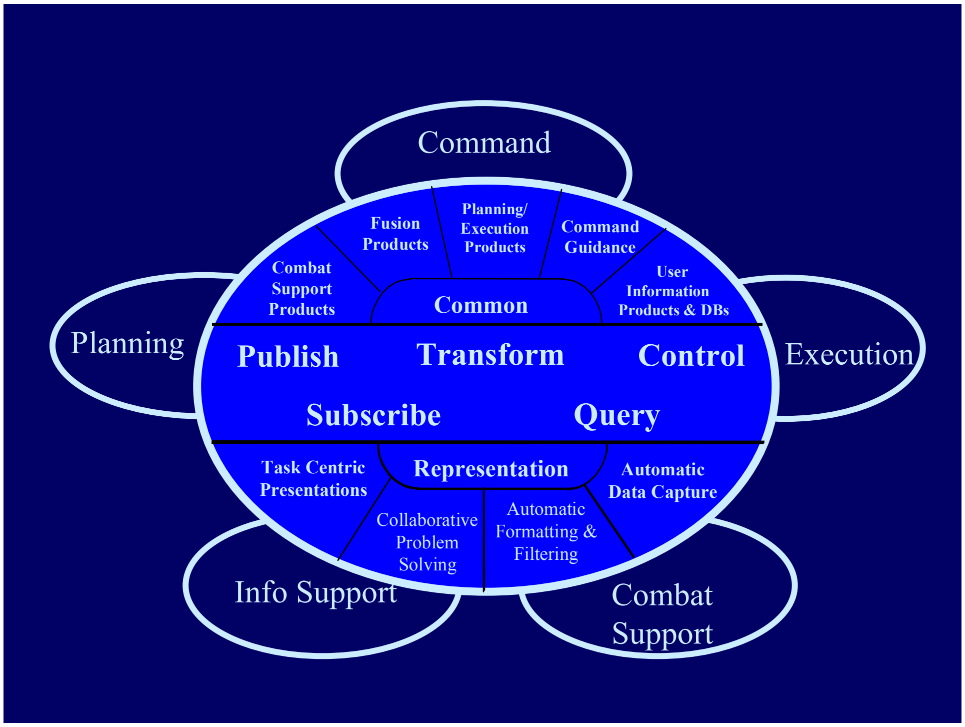


Figure 1 – JBI Capabilities

3. Force Template Concepts

In this section, we build on the definition given in the last section by discussing why force templates are needed, how they model coalition units, and what information they provide to the JBI.

Why are force templates needed? While the JBI provides platform for information transfer, others must provide the content. For an infosphere to have value, the participating entities must "plug in" and use it to exchange information and service resources. The force template contains the information that enables operational entities within the battlespace (and their clients) to quickly interact using the JBI platform.

The force template also includes the context and policy that define an entity's contract with the JBI. One of the key motivations for developing the force template concept is the need to allow the JBI to grow (shrink) in a modular fashion that reflects the phase of the associated military operation. In short, the JBI must handle dramatic and sudden content changes while maintaining an acceptable level of service. Without the force template mechanism, it becomes extremely difficult to track and manage the changes to JBI content resulting from the arrival and departure of coalition units.

Entities, Clients, and Passes. An entity is an organization that decomposes into multiple components. Those components may either be other entities (child entities) or clients. In this model, entities primarily correspond to operational military units and the organizations that support them. Both parent and child entities may have their own force templates. For example, a wing and its associated squadrons may each have their own force templates. These templates may be separate, but linked based on their relationship. The level at which force templates are required should reflect the modularity of the force (e.g., the level at which forces can be mixed, matched, or tasked).

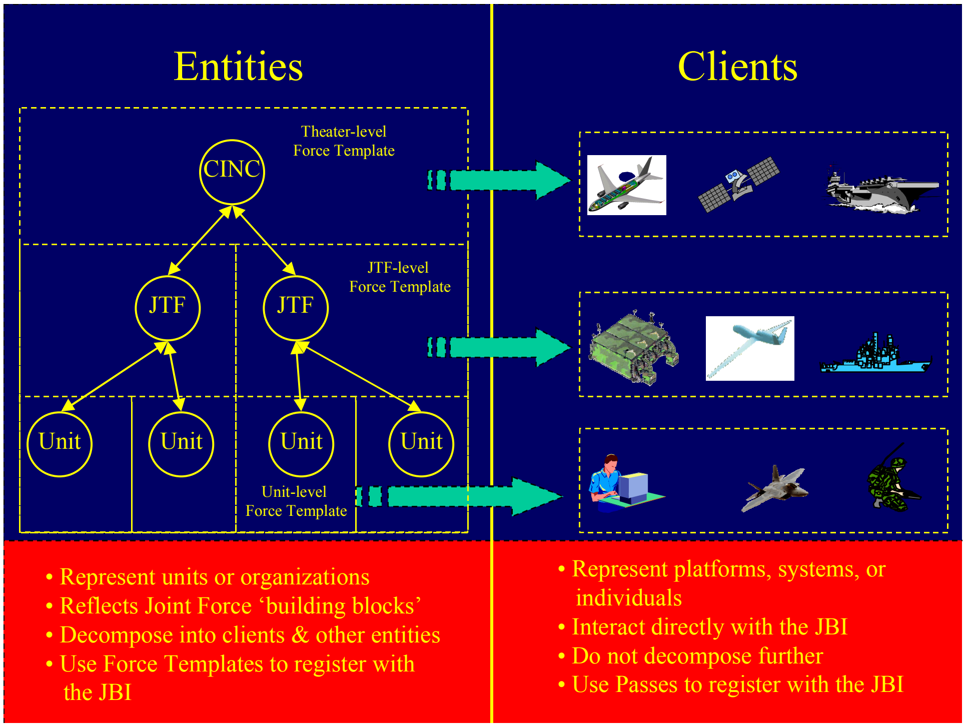
Clients are owned by entities. It is intended that clients correspond to specific individuals, systems, applications, repositories, or platforms. For example, an F-15 client may be owned by a fighter squadron entity. A client will interface directly with the JBI on behalf of its owner. Unlike entities, clients may not decompose into subcomponents. The entity that owns a client must be registered before the client can connect to the JBI platform. Entities at any level may own a distinct set of clients. The entity client relationship is illustrated in Figure 2.

A pass is an electronic description of a client that enables it to interface with the JBI. The pass defines what a client may do when connected to the JBI. This is primarily expressed in terms of authorized client publications and subscriptions. The information in the pass must be consistent with the force template of the entity that owns the client. The differences between force templates and passes are summarized in Table 1.

Table 1 – Comparison of Force Template and Passes

	Force Template	**Pass**
Purpose	Register entities with JBI	Register clients with JBI
Activation Prerequisite	Approval of Joint Force Commander (JFC) or parent entity	Registration of owner entity's force template with the JBI
JBI Interface	Force template controller	Client adapter
Content Characteristics	Distributed, hierarchical, decomposable	Consolidated, cannot be decomposed
Minimum Contents	– Entity info requirements – Entity info products – Entity level constraints – Passes for clients owned by the entity	– Info object advertisements – Subscription requests – Client level constraints

Force template contents. There is a wide spectrum of information that the force template could potentially provide the JBI. Some items are essential for the operation of the JBI; others are extensions of the capabilities outlined in the SAB report. As a result, three separate categories are used to characterize force template content; these are: necessary, desired, and speculative (also see Figure 3).



- Represent units or organizations
- Reflects Joint Force 'building blocks'
- Decompose into clients & other entities
- Use Force Templates to register with the JBI

- Represent platforms, systems, or individuals
- Interact directly with the JBI
- Do not decompose further
- Use Passes to register with the JBI

Figure 2 – Entity/Client Relationship

Necessary Contents:

Information needed by the entity. This refers to information that the entity says it needs to function within the theater. Information can be requested in terms of categorical requirements (expressed as a metadata query) or in terms of specific information object types (predefined subscription requests).

Information provided by the entity. This refers to information that the entity says it can provide within the theater. These will likewise be expressed using metadata descriptions or in terms of specific information object types (advertisements).

The constraints associated with the above. In many cases, information provided or requested will have constraints associated with it. Examples of subscriber constraints include desired quality of service, pedigree, preferred sources, and required delivery windows. Examples of publisher constraints include: anticipated publication times and rates, and dissemination constrains. These constraints may also be expressed in terms of rules about information object *content*. In this case, publisher advertisements may also include information on publisher capabilities (such as filtering and query capabilities). The JBI platform will use these constraints to broker information requirements against available information products

Security Information. This is a broad and evolving category. The force template could provide a number of security related items to the JBI. This may include:

- The identity and security credentials for individuals occupying key unit positions.
- Public keys for specific clients (individuals, platforms, or systems).

- Dissemination limitations on published information.

Desired Contents:

Information Pedigree. This refers to indicators of the quality, reliability, and integrity of entity publications. As such, pedigree ratings may be provided in part by the entity (self-assessment) and in part by the JBI (based on previous history or consumer experience).

Mapping of Specific Personnel to Operational Roles. Force templates for similar units will have a high degree of commonality that extends to positions within the unit. The force templates will communicate to the JBI which personnel are authorized to function in those positions. This mapping could enable the JBI Info Management Staff (IMS) to issue the proper security certificates for those individuals.

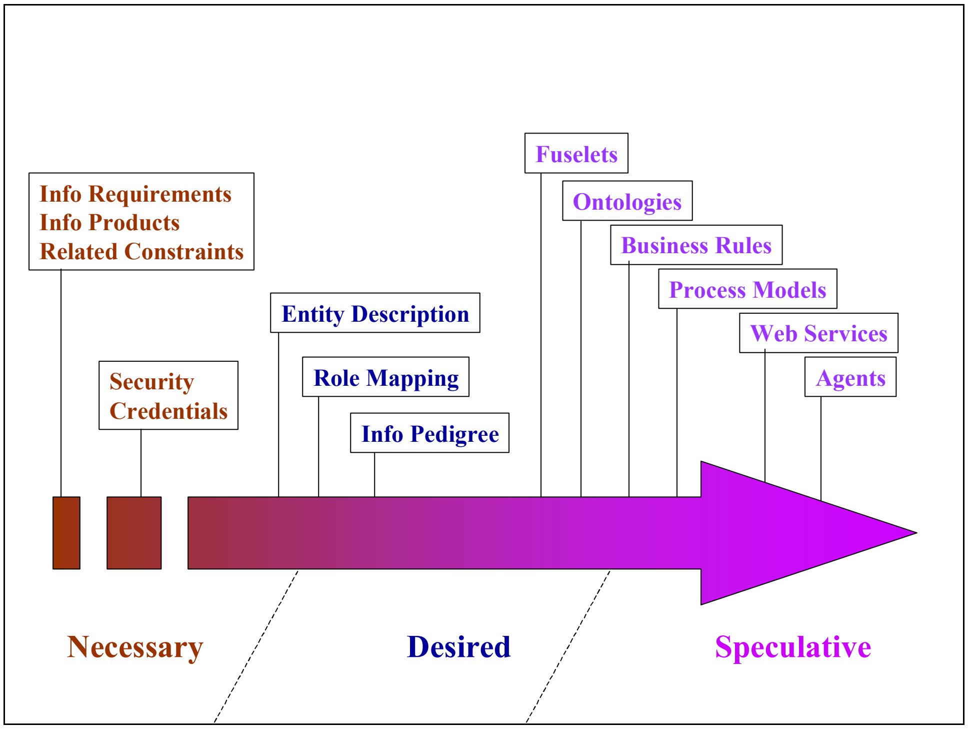


Figure 3 – Force Template Content

Entity Description. This will describe the characteristics of the entity interfacing with the JBI. Ideally, this will take the form of a "resource map" (similar to an active directory) that describes all entity components (e.g., devices, clients, data sources, and people) visible to the JBI. It also includes the child entities that compose the entity (e.g., squadrons within a wing). Each item on the map will list the characteristics of the particular resources. Examples of some unit characteristics include: mission description, unit organizational structure, location, capability description, resource maps, and pointers to associated force templates.

Speculative Contents:

Ontologies and Ontology Mappings. The more diverse the coalition, the greater the importance of shared semantics. For coalition operations to be successful, it is essential that a consistent set of terms be used to facilitate information sharing [1]. As a result, it is desirable to include ontologies specific to an entity, system, or related domain. Whenever possible, these ontologies should come with mappings to common ontologies utilized within the JBI.

Fuselets. Fuselets may be associated with either publications or subscriptions. Examples include XSLT, Excel spreadsheets, Active-X components, or Java beans. Ideally, the force template would contain references to fuselets available from the entity. These fuselets *should be associated with* specific publications within the JBI (but not necessarily by the providing entity).

Process Models, Rules, and Constraints. These items describe how the entity does business in the theater of operations. Ideally, these will be specified in terms of the included ontologies.

Available Services, or Agents. These items describe services provided by the entity for use by other (appropriate) JBI entities. Examples of services might include: computation of look angles for satellites, requests for surveillance of certain areas, and agent services for determining unit personnel location and status.

4. Entity/Client Interaction Model

The SAB report painted a general picture of what the JBI should do and what technologies it might leverage. It did not, however, provide guidance on how the JBI should behave. Since there is no official model for interaction with the JBI, we will take a first cut developing one here. The model proposed here (summarized in Figure 4) ensures the following requirements are met:

- The JBI platform has visibility and control over its inputs and outputs.
- Entities maintain control over what their clients are allowed to do within the JBI through the force template infrastructure.
- Dynamic changes to the force template can be made after registration, allowing the flow of information to evolve during the mission. These changes may be initiated by the top down (from the parent entity or the JBI information staff) or from the bottom up (by the client).
- The integrity and consistency of associated force templates and passes are maintained.

The first part of the model deals with the registration of the entity with the JBI. The notional steps in the process are listed below.

1. Locate the appropriate JBI.
2. Entity requests permission to connect to JBI platform.
3. JBI requests force template package from entity.
4. The entity transmits its force template to the JBI platform.
5. JBI processes force template package.
6. JBI tenders response: acceptance, partial acceptance, or rejection.
7. If acceptance is granted, a controller process is elaborated for the force template.

As discussed earlier, the entity must register prior to registration of its clients. Clients will not be allowed to register with the JBI until an acceptance or partial acceptance is tendered. It is assumed that child entities are not required to register before their parents. This feature offers flexibility in extending the JBI in cases such as when individual squadrons deploy to a theater without their parent wing.

The acceptance of the entity's force template triggers the allocation of a Force Template Controller (FTC) within the JBI platform. The FTC is a gatekeeper that ensures clients behave in a manner consistent with the force template. It also controls changes to the force template that may occur during the entity's JBI session. These changes may be initiated from the bottom up (e.g., client wishes to publish a new information object type) or from the top down (e.g., parent of entity or JBI information staff mandates changes to the force template).

The proposed client interaction model is illustrated above. The steps for registration of individual clients are listed below.

1. The FTC ensures that adapter processes are elaborated for each client associated with the entity's force template.
2. The passes associated with the clients are cleared for activation within the JBI. The individual clients may attempt connection to the JBI.
3. The client registers with the JBI through its associated adapter.
4. The adapter validates the client. It then receives permission to interact with the JBI in accordance with its pass.
5. If the pass is not validated, permission to interact is denied.

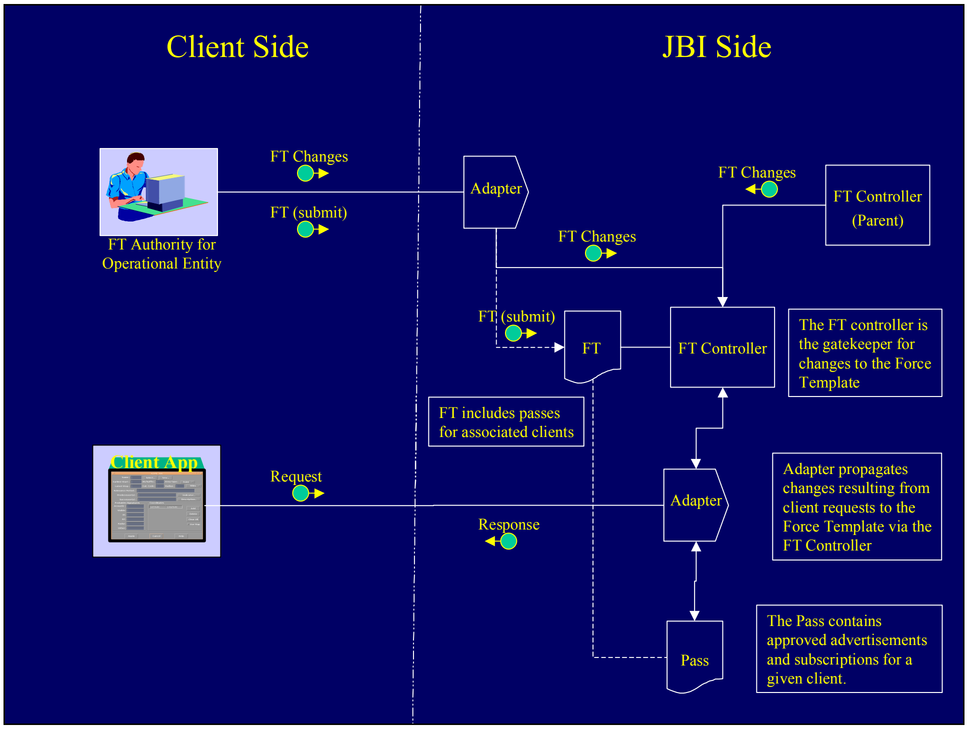

Client Side
JBI Side
FT Changes
FT (submit)
FT Authority for Operational Entity
Adapter
FT Changes
FT Controller (Parent)
FT Changes
FT (submit)
FT
FT Controller
The FT controller is the gatekeeper for changes to the Force Template
FT includes passes for associated clients
Client App
Request
Response
Adapter
Adapter propagates changes resulting from client requests to the Force Template via the FT Controller
Pass
The Pass contains approved advertisements and subscriptions for a given client.

Figure 4: Strawman Force Template Interaction Model

As discussed earlier, the force template contains all passes associated with the entity's clients. The pass contains the approved advertisements and subscriptions for a given client (refer to Table 1). After the entity registers, its passes are maintained by the JBI platform. When the client registers, it submits an encoded reference to the pass that is compared to the version on the JBI side. If they match, the client is given permission to interact with the JBI; otherwise, permission is denied.

Once successfully registered, the client can then initiate JBI operations (e.g., advertise, publish, subscribe, and query) for approved information objects. If the client needs to change its profile, this request is forwarded to the corresponding FTC (through the client's adapter). If the request is consistent with the force template permissions, then an affirmative response is sent back to the client. As a result, the client's adapter on the JBI platform updates the pass. If a negative response is given, however, the request is elevated to the appropriate authorizing authority for further consideration.

Correspondingly, if changes are directed from above (the legitimate authority within the entity, a parent of the entity, or from the JBI information staff), then those changes are also routed through the FTC. Since these changes are directed (not requested), the force template is automatically updated. This causes the changes to propagate back down to the passes of the affected clients. These changes may result from higher level approval of a client's request that was initially denied by the FTC.

Note that the copy of the force template, and associated passes, updated during the mission is the one maintained by the JBI platform. The entity still retains its copy of the original force template submitted. Because the entity can access (copy) the current force template at any time, it can choose to save versions of the force template as it evolves. If desired, these saved versions can then be used in the future (instead of starting over with the original).

5. Impact on Coalition C2I Operations

In this section we discuss how the force template model enhances coalition C2I. For the sake of this exercise, it is assumed that all in-theater coalition possess the credentials and systems necessary to interface with the JBI. Recall that when each coalition member registers with the JBI, their force template will (at a minimum) define what information they need, what they have, and the constraints associated with each.

Although the JBI will be primarily oriented toward military forces, the force template mechanism will provide the flexibility to accommodate relatively ad-hoc coalitions. To be successful, military operations other than war (MOOTW) will require the participation of a wide variety of organizations, including local civil authorities and NGOs [2]. As a result, future C2I systems must be designed with these organizations in mind and provide flexible, appropriate mechanisms for interfacing with them. In cases where these organizations are operating in-theater, they can help provide essential services, such as humanitarian relief, and may (indirectly) serve as important sources of intelligence. In turn, these organizations must be protected without compromising military operations. Successfully integrating these organizations into a common C2I environment will be complicated by the fact they have fundamentally different missions, practices, ontologies, and equipment from the involved military units. While not a total solution, the force template acts as a general-purpose repository for information that describes these aspects of each entity; future C2I applications can draw on these building blocks to overcome these problems.

Regardless of the coalition member's identity, their validated force template will serve as the basis for deciding how their information is utilized, and by whom. Once an entity registers with the JBI, the information products they promise to provide can be brokered according to their specified constraints. This enables each coalition member's information requirements to be intelligently matched with the resources designated as accessible to that member. As part of this process, the JBI will identify the available fuselets that can be used to transform sensitive published information into a form that is releasable to the coalition member. The JBI user will also be able to browse resource directories and identify useful categories of information objects not currently available to him (if those entries are not masked). Once identified, the member can use his force template as the basis for negotiating access to these resources from the publisher.

Although there is no guarantee that all of a coalition member's information requirements will be satisfied by this process, it enables him to leverage the full range of resources (both information and services) available to meet his needs. Given this, the coalition member may be able to satisfy his needs from an ad-hoc collection of available sources, rather than relying on a single source. Thus, in contrast instead of the wholesale denial of information that commonly occurs today, the JBI infrastructure will make it possible for the member to get some subset of what he needs. Within this context, the force template serves as an important enabling mechanism to fashion flexible, information solutions for a diverse set of coalition users.

6. Conclusion

If the last decade is any guide, future military operations will be carried out by dynamic, diverse coalitions composed of military, civil, and NGO members. The key to success in these operations will be insuring that these entities can quickly exchange both information and service resources within an information-centric C2I infrastructure (infosphere). We have introduced the force template as an enabling mechanism to facilitate this interaction. In this paper, we have taken a first cut at the force template concept by defining what it might contain. We also introduced a model for how it can be used to integrate, and control the interaction of, operational entities (including their children and clients) with the JBI infrastructure. Ultimately, the force template serves as a repository for mission critical information about a battlespace entity; this information includes its identity, what it wants, what it has to offer, and how it intends to operate within the theater. With these items, the infosphere will be able perform contextual brokering of the available resources of each infosphere member. The net result is that infospheres, such as the JBI, can become flexible platforms for the exchange of information and services among coalition partners, insuring (to the extent possible) that the right resource gets to the right member at the right time.

References

1. Anderson, R.H., and Baer, W. (2000) "Potential Military Applications of Peer-to-Peer Computer Networks", DRR-2670-RC, RAND Corporation, November 2001.
2. Estilow, R.A. (1993) "US Military Force and Operations Other Than War: Necessary Questions to Avoid Strategic Failure", The Maxwell Papers, No 3, August 1996.
3. United States Air Force Execution Office, "Final Report - Joint Expeditionary Force Experiment 1999", 3 December 1999, pages 44-45.
4. United States Air Force Execution Office, "Alliance Participation Lessons Learned Supplement to: Final Report - Joint Expeditionary Force Experiment 2000".
5. United States Air Force Scientific Advisory Board, "Report on Information Management to Support the Warrior", SAB-TR-98-02, December 2000.
6. United States Air Force Scientific Advisory Board, "Report on Building the Joint Battlespace Infosphere", SAB-TR-99-02, December 1999.

Effects-based Coalition Operations: Belief, Framing and Mechanism

Maris McCrabb

DMM Ventures, Inc.
107 Elise Place, Yorktown, VA USA 23693
buster@dmmventures.com

Abstract. Coalition operations over the past decade exhibit a propensity towards collegial decision-making even in the presence of formal, normally hierarchical, decision-making apparatus. Meanwhile, the US military, especially the US Air Force, is adopting effects-based operations (EBO) as a method of planning, executing, and assessing military operations that achieves desired effects that attain strategic objectives. EBO forces decision makers to look at outcomes and their explanations more so than on actions taken. Hence, an EBO approach significantly affects decision-making. Both these requirements, collegial decision making and EBO, affect supporting knowledge systems. This paper explores all these implications. Following a short explanation of the problem, the second section describes EBO. The third section contrasts collegial decision-making models with traditional hierarchical decision-making models. This draws largely from work done for the US Air Force Research Laboratory Human Effectiveness Directorate (AFRL/HE). Section 4 presents research on a situation-aware, recognition-primed, variable risk-propensity model of collegial decision-making based in an EBO context. Section 5 discusses the implications of that model and EBO on knowledge base design requirements. Section 6 concludes the paper and offers some areas for future research.

1 Section 1 Introduction

The essence of military command is allocating—deciding—scarce resources to attain desired goals. Linking the use of resources—actions—and the outcomes those actions obtain is the domain of strategy. There are many methods of developing strategy. Each ultimately revolves around how the decision-maker believes the actions taken will achieve the desired outcome. This causal explanation is called mechanism. Older strategies focused on the actions taken. Current approaches tend to focus solely on outcomes with little concern on understanding how those outcomes arose from the actions. Effects-based operations (EBO) focuses on causal explanations: why will (i.e., planning) or why did (i.e., post-execution) the actions planned (taken) result in the desired effects? This description suggests two critical elements. One is assessment: how can a decision-maker understand the causal mechanisms. The other is decision-making. What impact does an EBO method have on decision-making? Compounding this question is coalition operations where decisions are more framed through collegial processes rather than hierarchical processes. It is the issues of EBO, decision-making, and coalition operations that concern this research note.

2 Section 2 Effects-based Operations (EBO)

2.1 Description.

Effects-based operations (EBO) is an approach—a way of thinking—to planning, executing, and assessing military operations that focuses on the results of military operations—and the explanation of how those results came about—rather than the actions—sorties flown, rounds fired, or tons of relief materials delivered—of military units. (Davis, 2001) It is thinking strategically. (Dixit and Nalebuff, 1991) As such, it spans the gamut of military operations from humanitarian relief to major theatre war. It accounts for lethal and non-lethal applications of force delivered kinetically or via non-kinetic modes. EBO incorporates and expands upon traditional approaches such as targets-based and strategy-to-task. The most significant challenge for EBO is predicting and assessing how physical actions result in behavioural outcomes. Physical should not be confused with merely flying aircraft or dropping bombs.

Pushing keys on a computer keyboard instigating a computer network defence is a physical action. Issuing messages an enemy can "intercept" from a fictitious headquarters, as part of a deception operation is also a physical action. The goal of an effects-based approach is tracing and understanding how those actions affect the attacker or enemy commander's behaviour. Functions are defined as broad, fundamental, and continuing activities. Processes, or activities, are how work—tasks--is done. For commanders, the most basic activities are planning, executing, and assessing operations. EBO is a method for accomplishing those tasks. This section describes those activities from an effects-based perspective. (McCrabb, 2002)

2.2 Effects-based Planning.

EBO, as with any approach to planning, executing, and assessing military operations, starts with Commander's Intent. See Figure 1. The provision of end state, purpose, method, and risk begins the process of mission analysis where objectives, desired effects, specified, and implied tasks, constraints and restraints and other needed elements of information start. For example, the method specified in Commander's Intent may direct an analysis of nonlethal applications such as deception or psychological operations. Likewise, listed restraints on certain types of collateral damage—for instance, damage to electrical power systems—may preclude certain strategy options. The end state lists the set of conditions required to achieve the JFC's objectives. Purpose provides the rationale for the mission. In simpler terms, the end state gives *what* is to be accomplished. Method gives *how* the end state is to be accomplished. In addition, purpose gives *why* the end state is to be accomplished.

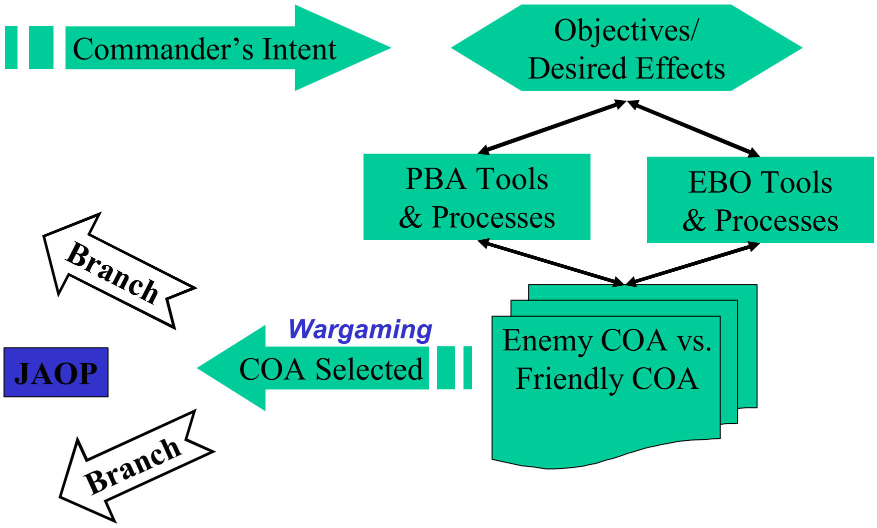
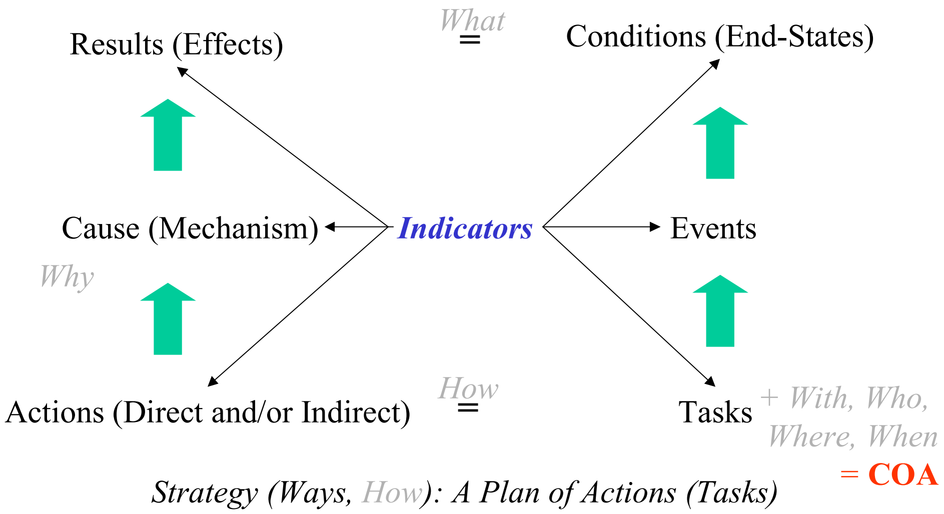


Fig. 1. From Commander's Intent to JAOP: COA Development

<u>Strategy (COA) development</u>. Together, these form the heart of a course-of-action (COA). At the JFC and JFACC level, the COA embodies the commander's strategy—the art and science of employing resources to accomplish objectives. The COA is the plan of activities the commander envisions that accomplish the objectives and desired effects. Commander's Intent, strategy, and COA can be used almost interchangeably though COA generally contains the most detail. Besides the *what*, *how*, and *why*, a COA includes *with* (resources), *who*, *where*, and *when*. It also includes mechanisms, sometimes referred to as the second *why* since mechanism explains why an action should result in some specified effect.

Between the method and COA, a complete description of the chosen strategy should be available. See Figure 2. EBO is a method, not a strategy. Attrition is an effect. Paralysis is an effect.

Targeting: COG/TS analysis. Targeting is the analysis of the situation in relation to the commander's intent and available resources in order to discover vulnerabilities that, if exploited, attain the commander's intent. Normally, this analysis starts with centre-of-gravity (COG) analysis and proceeds through target systems analysis (TSA) and concludes with identification of desired mean of impact, if that is appropriate. A COG is those characteristics found in the situation, for example, in an enemy, from which the adversary or friendly elements derive their will or capabilities. It is the point or points against which all our energies should be directed in order to exploit an enemy's COG or to defend our own. A COG may or may not be directly accessible and may change within the course of a campaign or operations. COG exists at all levels of warfare. The COG/TS analysis provides the objects for the commander's desired effects. For example, a communication link within an Integrated Air Defence System may be the object for disruption in order to gain the desired effect of freedom of air action over an enemy. An Information Warfare attack against one or several of those links might be the actions that would trigger a mechanism, e.g., inability to pass data, which results in attaining the desired effect of "disrupted communication."



Strategy (Ways, How): A Plan of Actions (Tasks)
that employ resources (means, with)
To accomplish Ends (What)

Fig. 2. COA Development: EBO and Operational Art

DAEO generation. The planning process takes commander's intent and turns it into orders executing units carry out. This mission data today is found in an Air Tasking Order. In the near future, it might be available in a Dynamic Air/Space Execution Order (DAEO). In order to counter emerging threats or exploit emerging opportunities, commanders require the means of reacting very quickly. This argues against a batch process and towards a more continuous process. Air and space power is often falsely charged with being unresponsive due to the length of the ATO cycle. Experience shows this not to be the case. From World War II and Vietnam War cases where close air support (CAS) missions were employed within minutes of requests, to the Gulf War's system of "push CAS" where aircraft were constantly on station, often returning with their munitions unexpended, air and space power showed great responsiveness and flexibility. Still, planning processes tend towards batching sorties into one ATO. The DAEO process envisions a largely continuous process where target queues are dynamically executed as they are built. Dynamic does not mean instantaneous. If a desired effect is known minutes, hours, or days in advance, the DAEO can be generated—and refined—as the requirement is known.

2.3 Effects-based Execution

DAEO execution. A key ingredient in the success of the DAEO process is the collaboration between the operational-level tasking organization—normally the Air Operations Centre (AOC)—and the tactical-level execution organization, normally a squadron through a Wing Operations Centre (WOC). This collaboration, which starts during the planning process, continues throughout execution. It is embodied in the concept of centralized command and control and decentralized execution. The AOC maintains track of the status of the plan's accomplishment of commander's intent from a theatre-wide and top-down perspective. The WOC maintains track of the status of specific tasks assigned by the AOC from a bottom-up perspective. Since each task likely contributes directly to the attainment of some direct effect and indirectly to the attainment of some cumulative effect, the close collaboration between WOC and AOC is essential.

2.4 Effects-based Assessment

COA assessment. Assessment activities begin well before tasks are executed. During the planning process, assessment requirements are an integral part of the JAOP, COD, and DAEO generation. The second set of assessment activities during planning are those directly related to assessing the likelihood the COA options developed will attain commander's intent. This assessment process occurs largely through wargaming. The enemy COA are war-gamed against Blue COA options using criteria established by the commander. Normally this includes adequacy, completeness, and feasibility plus other criteria such as probability of friendly losses, time to attain the objectives and desired effects, and collateral damage. The outcome of war games is used by staff to form their recommendation to the commander on which COA option to adopt. Note that often the COA options not adopted become branch plans so the wargame information is retained. Often commanders will modify staff recommendations. Under the DAEO construct, this feedback into the planning process is expected. As the COA is modified from its original form, the assessment process recalculates the probability of attaining commander's intent as well as the changes in the criteria values. Again, the goal is providing useful information to the commander for their decision-making.

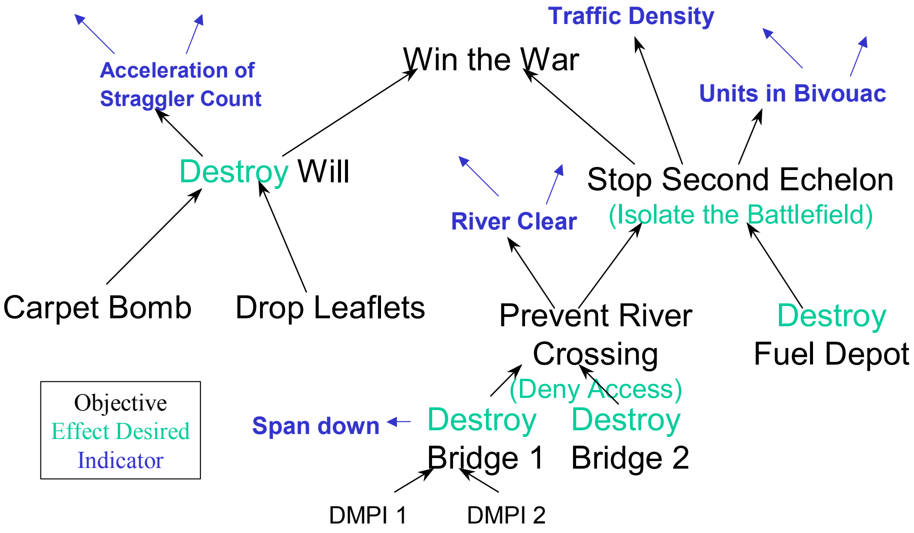


Fig. 3. Indicators Are Crucial For Assessment

Campaign assessment. The traditional combat assessment process can be viewed as a bottoms-up or vertical process. Effects-based campaign assessment can be viewed as a horizontal process. It is the merging of the two that provides a commander the richer view of operations than previously available. Effects-based assessment starts with indicators. See Figure 3. These are the evidence of effect, mechanism, or action. Combat assessment traditionally focused on effects and actions at the direct, physical effect level. Campaign assessment builds upon and broadens this to include the indirect, complex, and cumulative behavioural effects. For example, if the operational objective and desired effect is to isolate the second echelon, that cumulative effect is likely to include a mixture of direct, physical effects as well as indirect, behavioural effects. The functional and systemic damage assessments from BDA can provide information on the former—for instance, the status of lines of communications--while the indicators planned for the indirect, behavioural effects and mechanism—such as COMINT reports on enemy movement plans—adds more depth to the analysis on whether the isolation is being achieved.

2.5 Section Summary

This summary of EBO theory described the many and varied decision points commanders face during planning, execution, and assessment of a military operation. The following section delves into decision-making theory.

3 Section 3 Decision Making Models

3.1 Classic Hierarchical Decision Making Models.

This section paves the way towards developing a collegial decision making model that supports effects-based coalition operations. It briefly describes three generic models: the classic rational actor model, an early modification of that model, and the observe-orient-decide-act (OODA) model that has gained wide popularity as a military decision making model. Next, some early attempts to examine collegial decision making situations are described. These efforts focused on the pathologies of decision making that arise from group situations. "Decide by committee" to this day evokes negative images in most people. Despite this, most decision-making does take place among several individuals. Therefore, the last subsection describes the elements in a collegial decision making model that must be accounted for in developing a prescriptive model of collegial decision making.

3.1.1 Rational Actor model

The theory of rational choice in decision-making is one of the oldest portrayals of human behaviour (March, 1994). It is one of the most aspired to approaches to decision-making and one of the most mis-understood. Rational decision-making does not mean "good" decisions always arise. Rather, it is a procedural approach that is consequential, in that actions taken are believed to cause future outcomes, and preferential, in that the decisions made reflect the preferences of the decision maker. The key word here is belief. Later in this paper, the critical role of framing is discussed. That is how the argument is structured. It is the claim of this paper that in a collegial decision making environment, arguments (say, for example, about how a set of actions will lead to some given effects) that are structured around the most common belief schema of the parties involved are most likely to result in the shared understanding of the group. While perhaps intuitive, it must be pointed out that a group that employed a rational choice model is unlikely to reach much of a degree of shared understanding because of the preferential element located within that model. Hence, only to the degree the individuals in a group have shared belief, as common preferences, will a rational choice decision-making model work in a collegial environment. Shared belief is uncommon in coalition operations amongst all members, especially coalitions formed for a single purpose or contingency, or a long-standing coalition when new members arrive.

March points out that, "Some versions of rational choice theory assumes that all decision makers share a common set of (basic) preferences." (March, 1994: 3) Other elements (and limitations) of the model are that all alternatives are examined, that all decision makers have perfect information about the alternatives (especially consequences), and that there is some objective function used to define the selection criteria. From there, decision-making becomes almost mechanical. Such models are highly adaptable to quantification and mathematical processes.

3.1.2 Satisficing

March again: "Pure rationality strains credulity as a description of how decisions actually happen." (5) Numerous modifications to the basic model tend to soften one (rarely more than) assumption or another. Most modifications start by relaxing the assumption of perfect knowledge. Still, as predictive theories, such attempts fall well short of empirical scrutiny. In the late 1940's, Herbert Simon offered his theory of decision-making that has proved the most durable modification of the rational choice model.

The basic premise of Simon’s model is that decision makers search only for such information and examine only such alternatives and employ only such decision criteria to produce a result that is “good enough” in some behavioural (or emotional) sense. Limited (or bounded) rationality (commonly referred to as “satisficing”) explicitly attempts to model the costs of deciding. It recognizes that, in human terms, memory and comprehension are limited. Communication is not seamless. An important implication from this model is the use of templates and heuristics as means of re-using previously discovered or developed information. Decision makers explicitly or implicitly “match” conditions they face with those they faced before, or at least been made aware of. This can lead to reasoning-by-analogy and all the pitfalls of that approach. In a case study of US involvement in Vietnam, Khong demonstrated that the psychology of analogical reasoning makes it difficult, though not impossible, to use historical analogies properly. (Khong, 1992) One of the more crucial limitations of this approach is a pre-disposition towards a decision. The new “facts” of the situation must overcome the hard-wired “facts” of the previous experience.

Bounded rationality as a model of decision-making offers several advantages and disadvantages as a model for collegial decision-making. The advantages are that it explicitly deals with uncertainty and costs of decisions. The disadvantage is that through the use of schemas, it pre-disposes towards decisions and, in a group setting, requires at least some sense of shared understanding of the underlying schema.

3.1.3 OODA

Col John Boyd, USAF (ret.) developed his theory of decision-making based upon his experiences as a fighter pilot during the Korean War. In trying to answer why US pilots achieved such high kill-to-loss ratio over the Chinese pilots, despite flying aircraft that were only marginally superior to the Soviet-made ones the Chinese employed, he argued the US pilots could process what they saw (observations), match that against stored schema (orient), decide, then act on those decisions faster than their opponents. OODA was born. Over the subsequent years, Boyd extended this model to cover all decision-making circumstances.

Boyd's theory has much strength beyond its wide acceptance. Most important is the emphasis on orientation. Classic rational actor models and reasoning by analogy models share the common pitfall of ignoring context. Decision models such as plan-decide-execute or assess-plan-execute also can lead one into that trap. By stressing the need to orient, and especially by closely examining the elements that make up our mental images (experience, cultural traditions and genetic heritage) that provide the basis for our orientation, Boyd makes explicit the contextual underpinnings of those images. The other strength found in his orientation phase is the emphasis on analysis (or deconstruction) then synthesis (or reconstruction). While this might strike some as very Hegelian or Marxian, it presents a useful, structured approach to problem solving. As Martin van Creveld traced over nearly 4,000 years, technology makes warfare a very complex enterprise. (Van Creveld, 1991) Even with the best databases and tools, it is excessively much to expect any single person to grasp it all. Therefore, any technique that allows breaking the massive problem up into simpler ones without losing the interactions and dependencies of the whole is very useful.

While not often seen, Boyd's model includes implicit guidance and control loops as well as explicit feedback loops among every element in his model. This is a strength over simple general systems theory models of input-process-output. First of all, the implicit loops are often as important (perhaps more so) than explicit loops. For example, during the Bosnian crisis that erupted in summer 1995, coalition members often went "back channel" to their defence ministries to protest actions planned by the operational commanders. Second, separating out guidance and control from feedback is important. During the Gulf War planners routinely interacted with the tactical units on capabilities and status issues to ensure those units were not incorrectly tasked. The third benefit is the almost cybernetic quality of Boyd's model through the multiple access points for guidance or feedback. Again, the low level planners in 1991 found talks directly with mission commanders and other aircrews just as enlightening (perhaps more so) than reading official reports from those units. Van Creveld refers to these exchanges, from the commander's point of view, as "directed telescopes" that allow him to burrow right to critical points without losing the richness of experience that too often gets stripped away as information percolates up the chain. (Van Creveld, 1985) This is a significant issue for supporting knowledge systems.

3.2 Collegial Decision Making Models.

A major limitation shared by the rational choice, bounded rationality, and OODA decision-making models are their unitary actor perspective. Casual observation of the real world, and major studies of critical decisions of history, shows that in most cases, decisions result from group action, not the actions of a single decision maker. This is true even in cases where ostensibly a "single" decision maker appears to make the final choice. Allison's (1971) classic study of the 1962 Cuban Missile Crisis is a case in point. Before setting out the elements that must be present in a collegial decision-making model, it is worthwhile to examine some of the pathologies—collective miscalculations--that can arise from a group decision.

3.2.1 Groupthink

One of the more famous studies of group decision-making is Janis's (1982) *Groupthink*. Lack of norms or cohesiveness, manipulation by one or more members (especially by the group leader), panic are all examples of problems Janis found in the social psychological literature relating to group behaviour. Critical were the first two: the more cohesive the group, the more likely the group rejected views seen as nonconforming to the group's norms. These norms arise from the tendency of groups to evolve ways of preserving friendly intergroup relations. Groupthink, then, is defined as "a mode of thinking that people engage in when they are deeply involved in a cohesive in-group, when the members' strivings for unanimity override their motivation to realistically appraise alternative courses of action." (Janis, 1982: 9)

The addition of this mode to any decision-making model has profound implications for the topic of this paper. Most basically, coalition operations are occasions when groups are formed explicitly for issues that require deep involvement, usually some diplomatic crisis where the use of military force is a real possibility. Second, where unanimity is rarely explicitly required, the very nature of a coalition—ad hoc, specific issue oriented—causes unanimity to become the de facto guiding principle. The fragility-cohesiveness spectrum becomes the overriding motivation. Nevertheless, what is missing from Janis's definition, and the focus of this paper, is the shared representation that underlies the group. It is the "strategic culture" (Gray, 1996: 84) each brings, whether individual or nation, to the table.

3.2.2 Collegial models

We are now in position to describe the key elements required for a collegial decision-making model. First, to reiterate, the use of "collegial" represents the idea that the group shares, at a minimum, a goal. There indeed may be many goals; but lacking at least one, this model is useless. Second, the model must take into account the decision-making process whether described as "plan, execute, assess" or "decide, detect, deliver, assess" or "OODA." This paper uses OODA to describe the decision-making process and reserves "plan, execute, assess" for the functions the decision-making process is undertaking.

Most importantly, any decision-making model must incorporate and describe the belief structures and models of causality that reside, explicitly or not, within each member of the group. As described more fully in the next section, it is the points of tangency between group members in these areas that will constitute the "degree of shared-ness" of the group. This "shared-ness" can be thin, thick or mediated. There will be no attempt made to provide precise bounds on those categories. However, there are some guidelines available. Lacking any point of tangency in one of the two areas—belief structure or models of causality—constitutes at best a thin degree of sharing. The presence of intervening structures or processes automatically makes the sharing mediated since there is at least one other level of bargaining that must be accounted for. Observationally, coalition members are more likely to have mediated shared awareness than the other two types simply because most coalitions of interest to this paper are ones consisting largely of military forces of state actors. Finally, these points of tangency can exist "vertically" along an axis consisting of numerous groups. In NATO, for example, it could stretch from the Military Committee all the way down to individual flights within a package. For simplicity, this paper considers a "generic" group called "leadership."

4 Section 4 A Collegial Decision-making Model

Any description of leadership requires pointing out two crucial capabilities. One, it must be aware of its circumstances or situation and secondly, it must make decisions. The actions that result from those decisions are evidence of the behaviours of the decision maker. Hence, the friendly commander seeks to understand in order to influence those actions as the means of attaining the established desired effect.

4.1 Situation Awareness

Situation Awareness (SA) is defined as the perception of the elements in the environment within a volume of time and space, the completeness of their meaning, and the projection of their status in the near future. (Endsley and Jones, 1997: 17) The definition postulates three levels. Level 1 consists of perceptions. Level 2 consists of comprehension and Level 3 consists of projection. Levels 2 and 3 are crucial to decision-making. They provide knowledge and understanding of the environment to the decision makers through a cognitive hierarchy. Endsley and Jones make use of Boyd's Observe-Orient-Decide-Act (OODA) model and note that SA applies mainly to the Observe and Orient phases. (Ibid. 19-20)

An important consideration in the use of SA is the role played by models and schemata as a means of recognition priming during the Orient phase. These "provide guidance on the critical features of the environment that should be attended to and for the integration and comprehension of that information and the projection of future states, either directly or through related situation prototypes." (Ibid. 24) The models allow for decision making under conditions of incomplete or uncertain information. They provide default information. This is important since the manipulation of an adversary's risk can be a useful means of attaining changes in behaviour. For this note, we assume uncertainty equals risk to a commander. This may not always be so. Another view is that risk for a decision is where each option in the set of possible outcomes has a known probability. Uncertainty is where those probabilities are not known. (Kimminau, 1998: 22, fn 6)

These schemata can also be a source of vulnerability if the information being fed into these mental models is inconsistent with reality leading to incorrect decisions. This mis-orientation is an important element in Boyd's OODA model. This mis-orientation need not be the result of mis-information. It may be beneficial to provide accurate, but unexpected information. This could lead to cognitive dissonance. This plays upon the tendency for individuals to seek consistency between attitudes and behaviours. (Festinger, 1957) When forced to choose between incompatible beliefs or actions, dissonance occurs.

4.2 Framing

Recognition priming (RP) is a means of framing context for decision-making. The importance of framing is the key insight of prospect theory and distinguishes it from the classic rational actor model (RAM) of decision-making. An important consideration must be addressed. Prospect theory presents a richer view of decision makers and hence relies upon much more information about them than a RAM approach. Therefore, if such information is not likely to be available, an EBO approach that utilizes prospect theory is unlikely to succeed. Regardless of the approach employed, this "enhanced information content requirement" has important impact on knowledge systems that support planning, execution, and assessment activities.

Information about the adversary's decision-making apparatus comes from two sources, both related to the four-stage Intelligence Preparation of the Battlespace process. The first is the centre-of-gravity (COG) and target systems analysis (TSA) done during the third stage. Several models are available. The ones in the EBO CONOPS (McCrabb, 2002) derive from those developed by Warden and Barlow. Regardless of the model used, the important information derived is an understanding of the elements from which the adversary derives freedom of action, physical strength, or the will to fight. (HQ USAF/XO, 1999)

The second source of information comes from stage four of IPB that postulates enemy courses of action (COA). Again, it is an assumption of the SA-RP model presented here that behaviours could be derived from actions. Therefore, by postulating a series of enemy actions, that is, a COA, planners are predicting a set of behaviours. Using a wargame, planners can then play out Blue COA options and Red COA options in an interactive and iterative game. The goal is not precise estimates but rather general tendencies.

4.3 Model

Figure 4 is the complete model. The following subsections describe each element in some detail. "Complete" may be somewhat misleading. The areas of implementation and feedback are not described in much detail. The focus is on the actors and their interactions. Within the actors, the focus is on belief structures as means of framing (or orienting) and the use of models of causality. The latter are not described at all. The use of this collegial decision making model, for example in an effects-based approach to planning, executing, and assessing a military operation by a coalition, would dictate exactly which causal models would be of interest. It is hoped the examples used will relay that flavour.

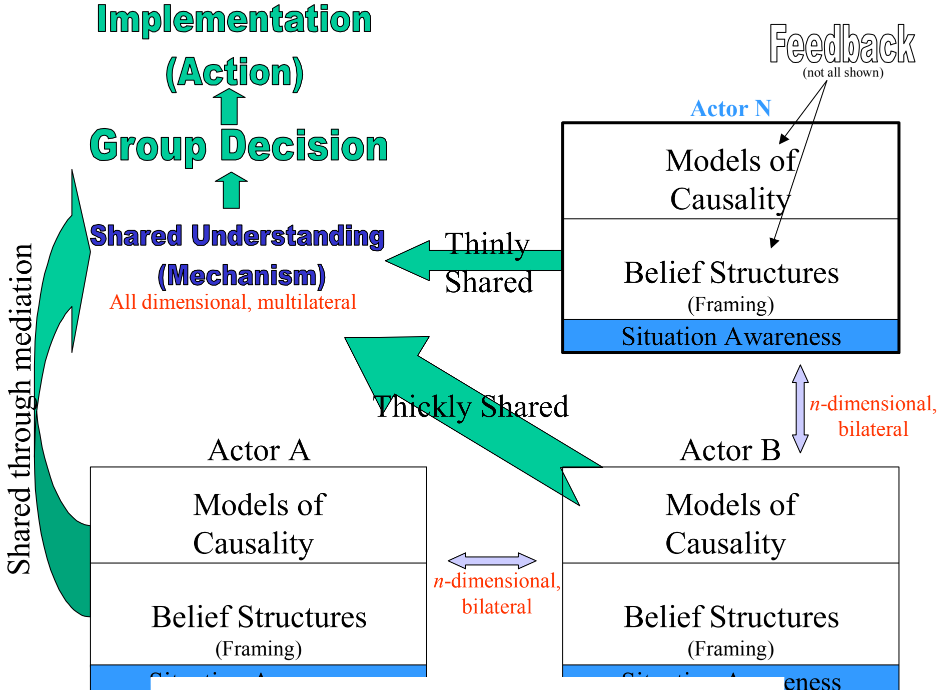


Fig. 4: A Model of Collegial Decision-making

4.3.1 Belief Structures

One way to view the internal schema of an actor is that belief structures constitute the values assigned to individual variables while the causal models constitute the relationships between the variables. However, that would assign much too much a boundary between the two. Concentrating solely on "actor *n*" the thickness of the lines around the actor and the relative thinness of the lines between situation awareness, belief structures, and causal models is supposed to relay the fact that internally the lines are much more permeable and translucent that the exterior lines. Beliefs themselves have relationships. This model uses Gray's notion of "strategic culture." That is the "socially transmitted habits of mind, traditions, and preferred methods of operation that are more or less specific to a particular geographically based community." Strategic culture incorporates expressions of strategically adaptive reasoning behaviour. (Gray, 1996: 84)

Preferences and adaptive reasoning are the critical elements in framing. It is how the actor mediates the "raw data" arriving from situation awareness (itself filtered "raw observational data"). Within belief structures, the point of emphasis is on preferences: the set of outcomes, or conditions, the actor prefers to see occurring. It has both the traditional positive element ("prefer to see") and negative element ("prefer not to see"). An example might be an actor's preference that an adversary accedes to one's demands yet without the actor having to cause extensive harm to the adversary. Some argue that is precisely the view NATO wished during Operation ALLIED FORCE, the campaign against Yugoslavia in 1999.

4.3.2 Causal Models

Causal models are used by the actor to make some prediction about how actions that might be taken in the group's name are likely to produce some outcome. By their very nature, causal models are probabilistic. In one sense then, the points of "shared-ness" between actors can be characterized as thick or thin based upon the various probabilities each actor assigns within a given causal model. During ALLIED FORCE, for example, some actors disagreed with the assertion made by the Air Component Commander, USAF Lt General Mike Short, that the best way to stop the Yugoslavians from forcing the Kosovar Albanians from their homes was to bring the war to the folks living in Belgrade. Most of the national representatives adhered to a more traditional view that if the source of the ethnic cleansing was the Yugoslavian military and paramilitary forces within Kosovo, then they were the right targets to attack, not electrical power plants that supplied Belgrade.

Besides showing dependencies, causal models play another important role for group decision-making. They are used to wargame potential course-of-action (COA) the group might develop. By the addition of some thought on what an adversary might do, the group can "play out" the opposing schemes as a way of predicting outcomes. War games can show where an actor's (or the group) situation awareness is lacking hence become a source of investigation. It can also highlight out intervention points, which is where the group, or its agent, might have to intervene in a plan and make adjustments based on the adversary's reactions to the group's COA.

4.3.3 Shared Understanding

Shared data is a necessary but not sufficient condition for shared awareness. Indeed, shared awareness is insufficient to achieve shared understanding. To move from shared data is the role of knowledge systems. The move from awareness to understanding requires much more. It requires understanding the strategic culture of one's coalition partner. As pointed out above, the most important elements in the strategic culture are the predisposition to causal mechanisms, risk proclivity, and belief structures. Techniques for overcoming or mitigating these are beyond the scope of this note. However, wargames, exercises, and other educative activities are the traditional means. These tend to work well for long-standing alliances such as NATO. Whether these techniques would work when faced with the ad hoc nature of "pick up" coalitions such as the one formed to combat global-reach terrorism is more problematic.

4.4 Implications for EBO

Warfare rarely is only about breaking things or killing people. The goal is to affect some sort of change in the opponent's behaviour. Generally that occurs either through brute force means, such as annihilation or attrition, or coercion. In terms of US Joint doctrine, the military aim, at root, is to set the conditions where other instruments of national power—normally political-diplomatic—can take over and attain the strategic aim. War really is politics by another means. To establish these conditions, military commanders must have some understanding of the behavioural effects their actions accomplish. This is true for the operational as well as the strategic levels of war. An isolated battlefield or halted military force has a significant behavioural component.

The challenge for the commander is to trace effects of various actions throughout the enemy to understand what overall affect is taking place. This requires models or knowledge representations that show linkages between physical actions and behavioural effects. Since one of the most important duties of a commander is decision-making, these models must anticipate the decision-making process of the adversary and ideally be adaptable to the decision-making proclivities of our own commanders.

5 Section 5 Implications for Knowledge Systems

5.1 Targeting

5.1.1 COGA

Perhaps the greatest need for knowledge systems lies in the area of targeting. Within that area, the most critical need is to support centre-of-gravity (COG) analysis. Clausewitz wrote "one must keep the dominant characteristics of both belligerents in mind. Out of those characteristics a certain centre of gravity develops, the hub of all power and movement, on which everything depends." (1976 [1832]: 595-

596) He lists five cases: in most it is military forces, it is the capital city where the enemy faces internal strife, the COG is allies and their military forces for small countries, and it is the personalities of the leaders and popular opinion where there is a popular uprising. Early airpower theorists expanded the scope of a COG. US Army Air Service Brigadier General William A. “Billy” Mitchell included “centres of production of all kinds, means of transportation, agricultural areas, ports and shipping; not so much the people themselves.” (1988 [1925]: 16) Italian General Giulio Douhet included “industrial and commercial establishments; important buildings, private and public; transportation arteries and centres; and certain designated areas of civilian population as well.” (1983 [1921]: 20) In each case, elements—called target systems—combine in unique ways to form COG. Knowledge systems are required to form, and understand, those combinations.

5.1.2 TSA

Joint doctrine defines a target system as “1. All the targets situated in a particular geographic area and functionally related. 2. A group of targets which are so related that their destruction would produce some particular effect desired by the attacker.” (JP 1-02) This notion of “relatedness” or system is essential to what is presented here. Instructors at the US Army Air Corps Tactical School (ACTS) in the 1930s emphasized systems analysis to their students. They focused on “major industrial and economic systems for production of weapons and supplies for their armed forces, and for manufacture of products and provision of services to sustain life in a highly industrialized society.” (Quoted in Faber, 1997: 217) Most importantly, they focused on the connections and dependencies between and within these systems that formed an “industrial web” where attacks against one element in the web would ripple throughout the web causing more problems then just the immediate damage done. From COG and target systems analysis, course-of-action (COA) options are developed. This is the second great area in which knowledge system support is crucial.

5.2 Strategy and COA Development

A COA is defined as “a *plan* that would accomplish, or is related to, the accomplishment of a mission.” It is also defined as “the scheme adopted to accomplish a task or mission.” Furthermore, “when approved, the … [COA] becomes the basis for the development of an operations plan or operations order.” (JP 1-02) There are several conceptual definitions closely related to COA. A concept of operations, within the same context and sub-contexts, “describes how the [Joint Force Commander] visualizes the operation will unfold based on the selected COA. This concept expresses *what*, *where*, and *how* the joint force will affect the enemy or the situation at hand.” (JP 3-0)

The end state, goal or objective is *what* is to be accomplished, purpose or rationale provides *why* the goal is sought, the plan or sequence of actions is *how* the goal or objective is going to be accomplished, and resources are the wherewithal (or “*with*”) for the plan. Specifying *who* will accomplish the actions and *where* and *when* in the sequence completes the COA. Military strategy is defined as the art and science of employing forces to secure objectives by the application of force or the threat of force (adapted from JP 1-02). A campaign, or operational-level, plan is defined as a series (or sequence) of related operations (or actions) aimed at accomplishing an objective within a given space and time (adapted from JP 1-02).

5.3 Wargaming

The third large area in which knowledge systems are needed to support effects-based coalition operations is in wargaming. EBO requires real or near-real time operational level wargaming of Blue versus Red COA. Development is sorely needed to build a robust, computerized operational level wargaming tool. This tool can take Blue COA options such as those generated by the Air Force Research Lab's Strategy Development Tool (SDT) and wargame them against Red COA options generated from some IPB tool or process. Today, COA versus COA wargaming if done at all, is done on paper using situation and event templates. Most computerized wargaming tools such as STORM (Synthetic Theatre Operations Research Model) have a force-on-force, target-attrition emphasis. Though they do support and analyse higher level objectives such as *establish air supremacy*, *defeat warfighting forces*, or *disrupt enemy leadership*; they are not adequate to satisfy EBO wargaming requirements.

Wargaming to support Effects Based Operations has to account for criteria related to both friendly and adversary COAs. Adversary COAs are derived based on the process defined in Joint Publication 2-01.1, *Joint Tactics, Techniques, and Procedures for Joint Intelligence Preparation of the Battlespace.* Determination of adversary COA is the last step in a four-step process. This final IPB step includes:

- Identifying the adversary's likely objectives and desired end states,
- Identifying the full set of COA available to the adversary,
- Evaluating and prioritising each COA, and
- Developing each COA in the amount of detail time allows.

The process in JP 2-01.1 needs to be computerized with an explicit focus on EBO. For example, the doctrine prescribes the use of psychological profiling of adversary leaders to determine their acceptable level of risk; but EBO will require broader cognitive modelling and behavioural analysis of not only warfighting decision making commanders, but also of political leadership and the general population. Friendly COA built using AFRL's SDT tightly link commander's intent (objectives) to desired effects. The focus is explicitly on physical and behavioural effects including direct, indirect, cumulative, and cascading effects. Centres of gravity and target analysis are used to identify targetable actions necessary to achieve the effects desired. Existing computerized wargaming tools are limited in that they do not address the interplay of various COA in a simulated environment nor do they appropriately deal with effects. Most of these are highly robust when it comes to engagements (e.g., tanks against tanks or aircraft against armour forces) but are quite thin at the campaign level and of little use in evaluating an operational-level COA. (McCrabb and Caroli, 2002)

6 Section 6 Conclusion

This research note presented some preliminary thoughts on how effects-based operations impacted classic military decision-making and how collegial decision-making, such that characterizes coalitions, places further demands on supporting knowledge systems. This is particularly so in the areas of targeting, where not only first order physical effects but n-order behavioural effects must be traced; course-of-action (COA) development where the many effects intermingle and produce cumulative effects; and wargaming where such plans are "played out" against various evaluative criteria.

Each of these elements influences group decision-making structures. Each actor is immersed in a strategic culture with the key elements of belief structure, causal models, and certain risk proclivities. Targeting and COA development affect the first two; wargames provide one means of at least making explicit the latter.

Collegial decision-making in a military environment is an area ripe for further research. One approach is agent-based modelling where distributed; decentralized decision-making could be examined to see to what degree dynamical structural behavioural outcomes are predictable. Classic chaos theory, with its emphasis on the sensitivity of initial conditions, seems to argue that such behaviours are unlikely to be predictable with any sufficient degree of certainty to overcome a wide range of prudence normally associated with the employment of force. On the other hand, work in complex adaptive systems theory seems to offer the promise of a higher degree of predictability. (Holland, 1995; Alberts, Garstka, and Stein, 1999; Czerwinski, 1998)

References

Alberts, D.S., Garstka, J.J., and Stein, F.P. (1999) "Network Centric Warfare," US Department of Defence C4ISR Cooperative Research Program, Washington, DC, USA

Allison, G.T. (1971) "Essence of Decision," Harper Collins, New York, NY, USA

Clausewitz, C. (1832) "On War," reprint Princeton University Press, Princeton, NJ, USA

Czerwinski, C.J. (1998) "Coping with Bounds," Institute for National Strategic Studies, National Defence University, Washington, DC, USA

Davis, P.K. (2001) "Effects-based Operations: A Grand Challenge for the Analytical Community," RAND, Santa Monica, CA, USA

Dixit, A.K. and Nalebuff, B.J. (1991) “Thinking Strategically,” W.W. Norton, New York, NY, USA

Douhet, G. (1921) “Command of the Air,” reprint Office of Air Force History, Washington, DC, USA

Endersley, M. and Jones, W.M. (1997) “Situation Awareness Information Dominance and Information Warfare,” Air Force Research Laboratory, Wright-Patterson AFB, OH, USA

Faber, P.R. (1997) “Interwar US Army Aviation and the Air Corps Tactical School: Incubators of American Airpower,” The Paths of Heaven: The Evolution of Airpower Theory, Air University Press, Maxwell AFB, AL, USA

Festinger, L. (1957) “A Theory of Cognitive Dissonance,” Stanford University Press, Stanford, CA, USA

Gray, C.S. (1996) “Explorations in Strategy,” Greenwood Press, Westport, CT, USA

HQ USAF/XO (1999) “Intelligence Preparation of the Battlespace: An Airmen’s Introduction,” White Paper, February 1999

Holland, J.H. (1995) “Hidden Order,” Perseus, Reading, MA, USA

Janis, I.L. (1982) “Groupthink,” Houghton Mifflin, Boston, MA, USA

Joint Pub 1-02 (2001) “Dictionary of Military and Associated Terms,” US Department of Defence, Washington, DC, USA

Joint Pub 2-01.3 (2000) “Joint Tactics, Techniques, and Procedures for Joint Intelligence Preparation of the Battlespace,” US Department of Defence, Washington, DC, USA

Joint Pub 3-0 (2001) “Doctrine for Joint Operations,” US Department of Defence, Washington, DC, USA

Khong, Y.F. (1992) “Analogies at War,” Princeton University Press, Princeton, NJ, USA

Kimminau, J.A. (1998) “The Psychology of Coercion: Merging Airpower and Prospect Theory, Maxwell AFB, AL: School of Advanced Airpower Studies

March, J.G. (1994) “A Primer on Decision Making,” Free Press, New York, NY, USA

McCrabb, M.J. (2001) “Explaining Effects,” Air Force Research Laboratory, Rome, NY, USA

McCrabb, M.J. (2002) “Concept of Operations for Effects-based Operations,” Aerospace Command, Control and Intelligence, Surveillance, and Reconnaissance Centre, Langley AFB, VA, USA

McCrabb, M.J. and Caroli, J.A. (2002) "Behavioral Modeling and Wargaming for Effects-Based Operations," Military Operations Research Society, Washington, DC, USA

Mitchell, W. (1925) “Winged Defence,” reprint Dover Publishing, New York, NY, USA

Van Creveld, M. (1985) “Command in War,” Harvard University Press, Cambridge, MA, USA

Van Creveld, M. (1991) “Technology in War” Harvard University Press, Cambridge, MA, USA

Requirements for Standards and Commonality with Regard to Knowledge Based Systems for Coalition Operations

Alice M. Mulvehill
BBN Technologies
10 Moulton Street
Cambridge MA 02138 USA
amm@bbn.com

Ted Kral
BBN Technologies
9655 Granite Ridge Drive, Suite 245
San Diego, Ca. 92123 USA
tkral@bbn.com

Abstract

The theme of the paper is: what is the impact that knowledge based systems for Coalition Operations have on the requirement for standards and commonality? Do knowledge based systems mitigate or compound the need for standards? The authors have fifteen or more years of experience in research and development of knowledge based prototype systems for use by diverse groups and virtual organizations. In all these initiatives, the degree and type of standardization became an issue. There were various approaches taken to satisfy the need and/or desire for standards, such as, common environments, common plan representation, common planning process, common hardware, common user, etc. The paper presents the authors' view on several of the techniques used, lessons learned, and the applicability to the domain of Coalition Operations. Insights are also provided into cognitive issues based on culture with regard to terminology, training, operational concepts and planning processes.

Introduction

It is not our intention in this paper to debate the issue on whether the use of standards is good or bad nor whether they are necessary or not in the development of computer software. Our intention is to report on the role that standards played in several major decision support programs and the relevance to knowledge based systems for Coalition Operations.

BBN Technologies has done extensive work in the area of communication, crisis planning, transportation and information assurance. BBN Technologies has developed a number of knowledge based decision support systems to support the various aspects of military planning. Our expertise includes the design and development of independent systems as well as the integration of heterogeneous systems in support of military exercises and/or demonstrations. In addition, our support of demonstrations like the Joint Warrior Integration Demonstration (JWID) have stressed intercommunication between disparate systems, collaboration among distributed planning teams, data sharing in multi-security environments, and planning coordination with coalition partners.

In our experiences, there have been efforts to provide some standard platforms, common operation infrastructures, and common terminologies in order to facilitate communication and collaboration in a distributed environment. The role of these standards ranged from the provision of linkages between two disparate systems through the usage of mapping tables to the development and usage of common schemas (plan representation), common planning workflow processes, and common ontologies. Figure 1 suggests that there exists a correlation between the degree of closeness between two entities (be they human or software system) and a tendency to share a common terminology. For example, when two systems, developed by separate contractors need to communicate, a simple mapping table like the one provided in Table 1 can be used to bridge the gap between the terms used to refer to a concept or process in one system with the terms used to refer to those same concepts or processes in the other system. This method works well when the two systems do not need to (or do not believe that they need to) communicate or collaborate often. As the need to work together increases, so increases the need for a more standardized and extensive communication environment.

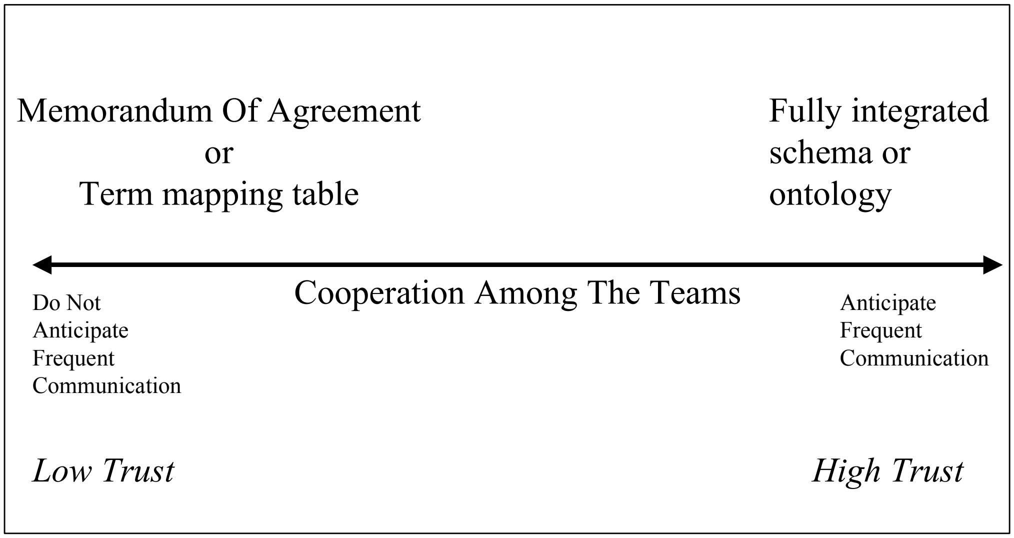

Memorandum Of Agreement
or
Term mapping table
Fully integrated schema or ontology
Cooperation Among The Teams
Do Not Anticipate Frequent Communication
Anticipate Frequent Communication
Low Trust
High Trust

Figure 1 – Communication Continuum

The ARPA/Rome Lab Planning and Scheduling Initiative (ARPI) Experience

This was a large Joint DARPA and Rome Laboratory initiative stretching over more than five years. It also involved a large number of prominent researchers and organizations in the field of planning and scheduling. As such we could not due justice in the space allotted to fully report on this effort. Instead those interested readers are directed to the reference article by Austin Tate (Advanced Planning Technology, Technological Achievements of the ARPA/Rome Laboratory Planning Initiative, AAAI Press, 1996).

Points to be stressed are that this initiative did explore many aspects of the planning domain and the supporting technologies including standards. Effort was devoted toward the development of a common environment to conduct experiments. Emerging from this were concepts of Technology Integration Experiments (TIE) and the process of the Integration Feasibility Demonstrations (IFD). Additionally, there was a considerable amount of effort applied to the selection of standards and common tool use to promote interoperability between various technologies used for automated and semi automated (mixed-initiative) planning (user in the loop). One significant pursuit that this program devoted a significant amount of program time and resources to was in the development of a “Common Plan Representation”.

The Joint Task Force Advanced Technology Demonstration (JTF-ATD) Experience

This again was a significantly large effort for which we could not due justice in describing in the space allotted. Again, *references* are provided at the end of this paper for those interested in gaining more insight into this initiative. The program was intended to capitalize on the results of the ARPI effort and to develop a distributed planning environment based on a linkage of supporting functional planning cells called anchor desks and the operational planning cell. While the ARPI initiative explored standards and basically followed a “de facto” standards policy, the JTF-ATD effort stressed the enforcement of standards centered around the CORBA technology and the concept of a series of web based object servers. The intent was to separate application development from concern regarding the mechanics of interoperability and accomplish that through the use of servers with a common interface and schema. This effort also pursued the development of a common plan representation in the form of a common plan object. A considerable investment of this program was devoted to training individual development groups on the standards and also enforcement of these standards when software was delivered. Figure 2 is a graphical representation of the JTF Reference Architecture standard.

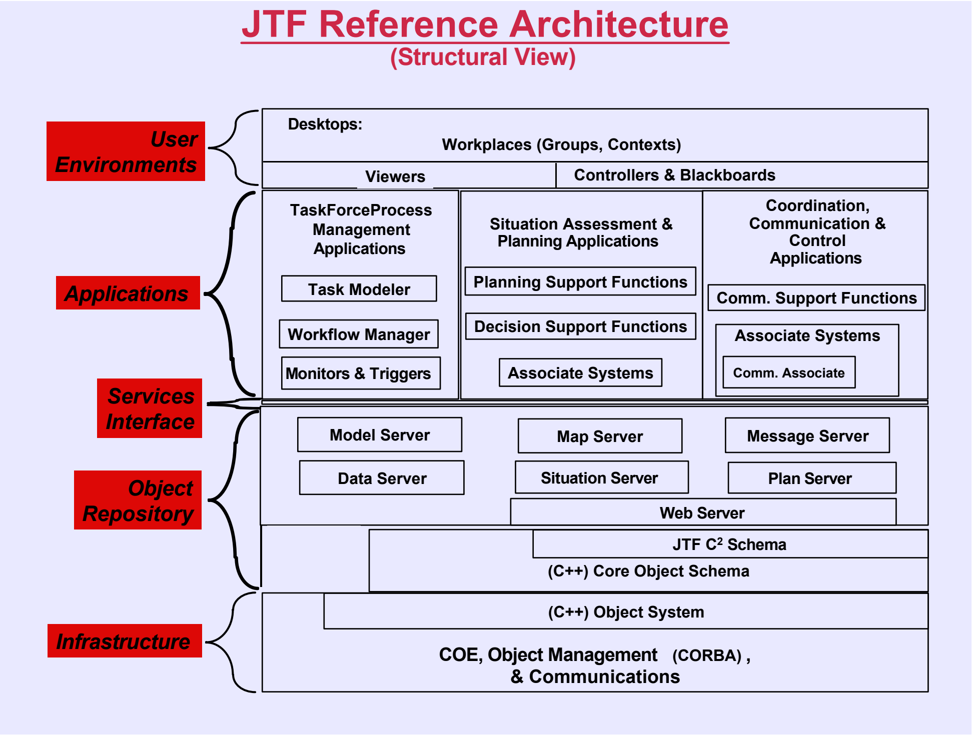


Figure 2 - JTF ATD Reference Architecture

The ACOA Experience

Since this is a more recent program and supposedly builds from lessons learned from previous endeavors, we will spend more time on this experience. BBN was one of the key developers of components of the AITS-JPO Adaptive Course of Action (ACOA) ACTD. The goal of ACOA is to demonstrate advanced technology to help develop multiple deployment scenario courses of action. The objective of ACOA is to include its capabilities under the Global Command and Control System (GCCS)

The ACOA ACTD is based on a user-centric, iterative development philosophy, following a rapid application software development lifecycle. The primary user is located in USPACOM and provides operational feedback on ACOA capabilities. ACOA has been tested for military utility as part of military command post exercises—the most recent during Ulchi Focus Lens 01.

The ACOA ACTD (see Figure 3) consists of several integrated knowledge based tools, including: The WebPlanner, for which BBN is the prime developer, is an integrated system that includes the Operations Planning Tool (OPT), Course of Action Selection Tool (COAST), Force Management Tool (FMT), Joint Assistant for Deployment and Execution (JADE), and TURBO PLANNER. OPT provides planning process templates used to assemble and share critical plan information and generate key military plans, orders and messages. COAST employs fuzzy logic technology to assist in developing and comparing alternative courses

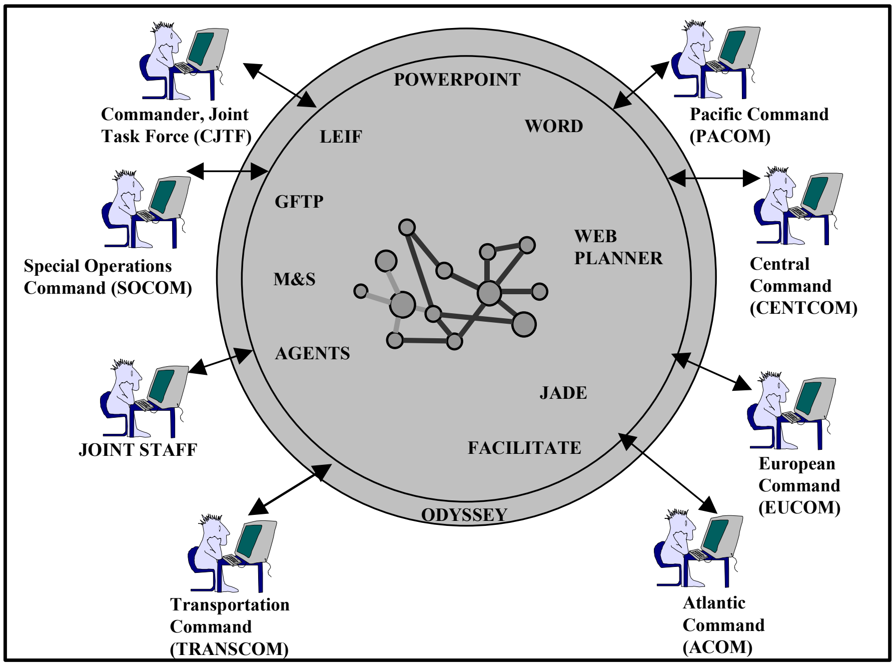


Figure 3 – Collaboration within the ACOA Environment

of action. FMT adds capability to identify ready and available forces, task organize forces to specific missions, specify deployment destinations, and time-phase forces for deployment. JADE provides a suite of tools to match specific force capabilities with required tasks and quickly generate time-phased force and deployment data using pre-defined force packages and “Drag-and-Drop” technology. In ACOA, these tools can be operated by multiple distributed planners via the Campaign Object Schema.

To illustrate how the needs for two systems (or human planners) can change over time, we will now describe how the interoperability of two of the ACOA components (The WebPlanner and JADE) evolved over time. Both of these systems were involved in a previous Technical Integration Experiment (TIE) during the DARPA ARPI program under their previous names of Target and ForMAT. In the ARPI TIE, while there was not any anticipated notion that the two systems would communicate with each other on any regular basis, the TIE was intended to allow the system Target to make queries against the ForMAT system for information about how forces were deployed in previous, but similar planning contexts. Table 1 shows a piece of the data mapping table that was established by the developers in order to allow these two systems to communicate. The term on the left is the term used in Target, and the term on the right is what that concept is called in ForMAT. The data mapping table was required because neither system was inclined to change its terminology.

```
(“OPERATION NAME” mission)
(“AREA OF RESPONSIBILITY” geographic-location)
(“SUPPORTED CINC” theater)
(“FORCE CAPABILITY” function)
(“FORCE SERVICE” service)
(“FORCE UIC”      uic )
(“A” army)
(“F” air-force)
(“M” marines)
(“N” navy)
```

Table 1 – Term Mapping Table

During ACOA there was a requirement for all systems, including the WebPlanner (the successor of Target) and JADE (the successor of ForMAT) to collaborate with each other using a common schema and a common Campaign Object Server. Iinstead of a data mapping table, common data is stored in the ACOA Campaign Object for use by any tool that understands the Campaign Object Schema. Figure 4 illustrates how JADE uses this

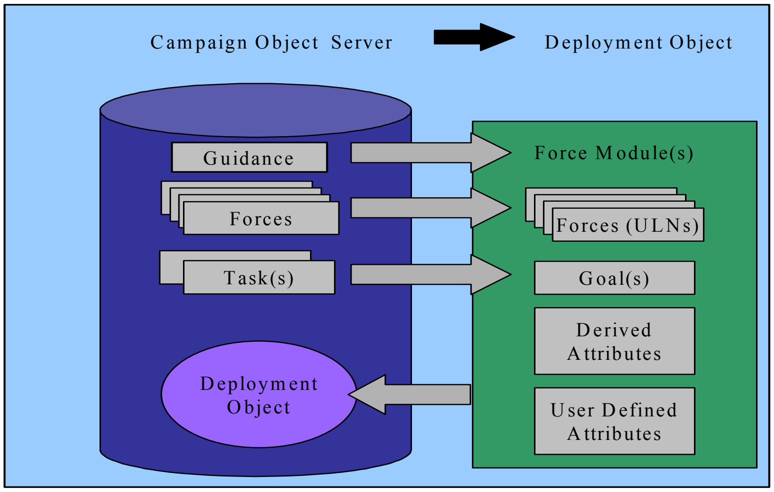


Figure 4 – Deployment Plan Development and The ACOA Campaign Object Schema

data to develop the Deployment Plan. You will still notice that a few term inconsistencies still exist, e.g., between tasks and goals. This means that the JADE system has to do some of its own translation in order to maintain its own processing capability while interacting with others. We believe that there are lessons to be learned from the communication history of these two systems that will apply (by analogy) to multi-national coalition team formation and development

Using Ontologies

The data mapping table in Table 1 is a simple instance of ontology mapping. Ontologies are being developed as part of the DARPA DAML (Darpa Agent Markup Language) program to better enable software agents to read text. Software agents and agent teaming methods are being developed as part of the DARPA CoABS (Control of Agent Based Systems) program to allow for the rapid formation of mixed-initiative agent based systems in response to some crisis or threat (for more information, see Burstein, M., Mulvehill, A., and Deutsch, S. 1998). BBN is involved in both of these programs. BBN is the integrator for the DARPA DAML program where researchers are developing ontologies and tools that allow for mappings between ontologies. The ontology mapping will allow for the development of shared ontologies and common operating environments where software systems, software agents, and the human users of those systems can preserve their own terminological preferences while still communicating with others.

Our experience to date in the the CoABS and DAML programs leads us to suspect that multi-national coalition teams will require the establishment of some standard operating ontology and that ontology mapping tools will be required in order to facilitate the entry of new players into a forming coalition. We believe that the entry of new members to an existing Coalition is analogous to how ForMAT and Target worked, e.g., members of the team develop very defined expectations of what other members of the team will do. But just as the ForMAT/Target relationship evolved, so too will coalition teaming arrangements. Perhaps ontological mapping tools can facilitate that evolution.

Forming Coalitions -- Lessons Learned from JWID

A Joint Warrior Integration Demonstration (JWID) is a means to bring together multiple systems to test how well they perform together to support some planning scenario. While BBN has been involved to some extent in may JWIDs, two of the JWIDs which could provide valuable lessons learned for coalition formation were JWID-94 and JWID-95.

One of the prime objectives of JWID-94, (Figure 5),was to show evolving processes and technology for distributed collaborative planning (DCP) and how DCP tools could be used to support deliberate as well as crisis action planning for a Joint Task Force (JTF) deployment. Systems and networks that support and enhance the communications infrastructure for the JTF operation, including multi-level security were also tested.

During JWID-94, a disaster relief scenario and a combat operations scenario were used to test the usage of several tools, technologies, and systems, including: Tachyon, Advanced Planning System (APS), Force Level Execution System (FLEX), Weather Anchor Desk, Air Campaign Planning Tool (ACPT), Theater-Level Analysis Replanning Graphical Execution Toolkit (TARGET), Cronus, Force Management and Analysis Tool (ForMAT), Analysis of Mobility Platform (AMP), In-Theater Airlift Scheduler (ITAS), Rapid Application of Air Power (RAAP), Web Authoring and Management System (WebMan), The Logistics Anchor Desk (LAD), and the Targeting Management System (TMS)). For this JWID, the BBN system TARGET was used as the distributed toolbox and environment for collaboration.

The following excerpt is from the conclusions and recommendation sections of the JWID94 final report with regard to the results obtained from this exercise:

> "Tools and architecture for planning military and non-military responses to crisis situations were well represented and showed their value added in the Joint Task Force environment. The TARGET system, (Figure 6), used a shared database as a common point for planning, which thereby provided its value as a tool for organizing, weighting, and reviewing assumptions, planning factors, rationales, etc. that are used by the staff in formulating recommended Courses of action. The Air Campaign Planning Tool (ACPT) generated an Air Campaign Plan, sharing its data with TARGET and its resulting Candidate Target List (CTL) with the Rapid Application Air Power (RAAP) tool. The tools most preferred for use during DCP were video, voice, briefings, and pointers. Conferencing sessions were very successful in demonstrating the effectiveness of using distributed networking, COTS collaborative planning software, security guards, and Video Teleconferencing in concert to create a powerful conferencing environment. This capability is particularly valuable in the area of crisis management, where problems can be ill-defined, accurate situation assessment critical, and clearly communicated consultation of prime importance. Collaborative planning has

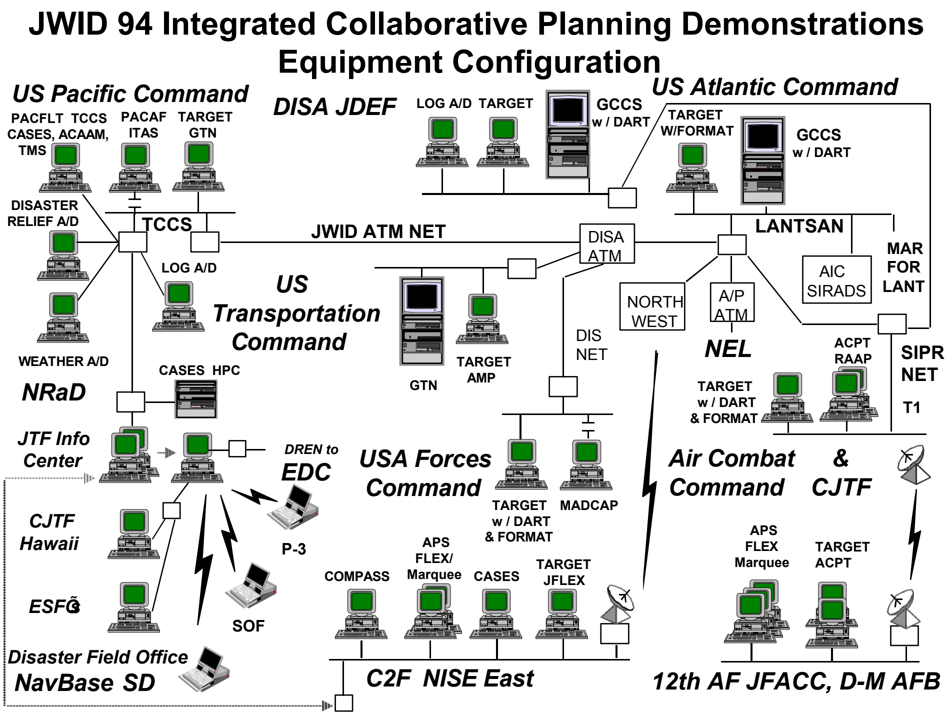
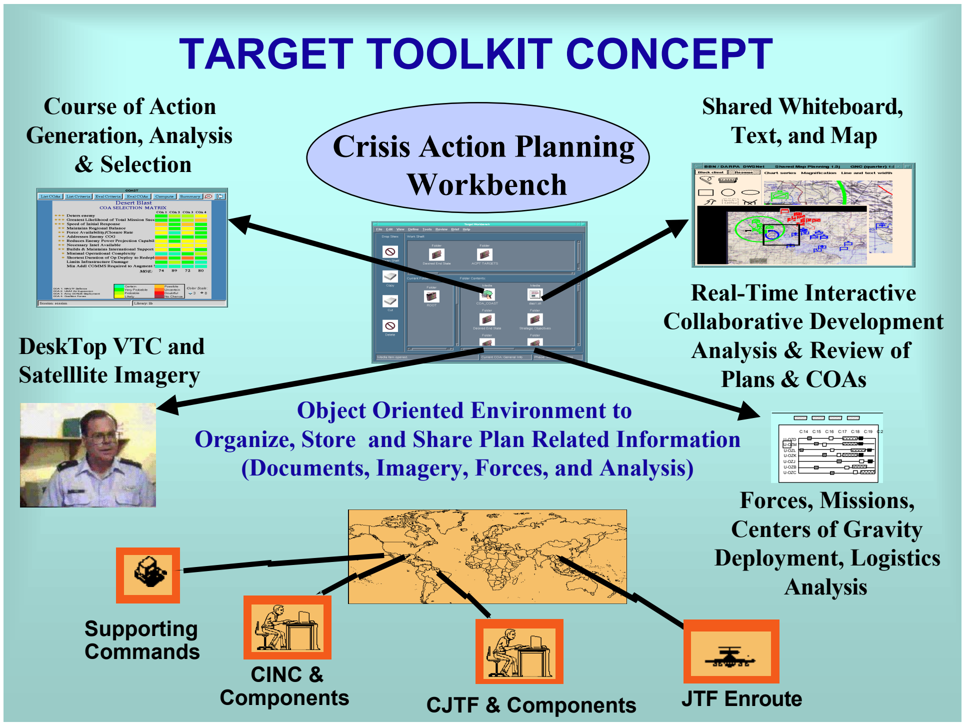


Figure 5. JWID 94 Configuration



TARGET TOOLKIT CONCEPT

Course of Action Generation, Analysis & Selection

Crisis Action Planning Workbench

Shared Whiteboard, Text, and Map

Real-Time Interactive Collaborative Development Analysis & Review of Plans & COAs

DeskTop VTC and Satelllite Imagery

Object Oriented Environment to Organize, Store and Share Plan Related Information (Documents, Imagery, Forces, and Analysis)

Forces, Missions, Centers of Gravity Deployment, Logistics Analysis

Supporting Commands

CINC & Components

CJTF & Components

JTF Enroute

Figure 6. Theater Analysis , Replanning, and Graphical Execution Toolkit (TARGET)

useful functions to make planners more systematic and objective in their planning. Additionally, the ability to share the thought process with other agencies can be a plus, provided developers implement protocols to prevent database corruption and input/output saturation.

In summary, JWID-94 results illustrated how the following factors affected distributed collaborative planning and interoperability:

- platform
- speed and efficiency of I/O between functionally related systems
- the impact of the network type on intercommunication
- the impact of environmental issues on interoperability
- collaboration between systems and among geographically distanced sites
- human collaboration techniques
- skill level of the operator." (Defense Information Systems Agency, 1994)

Could any of these lessons learned be used to develop a set of standards that could be used to support multi-national coalition formation and development?

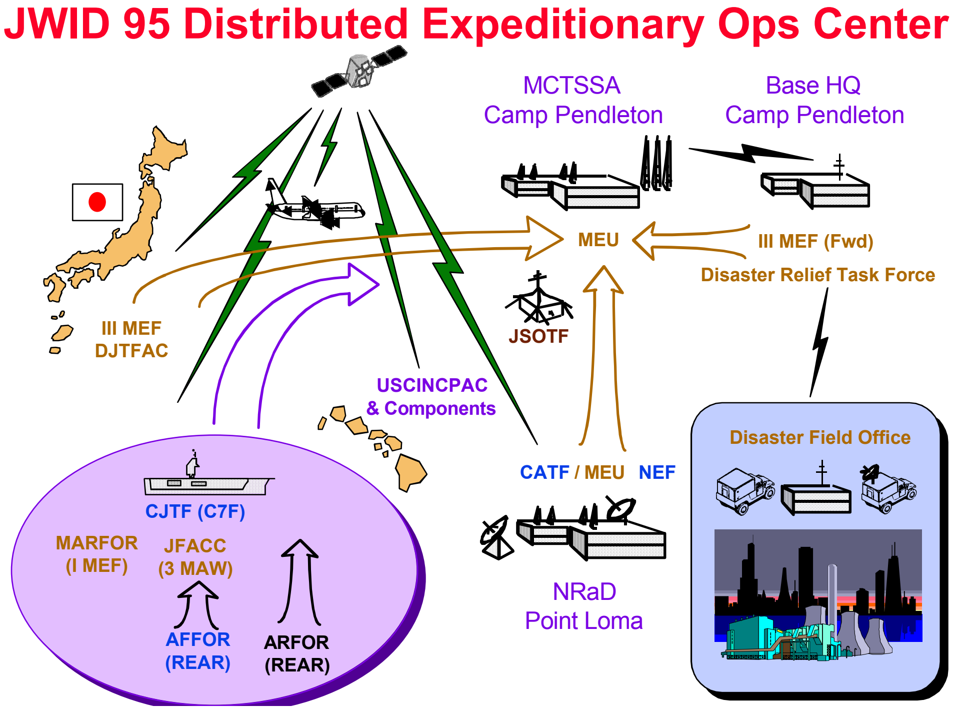


Figure 7. JWID 95 Configuration

JWID 95 (Figure 7) conducted in the subsequent year attempted to probe these areas. The following excerpts are from the final report:

> "Several overarching technology areas demonstrated in JWID 95, are changing the way that the Warfighter will share, access, process and disseminate information. World Wide Web (WWW) technology was used extensively to enhance information exchange and access. Collaborative planning tools such as whiteboards, shared applications, and on-line chat functionality provided low bandwidth solutions to sharing and collaboration. Anchor desks used these collaborative capabilities to support functional areas however, a COE is needed to enhance interoperability. For JWID 95, the Joint Staff, J6, extended an invitation to the member nations of the Combined Communications Electronics Board (CCEB) to participate. This invitation was accepted by Australia, Canada, and the United Kingdom. New Zealand, the remaining CCEB nation, initially planned an active role, but ultimately participated only as an observer. Three principle objectives for Allied involvement were accomplished during JWID. They were:
>
> - Receipt and display of US Common Operational Picture (COP).
> - Participation in the development and distribution of the US ATO.
> - Participation in the course of Action (COA) development through Distributive collaborative Planning sessions.
>
> The recommendations regarding Allied Participation, based on the JWID95 experience were that CONOPS should be developed, based on CINC requirements, for releasability of classified information to Allies. Appropriate JTF architecture documents and focus on the doctrine process, procedures and MLS systems should be provided to each participant. "(Defense Information Systems Agency, 1995)

Could any of the lessons learned from the JWID95 experience, particularly with Allied participation be used to develop a set of standards that could support multi-national coalition formation and development?

Forming Coalitions – Cultural and Social Issues

In forming a coalition, a human planner, along with his/her computing hardware and software, and perhaps software agents, will be invited to join a coalition team. The new member should be provided with a an API, process model, and some specified set of communication terminology. The size of the communication terminology provided could be based on how similar the new member is relative to existing team members. Similarity can be assessed in terms of: culture, technological sophistication, planning style, and social practices. If the new member is very similar, than he/she may be presented with a common ontology or schema. If the new member is very different, then mapping tables may need to be defined to allow them to map from their terms to the terms of the rest of the coalition.

Work by Hofstede (Hofstede, Geert, 1997) suggests that the similarity between planners from different countries can be determined from a set of dimensions. The work of Hofstede and of others like Marcus et all (Marcus, A. and Gould, E.W., 2000) who have used Hofstede's work to provide directions on how user interfaces should be designed, suggest that there is a correlation between dimensional ratings and communication and collaboration style. Perhaps, new potential coalition members can be evaluated using this method, and communication and collaboration mechanisms determined based upon their scores.

Conclusion

If one draws an analogy between the methods required to link computer systems and applications together to the methods needed in order to link multi-national human planners together, then the lessons learned from an attempt to link a number of heterogeneous systems together to participate in the programs we described in this paper can be used to support the development of multi-national coalition teams. Additionally, the use of standards appears to be related to the interoperability one desires in the functionality or the operation of the software applications. Another factor is whether or not the concept of development involves the independent development of heterogeneous components which are then integrated as pieces to form larger integrated software applications or systems.

In summary, it is the opinion of the authors that the use of knowledge based systems does not make the issue of standards any more demanding than does the development of software in general. With regard to mitigating the issues of standards we see no current conclusive proof based on our observations and involvement in software development to indicate that the use of knowledge based systems in coalition operations does or does not make the requirement for standards and commonality any less. In fact the determining fact is more driven by other functional factors than the technology methods employed in development. The degree of standard requirements seems directly related to the degree of interoperability and integration desired. The impact is also determined on whether or not management attention is given to standards and the defining of the desired role in the initiative. In other words, standards can have as big as an impact as you desire. However, our recommendation is to adhere to a "minimum essential" policy with respect to standards placed on software systems. We have further observed that it is best to address the area of standards at the beginning of a program and not to ignore the issue or attempt to retrofit later.

References

Burstein, M. H., Mulvehill, A. M., and Deutsch, S. (1998) "An Approach to Mixed-Initiative Management of Heterogeneous Software Agent Teams", in Proceedings of the 32nd Hawaii International Conference on System Sciences.

Dean, Michael (1998) "JTF ATD Pushes OO Concepts & Techniques Including OMG's CORBA" 10th Software Technology Conference in Salt Lake, Utah.

Defense Information Systems Agency (1994) “Joint Warrior Interoperability Demonstration 1994, Demonstration Report”, Joint Interoperability Test Center, Fort Huachuca, Arizona.

Defense Information Systems Agency (1995) “Joint Warrior Interoperability Demonstration, 1995, Demonstration Report”, Joint Interoperability Test Command, Fort Huachuca, Arizona.

Hayes-Roth, F. (1995). The JTF ATD architecture for agile, mobile, collaborative crisis response: A case study in architecture-driven development. Proceedings of the First International Workshop on Software Architecture. Carnegie-Mellon Univ., Pittsburgh, PA.

Hofstede, Geert, (1997) “Cultures and Organizations: Software of the Mind”, McGraw-Hill, New York.

Marcus, A. and Gould, E.W. (2000) “Crosscurrents, Cultural Dimensions and Global Web User-Interface Design”, pp. 31 – 46, ACM Interactions, Vol VII.4.

Schill, John. (1995) Joint Task Force Advanced Technology Demonstration (JTF ATD). Proceedings of the 1995 Conference on Supercomputing, San Diego CA.

Tate, Austin (Ed) (1996) “Advanced Planning Technology, Technological Achievements of the ARPA/Rome Laboratory Planning Initiative”, AAAI Press.

Knowledge Based Approach to OOTW Coalition Formation

Michal Pěchouček, Vladimír Mařík, Jaroslav Bárta

Gerstner Laboratory, Department of Cybernetics,
Czech Technical University,
Technická 2, 166 27 Prague, Czech Republic
email: {pechouc|marik|barta} @labe.felk.cvut.cz

Abstract. The task of planning humanitarian relief operations within a high number of hardly collaborating and vaguely linked non-governmental organizations is a challenging problem. We suggest an alternative knowledge based approach to the coalition formation problem for humanitarian and peace-keeping missions. Owing to the very special nature of this domain, where the agents representing individual organisations may eventually agree to collaborate, but are very often reluctant to share their knowledge and resources, we tried to reduce the problem complexity by splitting the community of agents into alliances. We combined classical negotiation mechanisms with the acquaintance models and social knowledge techniques in order to reduce the communication traffic and to keep the privacy of knowledge. Experimental results are discussed in the paper.

1 Introduction

The application domain of this coalition formation research belongs to the area of **war avoidance operations** such as peace-keeping, peace-enforcing, non-combatant evacuation or disaster relief operations. Unlike in classical war operations, where the technology of decision making is strictly hierarchical, **operations other than war** (OOTW) are very likely to be based on cooperation of a number of different, quasi-volunteered, vaguely organized groups of people, non-governmental organizations (NGO's), institutions providing humanitarian aid, but also army troops and official governmental initiatives.

Collaborative, unlike hierarchical, approach to operation planning allows greater deal of flexibility and dynamics in grouping optimal parties playing an active role in the operation. New entities shall be free to join autonomously and involve themselves in planning with respect to their capabilities. Therefore any organization framework must be essentially "open". OOTW have, according to (Walker, 1999), multiple perspective on plan evaluation as there does not need to be one shared goal or a single metrics of the operation (such as political, economical, humanitarian). From the same reason, the goals of entities involved in a possible coalition may be in conflict. Even if the community members share the same goal, it can be easily misunderstood due to different cultural backgrounds.

The main reason why we can hardly plan operations involving different NGO's by a central authority results from their **reluctance to provide information** about their intentions, goals and resources. Consequently, besides difficulties related to planning and negotiation we have to face the problems how to assure sharing the detailed information. Many institutions will be ready to share resources and information within some well specified community, whereas they will refuse to register their full capabilities and plans with a central planning system and to follow centralized commands. They may agree to participate in executing a plan, in forming of which they played an active role. In our interpretation, an agent is a complex, organized entity (representing a NGO, humanitarian organization, army troop, etc.) playing an active role in the OOTW planning. A multi-agent system consists of a number of agents that group themselves in various, temporary coalitions (each solving a specific mission/part of the mission).

The main ambition of our research has been to analyze the problem of OOTW coalition formation and to propose a novel approach that would (i) make the coalition formation process simpler in comparison to the "classical" methods, and thus more efficient and (ii) at the same time maintain confidentiality of the private information. In our case, we decided to sacrifice the total optimality of the formed coalitions as we found this is not the most important aspect in the OOTW planning. We have suggested a concept of alliances – a set of agents that agreed to share some of their private information and to cooperate eventually. The coalition formation complexity is reduced by splitting the whole community of agents into disjunctive subsets (alliances) and by the attempts to create a coalition preferably within the single alliance. Social knowledge stored in the acquaintance models of individual agents has been widely explored in order to:

- **minimize required communication traffic** which influences the problem solving efficiency,
- **keep the quality of the coalition** that resulted from the coalition formation process operation 'reasonably good' – the quality has been measured by the humanitarian relief aid deliver time and by how much the coalition covers the request (in percent),
- **minimize the loss of agents' semiprivate information** when negotiating the mission – i.e. revealing the information about services the agent may provide, its status and intention in the minimum extent, and
- **minimize the amount of shared information** – information that possible coalition leaders know about other agents and use it in order to plan an optimal mission.

The developed approach has been tested on the CPlanT multi-agent system implementation.

2 CPlanT System Architecture

CPlanT is a multi-agent system for planning humanitarian relief operations where any agent can initiate the planning process. Classical negotiation algorithms such as contract net protocol (CNP) are used in combination with the acquaintance models techniques (Smith, 1980; Mařík, 2001). The CPlanT architecture consists of several specific classes of agents:

Resource Agents (R-agents) represent the in-place resources that are inevitable for delivering humanitarian aid, such as roads, airports. Unlike the below-defined H-agents, the R-agents are regarded as passive and they do not initiate any kind of humanitarian effort.

In-need Agents (In-agents) represent the centers of conflict that call for help (e.g. cities, villages, etc.).

Humanitarian Agents (H-agents) represent the participating humanitarian agencies. Like the R-agents, the H-agents contribute to humanitarian aid missions. Therefore, one may regard the H-agent as a subclass of R-agents. However the H-agents are proactive and they can initiate coalition formation process.

In this paper, we will investigate coalition formation among the H-agents.

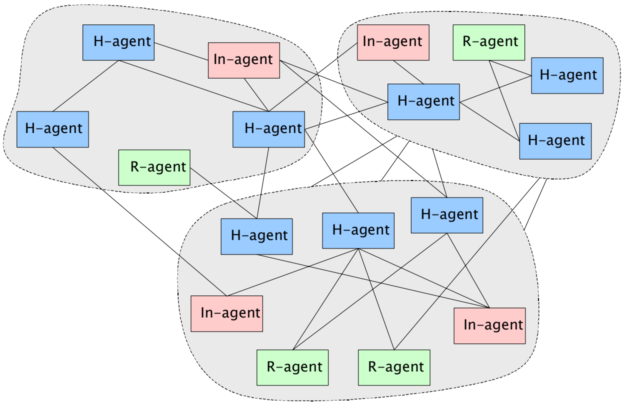


Figure 1 – CPlanT Multi-Agent Architecture

3 Knowledge Architecture

3.1 Agent's Neighborhood

Each H-agent may participate in one alliance of 'friendly' agents and at the same time it may be actively involved in several coalitions of agents cooperating in fulfilling specific shared tasks. Computational and communication complexity of forming such a coalition depends on the amount of pre-prepared information the agents administer one about the other and on sophistication of the agents' capability to reason about the other agents' resources, plans and intentions. The agents can allow others to reason about them and at the same time they can reason differently about the agents that belong to their different scopes of reasoning – neighborhood. Therefore, we distinguish among several types of agents' neighborhoods:

- $\alpha(A)$ – agent's **total neighborhood**, a set of all agents that the agent A is aware of, (e.g. knows about their existence and is able to communicate with them)
- $\mu(A)$ - agent's **social (monitoring) neighborhood** that is a set of agents, which the agent A keeps specific information about (e.g. services they provide, status, load, etc.). This neighborhood consists of the set of the agents that the agent A reasons about – $\mu^+(A)$ and the set the agents that reason about the agent A – $\mu^-(A)$. Therefore
$$\forall\ B \in \mu^-(A): A \in \mu^+(B).$$
- $\varepsilon(A)$ – agent's **cooperation neighborhood** that is a set of agents jointly collaborating (or committed to collaboration) in achieving one or more shared goals.

3.2 Knowledge Sharing

In order to reason one about the other, the agents must share some of their knowledge. Let us introduce the operator $(\mathbf{Bel}\ A\ \varphi)$ that expresses the agent's A awareness of the formula φ being true (Wooldirdge 2000). We say that the agent A_0 intentionally shares its knowledge $\mathbf{K}(A_0)$ with a set of agents $\delta(A_0) \subseteq \Theta$ provided that:

$$\mathbf{K}(A_0) = \{\varphi\} : \forall\varphi \in \mathbf{K}(A_0) : \forall A_i \in \delta(A_0) : (\mathbf{Bel}\ A_i\ \varphi) \wedge \forall B_i \notin \{\delta(A_0) \cup \{A_0\}\} : (\mathbf{Bel}\ A_0 \neg(\mathbf{Bel}\ B_i\ \varphi)).$$

From the previous follows, that if an agent B knows some of the shared information without the agent A_0 being aware of this fact, the agent B is not regarded as a member of the $\delta(A_0)$ set of agents, representing A_0's knowledge sharing neighborhood. According to this classification, we suggest three levels of the H-agent's knowledge sharing:

Public Knowledge is shared within the entire multi-agent community. If it is assumed that all the agents know one about the other (i.e. $\forall A, A \in \Theta : \alpha(A) = \Theta$), public knowledge $\mathbf{K_P}(A_0)$ of an agent A_0 is defined as

$$\mathbf{K_P}(A_0) = \mathbf{K}(A_0) \text{ where } \delta(A_0) = \alpha(A_0).$$

This class of knowledge is freely accessible within the community. As public knowledge we understand the agent's name, the type of the organization the agent represents, the general objectives of the agent's activity, the country where the agent is registered, agent's human-human contact (telephone, fax number, email), the human-agent type of contact (http address), the agent-agent type of contact (the IP address, incoming port, ACL) and, finally, available services.

Semi-Private Knowledge is shared within agents' social neighborhoods. Semi-private knowledge $\mathbf{K_s}(A_0)$ of an agent A_0 is defined as

$$\mathbf{K_s}(A_0) = \mathbf{K}(A_0) \text{ where } \delta(A_0) = \mu(A_0).$$

As in the OOTW domain, we do not assume the knowledge to be shared within the overlapping alliances, we will require the social neighborhood to have the following property: $\forall\ A \in \Theta : \mu^{-}(A) = \mu^{+}(A) = \mu(A)$. Members of a social neighborhood share information about availability of their resources.

Private Knowledge is owned and administered by the agent itself. Private knowledge $\mathbf{K_P}(A_0)$ of an agent A_0 is defined as

$$\mathbf{K_{pr}}(A_0) = \mathbf{K}(A_0) \text{ where } \delta(A_0) = \{\},$$

An important type of private knowledge includes agent's collaboration preferences, alliance restrictions, coalition leader restrictions and possible next restrictions, but also agent's planning and scheduling algorithms.

3.3 Alliance, Coalition, Team Action Plan

In the subject domain, we will understand as the multi-agent community Θ the whole collection of agents participating in the above-described OOTW (quasi-volunteered, vaguely organized groups of people, non-governmental organizations, institutions providing humanitarian aid, army troops or official governmental initiatives). We will introduce the concept of an **alliance** as a collection of agents that share information about their resources and all agree to form possible coalitions. The alliance is regarded as a long-term cooperation agreement among the agents. Members of an alliance will all belong to one others' social neighborhood. Provided that we assume that each agent belongs also to its own social neighborhood – $\forall\ A \in \Theta$: $A \in \mu(A)$, we define the alliance as follows:

An **alliance** is a set of agents κ, so that $\forall\ A \in \Theta : \exists\kappa : A \in \kappa \wedge \forall\ A_i \in \kappa : \kappa = \mu(A_i)$.

A singleton agent is regarded as an alliance with just one member. From the requirements for the reciprocal knowledge sharing within an alliance follows that

$$\forall\ A \in \kappa : \kappa = \mu(A).$$

Therefore, an important property of an alliance is that it cannot overlap with another alliance:

$$\forall\ \kappa_1, \kappa_2 \subseteq \Theta\text{: } (\exists A\text{: } A \in \kappa_1 \wedge A \in \kappa_2) \Rightarrow \kappa_1 \equiv \kappa_2.$$

Let us define a **coalition** as a set of agents, which agreed to fulfill a single, well-specified goal. Coalition members committed themselves to collaborate on the within-coalition-shared goal. Under the assumption $\forall A \in \Theta$: $A \in \varepsilon(A)$ we define coalition as follows:

A **coalition** is a set of agents χ, so that $\forall\chi(\tau) \subseteq \Theta$: $\forall\ \mathrm{A} \in \chi(\tau) : \chi(\tau) \subseteq \varepsilon(\mathrm{A})$.

Let us introduce a set $\varepsilon(A,\tau)$ that is an agent collaboration neighborhood with respect to a single shared goal τ. Then

$$\varepsilon(A) = \bigcup_{\tau} \varepsilon(A,\tau), \text{ and } \forall\chi(\tau) \subseteq \Theta\text{: } \forall\ A \in \chi(\tau) : \chi(\tau) = \varepsilon(A,\tau).$$

A coalition, unlike an alliance, is usually regarded as a short-term agreement between collaborative agents. As we will see in Section 6, it is better for a coalition to be a subset of one alliance, but it is not an inevitable condition. A coalition can consist of agents who are members of different alliances.

Another term that we have to introduce is a **team action plan**. In planning humanitarian relief operations, similarly as in the case of any other collaborative action planning, the agents must agree on how they will achieve the goal τ. The team action plan is thus a decomposition of a goal τ into a set of tasks $\{\tau_i\}$. The tasks will be delegated within the coalition members. Apart from the responsible agent, each task shall be denoted by its due time, start time and price. Provided that an agent A_j is responsible for implementing the task τ_i (where $\tau = \{\tau_i\}$) in time $\mathsf{due}(\tau_i)$, starting at $\mathsf{start}(\tau_i)$ for the price $\mathsf{price}(\tau_i)$, we define the team action plan as follows:

A **team action plan** $\pi(\tau)$ is as a set $\pi(\tau) = \{\langle\tau_i, A_j, \mathsf{start}(\tau_i), \mathsf{due}(\tau_i), \mathsf{price}(\tau_i)\rangle\}$.

We say that the team action plan $\pi(\tau)$ is **correct** if all the collaborators A_j are able to implement the task τ_i in the given time and for the given price. The team action plan $\pi(\tau)$ is **accepted** if all agents A_j get committed to implementing the task τ_i in the given time and for the given price. Similarly, we say about the goal τ to be **achievable,** if there exists such $\pi(\tau)$ that is correct. The goal τ is said to be **planned,** if there exists $\pi(\tau)$ that is accepted. Obviously, there is an important relation between the team action plan and the coalition. We say that a coalition $\chi(\tau)$ achieves a goal τ by implementing a team action plan $\pi(\tau)$ if and only if $\chi(\tau)= \{A_j\}$ and $\pi(\tau)$ is correct.

3.4 Disclosure of Private and Semi-Private Knowledge

Measuring the loss of information, that the agents may want to keep private, is an uneasy task. The revealed piece of information has got different value to agents with different meta-reasoning capabilities (Pěchouček & Norrie, 2000). In order to vaguely categorize various types of information leaks, let us distinguish between strong and weak leaks.

- **Strong information disclosure**: If an agent looses some type of private (or semi-private) knowledge in the strong sense, it does so as a side effect of some proactive step (such as sending a request).
- **Weak information disclosure**: If an agent looses the private knowledge in the weak sense, it deliberately discloses some piece of its knowledge to other agents (e.g. when sending an `inform`-type message).

Each agent undertakes the weak loss of some of its knowledge when forming an alliance. At this moment the agent's semi-private knowledge gets disclosed within the alliance members. In our system, the agent looses some of its private knowledge in the strong sense, if it communicates with an agent which is outside of its alliance. Once the agent A_1 from an alliance κ_1 sends a request for a service τ to the agent A_2 from the alliance κ_2, the agent A_1 reveals the information about the **intent** (e.g. A_1 does something that requires τ) and information about agent's A_1 capabilities (e.g. A_1 cannot do τ). At the same time, a proposal for collaboration from the agent A_2 reveals the information about agent's A_2 capabilities (such as A_2 can implement τ in time t_1). However, this type of knowledge disclosure has been reduced as the agent A_2 acts on behalf of the entire alliance. Therefore, if A_2 offers some services that are not used at the end, there is a loss of information about capabilities of the entire alliance (and not of the agent A_2 itself).

4 Agents' Acquaintance Model

Let us very briefly introduce the concept of agent's social intelligence and acquaintance models. Apart from its **problem-solving knowledge** that guides agent's autonomous local decision making processes (such as coalition formation, or team action planning), the agents usually exploit **social knowledge** that expresses the other agent's behavioral patterns, their capabilities, load, experiences, resources, commitments, knowledge describing conversations or negotiation scenarios (Mařík et. al., 2001). This knowledge is usually stored separately from the agents' computational core – in an agent's **acquaintance model**. There have been investigated several acquaintance models previously. Based on the *tri-base acquaintance model* (Pěchouček et. al., 2001), the social knowledge in CPlanT is organized in four separate knowledge structures:

- **community-base** (Com-BB) – which is a collection of the community members' public knowledge

  $$\text{Com-BB}(A_0)=\{\mathrm{K_p}(A_\mathrm{i})\} \text{ for } \forall A_\mathrm{i} \in \alpha(A_0)$$

- **self-belief-base** (Self-BB) – where the agent's reflective knowledge about itself is located; here the agent stores its public knowledge that is accessible to anyone, its semi-private knowledge that is shared within the alliance and its private knowledge that is not shared by anyone,

  $$\text{Self-BB}(A_0)= \{\{\mathrm{K_p}(A_0)\}, \{\mathrm{K_S}(A_0)\}, \{\mathrm{K_{Pr}}(A_0)\}\}$$

- **social-belief-base** (Soc-BB) – where the agent stores the semi-private knowledge of its peer alliance members,

  $$\text{Soc-BB}(A_0)=\{\mathrm{K_S}(A_\mathrm{i})\} \text{ for } \forall A_\mathrm{i}\in \mu(A_0)$$

- **coalition-base** (Coal-BB) – which is a dynamic collection of the peer coalition members, the past and possible future coalitions as much as permanent coalition-formation rules[1].

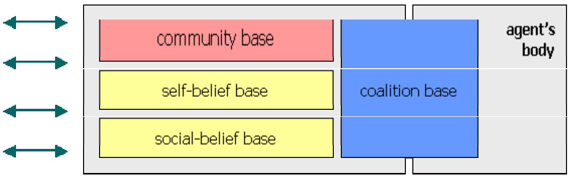


Figure 2 – Structure of the CPlanT Acquaintance Model

Exploitation of the acquaintance model reduces communication traffic required for collaborative activity planning. In principle, the social knowledge substantially reduces the set of agents (ideally to one) that will be requested by the coordinating agent in the CNP process (Smith, 1980). An important flaw of this approach is rooted in high requirements for the social model maintenance. The social knowledge maintenance may be driven either by the owner of the acquaintance model (the coordinator) or by those which are represented in the model – hence service providers (collaborators). We refer to the former strategy as the **requestor-driven** knowledge maintenance and to the latter strategy as the **provider-driven** knowledge maintenance. As an example of a **requestor-driven** strategy

[1] The coalition-formation rules are instances of the agent's problem-solving knowledge, while the information about the coalition members, past and future coalitions are instances of the social knowledge. Therefore the coalition base belongs in part to the acquaintance model and to the agent's body

there is the concept of **periodical revisions** (Mařík at.al., 2000) where the knowledge owner periodically checks consistency of the model with the potential collaborators. In other systems, there has been a **cooperation trader** (Cao at. al., 1997) type of agent, which was in charge of maintaining the agents social knowledge. As explained in Section 164 we have adopted the **provider-driven** knowledge maintenance in CPlanT.

Self-Belief Base			
public knowledge:		Semi-private knowledge:	Private knowledge
Port: 1500 **ip_address**: "147.32.86.167" **Country**: suffer terra **City**: north port **Type:** Religious **Ontologies:** fipa-am, cplant-ontology		Food: 3000 Nurses: 50 Trucks: 13	Alliance restrictions: ("country","Suffer Terra") Leader restrictions: ("type","Military"). City restrictions: ("muslim",50) Cooperates with: ("type","government")
Social belief base			
	Agent: ST Police	Armed-people:30	
	Agent: Christian STHO	Food: 3500 Clothing: 280 Nurses: 22 Medical-people: 12	
Community belief base			
	Agent: Suffer Terra Government	Suffer Terra Government@iiop://147.32.84.131:2188/Suffer Terra Government Type: Government Services: Food, Civil-material, Medical-material, Clothing Ontologies: FIPA-AGENT-MANAGEMENT, MAP-ONTOLOGY, PORT-ONTOLOGY, CPLANT, ALLIANCE Languages: SL1, KIF, State: ACTIVE Country: Suffer Terra, City: Suffer Town	
	Agent: Christian STHO	Christian Suffer Terra Humanitarian Organization@iiop://147.32.84.131:2210/Chr ST Humanitarian Organization Type: Religious Services: Food, Clothing, Medical-people, Nurses, Medical-material Ontologies: FIPA-AGENT-MANAGEMENT, MAP-ONTOLOGY, PORT-ONTOLOGY, CPLANT, ALLIANCE Languages: SL1, KIF, State: ACTIVE Country: Suffer Terra, City: North Port	
Coalition Base			
Rules	(VOLCANIC-AVERAGE-SMALL-TOWN → Time: 220 (Requirements: Medical-material 60, Food 1500, Civil-material 30000, Medical-people 16, Civil-people 27, Nurses 19) …		
Coalitions	(coalition (**Members**: Suffer Terra Government, Suffer Terra Police, Christian Suffer Terra Humanitarian Organization) (**Services**: Food, Civil-material, Medical-material, Clothing, Military-people, Food, Clothing, Medical-people, Nurses) (**Price-for-coordination**: 5)) (planned-coalition (Task name: Suffer-Town-24-1-2002/17-49-53.1 (Coalition members: Suffer Terra Government, Suffer Terra Police, Christian Suffer Terra Humanitarian Organization) (Coalition leader: Christian Suffer Terra Humanitarian Organization (Disaster: Volcanic, Degree: 1, (Allocations: Civil-material, 80000, Allocation Time: 350 Food, 80000, Allocation Time: 350 Medical-material, 80000, Allocation Time: 350)) …		

Table 1 – Instance of an H-agent's acquaintance model

5 Inter-Agent Communication

Before explaining the lifecycle of the system let us comment the main communication techniques that have been used in CPlanT: central communication agent, contract net protocol, and acquaintance models. We have tried to minimize the role of the central communication component, as it is an important communication bottleneck of the system operation and a center where the agents' private knowledge may be sniffed (see Section 6).

5.1 Contract Net Protocol

The CPlanT implementation relied heavily on the **contract net protocol** (CNP) negotiation scenario (Smith, 1980). Any agent can initiate the coalition forming process (hereafter we refer to this agent as a coalition **coordinator**) by requesting some agents in the community (**collaborators**) for specific services. Upon receiving proposals for collaboration, the coordinator carries out a computational process by which it selects the best possible collaborator(s) – see Figure 4. The coalition planning process can also be multi-staged. Such an approach requires substantial computational resources and fails in complex communities. For each single-staged CNP within a community of *n* agents, it is needed to send 2(*n*+1) messages in the worst case.

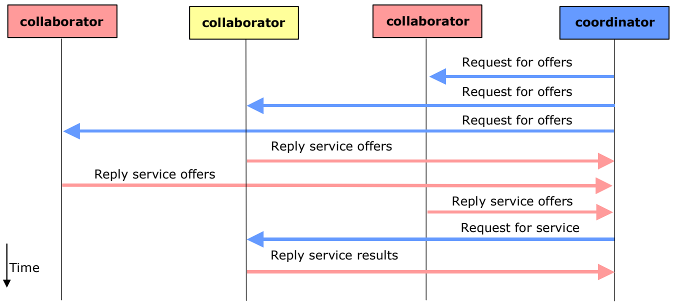


Figure 3 – Contraction based on a Single-staged Contract Net Protocol

At the same time many agents may not want to enter the CNP negotiation, as they wouldn't wish to undertake the risk of disclosing their private knowledge.

5.2 Acquaintance Model Contraction

The alternative communication strategy to CNP is based on exploitation of the agents' social knowledge. A coalition coordinator subscribes (by sending a `subscribe`-type of message) the potential collaborators for specific services they may want to exploit in the future. Upon a change in the collaborators' capabilities, they provide the coordinator with an update in the form of an `inform`-type of message. When the coordinator triggers the coalition formation phase, it parses the prepared service offers and selects the best collaborator(s) without any further negotiation. The coordinator sends a request, the collaborator updates its resources and confirms the contract. Any change in collaborator resources is advertised to all coordinators which subscribed the collaborator (see Figure 4).

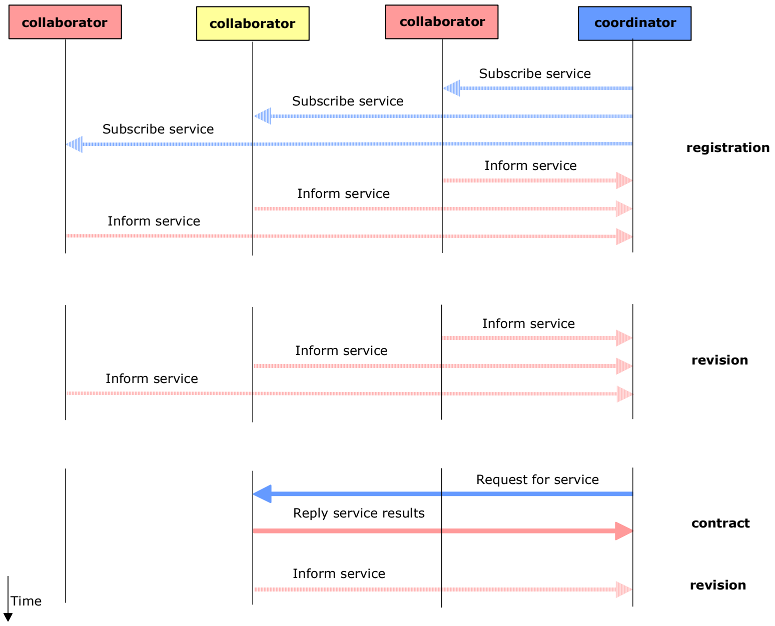


Figure 4 – Contraction based on Acquaintance Model exploitation

If there is a single event in the community Θ that affects all the agents ($n = |\Theta|$) and all the agents are mutually subscribed, then in the worst case there is ($n(n-1)$) messages required for the social knowledge maintenance on this event. However, this is rarely the case. Agents never subscribe all each other (we could easily use a central communication component instead).

6 CPlanT Operation Lifecycle

The CPlanT multi-agent system operates in four separate phases: **registration** for agents' login/logout to/from the community, **alliance formation** for forming of alliances, **coalition formation** for finding a group of agents which can fulfill a well specified task and **team action planning** for resource allocation within the specific coalition. In the following, we will comment each of the phases.

6.1 Registration

Throughout the registration phase, a new-coming agent registers within the multi-agent community. This agent registers its public knowledge with the special central registration agent – the **facilitator**. Subsequently, the facilitator informs all the already existing agents about the new agent, and it also informs the new agent about all existing agents. Similarly, the agents can deregister with the facilitator. Any registered agent stores the public knowledge about all members of its total neighborhood $\alpha(A)$ in the `Com-BB`(A) base of its acquaintance model. We have used the central communication unit – facilitator in the registration phase only. As the agents register just their public knowledge, we do not breach the requirements for confidentiality of the private information.

6.2 Alliance Formation

In this phase, which follows the registration process, the agents analyze the information they have about the members of the multi-agent system and attempt to form alliances. In principle, each agent is expected to compare its own private knowledge (i.e. alliance formation restrictions) with the public knowledge about the possible alliance members (i.e. type of an organization, its objectives, country of origin, etc.). Had the agent detected a possible future collaborator, the agent would propose joining the alliance. Throughout the negotiation process, the agent either chooses the best alliance according its collaboration preferences of agents into already existing alliances. Failing to do so, an agent may start a new alliance by itself.

According to their preferences in Self-BB and community public knowledge in Com-BB, the agents carry out a selective contract net protocol process during this phase. The **quality of an alliance** is understood in terms of maximizing the individual agent's contribution to the alliance (i.e. covering the biggest amount of services that the other members of the alliance cannot implement). It is important to note that this process does not give us any guarantee for optimality of the alliance allocation. Each agent will join the most profitable alliance with respect to existing alliance configuration. With changing the order of agents' registration with the alliance, the formation algorithm will come up with different alliances.

6.3 Coalition Formation

In this phase, the agents group together not according to a similar mission objective, but they form coalitions with respect to a single, well-specified task that needs to be accomplished. Both, the CNP technique and the acquaintance model have been used in the coalition formation process. First, let us talk about the coalition formation process within a single alliance. The alliance members know the most of each other and are able to suggest a coalition that will very likely have foreseen properties. Whichever agent, member of an alliance, can face the role of the coordinator of the goal τ implementation. The coordinator, who is to be set randomly in our implementation, parses its social neighborhood $\mu(A)$ and detects the set of the most suitable collaborators (cooperation neighborhood) – $\varepsilon(A, \tau)$. Upon an approval from each of the suggested agents, the respective coalition $\chi(\tau) = \varepsilon(A, \tau)$ is to be formed. Maintaining the agents' social neighborhood will save an important part of agent's interaction in the time of coalition formation. Agents will not need to broadcast a call for collaboration each time they will be required to accomplish a task. Instead, they will consult this pre-prepared knowledge and will contract the agent of which they knew it is the best to work with. The coordinator optimizes the **quality of a coalition** by seeking the coalitions that would contribute the most and in the shortest possible time.

As said in the previous, the agents' prefer not to form coalitions across alliances ($\forall\ \tau$: $\varepsilon(A, \tau) \subseteq \mu(A)$). However sometimes an alliance fails to achieve a goal. The coordinator, who failed to form a coalition within one alliance, negotiates the proposal for collaboration by classical CNP with the agents from its total neighborhood $\alpha(A_0)$.

6.4 Team Action Planning

Once a coalition is formed, the agents share a joint commitment to achieve the goal τ. Within this phase, a team of collaborative agents jointly creates a team action plan $\pi(\tau)$. The team action plan, that is a result of the coalition planning activity, is a joint commitment structure that defines exactly how each team member will contribute to achieving the shared goal (amount of resources, deadlines, etc.). The coordinator is supposed to (i) decompose a goal τ into subtasks $\{\tau_i\}$ and (ii) allocate the subtasks within the already formed coalition $\chi(\tau)$. There may be many achievable team action plans $\pi(\tau)$. The coordinator seeks for the cheapest or the fastest possible plan.

As there is no semi-private knowledge shared across the alliances, the agents from different alliances coordinate their activities by means of the contract net protocol. The intra-alliance team-action planning mechanism is not the pure acquaintance model contraction, where the team-action plan would result from the coalition leader deliberation process followed by a contract. All coalition members construct the precise team action plan collaboratively.

The collaborators advertise their services in the most informative while efficient form. We have suggested linear approximation of the discrete function that maps the delivery amount into due dates. Therefore the coordinator's acquaintance model stores the social knowledge that is imprecise, but very compact and efficient to parse. According to this social knowledge, the coordinator suggests the most optimal request decomposition and resource allocation – $\pi(\tau)$ and transforms it into a contract proposal. This proposal is sent to the other coalition members, which reply with a specific collaboration proposal. However, the coordinator may find this proposal to be different than expected owing to the fact that the approximate information provided by the collaborator was far to imprecise. Instead of agreeing upon a joint commitment for this sub-optimal team action plan, the coordinator adapts the conflicting social knowledge and fires another round of negotiation. With each further negotiation stage the team action plan should be closer to the optimal team action plan. This process is to be iterated until there is no conflict in the expected capacity of the collaborators and the proposed delivery.

7 Implementation and Testing

7.1 Implementation

Testing correctness of the CPlanT required a well-defined, formal, but realistic enough scenario that can represent, model and initiate all aspects of agents' nontrivial behavior. The above specified principles and ideas have been tested and implemented on a subset of the OOTW types of operations – humanitarian relief operations. For this purpose we designed and implemented a hypothetical humanitarian scenario Sufferterra representing a suffering island and several imaginary countries ready to help. The Sufferterra scenario was inspired by (Walker, 1999; Rathmell, 1999, Reece &Tate, 1998). The scenario knowledge has been encoded in XML and the computational model of the scenario has been implemented in Allegro Common Lisp.

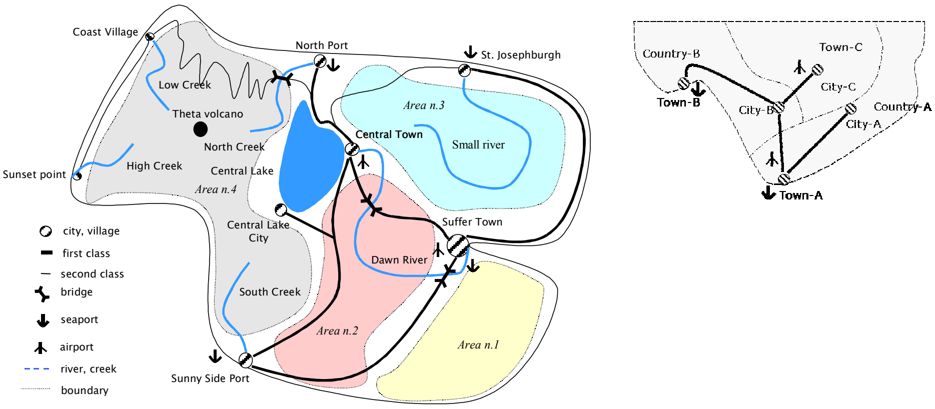


Figure 5 – Sufferterra – subject of humanitarian operations

The R-Agents specify the physical arrangements of the geographical objects and the resources they provide. The problem specification does not distinguish the level of modeling granularity, i.e. each physical object may be implemented as an R-agent or several physical objects can make together an R-agent. For the testing purposes we have implemented a single R-Agent that represents the entire map of the area. The H-agents subscribe the R-Agent for specific information, by which these subscribers are informed about any change in physical arrangements of the relevant part of the map. There is a simple IN-Agent implemented as a part of the CPlanT community. Through one of the running instances of the IN-Agent, one can compose a “call-for-help” request and execute the coalition planning process. Such a request includes the type of disaster (“volcanic”, “hurricane”, “flood”, “earthquake”), the degree of disaster (1..9), location and the targeted H–Agent.

```
<city>
     <name> "Suffer Town" </name>
     <national-composition> "((christian 67) (muslim 18) (native 13) (other 2))"
     </national-composition>
     <population> "50000" </population>
     <seaport>
           <ID> "1" </ID>
           <capacity> "25" </capacity>
           <material-hour> "200000" </material-hour>
     </seaport>
     <airport>
           <ID> "1" </ID>
           <capacity> "30" </capacity>
           <material-hour> "100000" </material-hour>
           <runway> "3000" </runway>
     </airport>
</city>
```

Figure 6 – Example of XML definition encoding of the ‘Suffer Town’ object

CPlanT has been successfully tested on the Sufferterra humanitarian relief scenario. The implementation is complemented by a visualizing meta-agent, which is implemented in Java. This meta-agent views logical structure of the system e.g. alliances, coalitions, team action plans and other properties of the community. There is a separate visualization for communication traffic monitoring. This component, that is not an agent, but rather a part of the multi-agent platform, serves mainly to debugging purposes. The community can be viewed and the requests can be sent from the web server via classical Internet browsers and from the WAP phones interface as well.

7.2 Experiments, Testing

Several different objectives were followed within the frame of the experiments: to evaluate the communication and computation requirements, quality of the solution provided and disclosure of private and semiprivate knowledge.

Communication traffic

As stated in Section 5, an important part of the agent deliberation process has been decomposed into the inter-agent negotiation process. This is why we have concentrated our attention primarily to savings of the communication traffic in the entire system. The communication traffic has been observed in different architecture arrangements of the community (e.g. different number of alliances) and for different complexity of the tasks sent to the community (e.g. different number of contracts). Having 20 agents we have experimented with the sample of all agents in one alliance, with agents clustered into 2, 4, 7 and 20 alliances. All the experiments have been carried out on the set of 19 measurements for each community arrangement. From the definition of the community lifecycle (see Section 6) follows that the latter case (∀ A: $\mu(A)=\varnothing$) does not exploit any advantages of the acquaintance model contraction and the community behaves such as no social knowledge is administered and used. An important part of the communication traffic is carried out in the critical time – i.e. in the moment when the system is requested to provide a plan. By exploiting social knowledge that has been prepared in advance, we aimed at minimising communication traffic in this moment. The cost we have paid for this was the increased communication traffic in the idle times of the community. In the idle times, the agents are busy with maintaining the social knowledge stored in their acquaintance models. The communication traffic grows with the increasing number of alliances as each alliance member stores a more voluminous acquaintance model and it searches for a coalition by parsing the acquaintance model only.

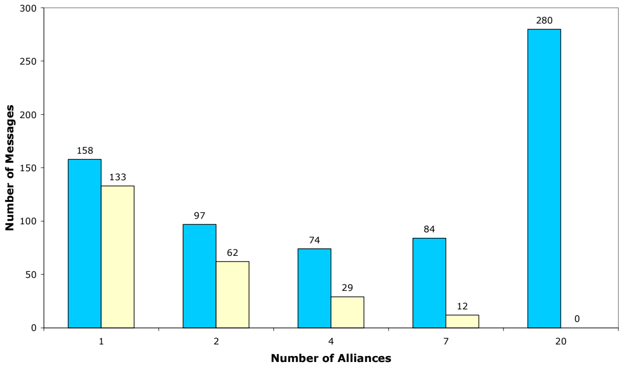


Figure 7 – Communication traffic in communities with different number of alliances. The light bar depicts the maintenance messages, while the dark bar illustrates the overall communication in the system.

From the graph in Figure 7 we can see that with an increasing number of alliances (and a decreasing average number of alliance members) we reduce the communication requirements for maintenance of the model. The most of the communication in the critical time (the difference between dark and light bars in the graph) we save in the case of just one huge alliance. The optimal arrangement of the community was identified in the case of four alliances. However, it is not possible to define an optimal system structure because the agents cannot predict future tasks and the number of agents required for implementing these tasks. It is clear that for tasks requiring low number of agents, we will prefer small alliances while for the task requiring many agents, larger alliances will be preferred. The, the optimal size of a coalition is given by the nature of the tasks/goals under consideration.

Evaluation of quality of the coalition

The evaluation of quality of the formed coalition is an important aspect in any coalition formation research. In Sufferterra scenario, there are two key attributes that influences the coalition value: (i) **success rate** – how many of the requested resources the coalition provides and (ii) **delivery time** – by when the coalition delivered the resources to the requestor. Experiments did not give any evidences to conclude any dependency between the success rate of the coalition and the used communication mechanism. However, with an increasing number of alliances, the overall delivery time is kept increasing due to additional costs of coordination among the alliances.

Knowledge disclosure

The key challenge has been minimization of the private and semi-private knowledge disclosure. We have tried to measure both types of the information disclosures. Once the **private** information has been identified by another agent, this agent finds about the intent of the respective agent. As already noted, this very often happens when an alliance fails to plan all the requests and starts a contract net protocol process within members of the other alliances. Those, who will not be awarded the contract, know that the coordinator intends to operate in a mission and that it needs the resources requested.

The **semiprivate** information is disclosed in the same situation, when the possible collaborator proposes a service (as a reaction to a coordinator call for collaboration) that will not be accepted by the coordinator. In such a case, the coordinator finds out about the services the suggested collaborator can provide. Both the above mentioned cases are classified as strong knowledge disclosures (see Section 3.4). The weak knowledge disclosure happens in the registration phase within a single alliance and represents the amount of information that has become shared within the alliance.

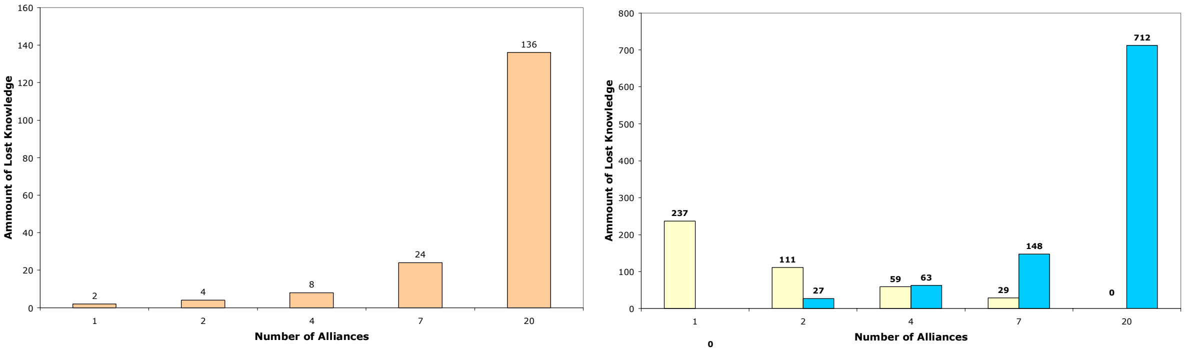


Figure 8 : The graph on the left-hand side shows the dependence of the amount of private information disclosure in different architectures of the community. The graph on the right-hand side illustrates disclosure of the semi-private knowledge. The light bar depicts the weak and the dark bar strong knowledge disclosure.

As expected, the biggest disclosure of intents appears in the case of 20 alliances, as there is the highest CNP-based communication traffic among the alliances (see Figure 8). There is no weak disclosure once the agents are utterly independent (20 alliances). On the other hand, there is no strong semiprivate information disclosure in one alliance while the independent agents are starting to loose their semi-private information in the strong sense. It makes no implication to put together the strong and weak knowledge disclosures because of their different nature.

An interesting fact is that neither of the two extreme cases is the best for concealing the agents' private and semi-private knowledge. With one alliance, the semi-private knowledge becomes public while with no alliance each contract net protocol will reveal information about the contractors' intentions. It is rather difficult to find a good compromise in a number of alliances. What matters, is the probability that a request will not be fulfilled within one alliance and the coalition leader will have to subcontract other agents. Amount and structures of alliances in our domain emerge naturally according to the agents' private knowledge and other collaboration restrictions. Therefore it makes no sense to suggest an optimal number of alliances for a given community.

8 Relation to Coalition Planning Research

There has been a lot of work carried out in the area of coalition formation and coalition planning. It has been shown that finding the optimal coalition is an NP complete problem (Sandholm & Lesser, 1997). Researchers mainly suggest different negotiation strategies and analyze complexities of the coalition formation process (Shehory & Kraus, 1995). When a subject of optimization is the quality of the formed coalition, the agents usually act **collaboratively**. There have been published many of centralized planning mechanisms for coalition formation (Sandholm et. al., 1999). On the other hand, the **self-interested** agents maximize their own profit when participating in a coalition, no matter how well they will perform as a group. Many researchers analyzed properties of communities of self-interested agents such as their stability, worst-case profit, or payoff division among the agents (Li & Sycara, 2001). The domain we have investigated is partially of cooperative and self-interested type at the same time. Humanitarian aid providing agents tend to cooperate in the time of a crisis while they are self-interested and compete each other in a long-term horizon. Therefore, there was suggested a concept of alliances – collectives of agents that agreed to collaborate (to potentially form a coalition).

More importantly, the profit is very often the key optimization criterion when the agents optimize a coalition formation process (either collaboratively or competing each other). Besides the quality of the coalition, in the OOTW domain there are two (maybe more important) aspects to be taken into account. As forming an optimal coalition is a very complex problem, the **response time** becomes an important issue. Agents are limited in resources and a reasonably good answer, that is quickly provided, is very often much better than an optimal coalition found later (Steinmetz et. al, 1998; Sandholm & Lesser 1997). Practitioners would add that implementing a multi-agent system with a large number of agents, that are supposed to interact heavily, results in a **communication traffic overload** (Kaminka et. al., 2001). In our research we have tried to decompose the reasoning process and distribute it among the agents. While keeping the agents' deliberation process simple, we have concentrated our efforts on minimizing the communication interaction among the agents in order to suggest community structuring in a reasonable time. As the OOTW agents are also self-interested in certain way, they want to stay hidden in front of someone and advertise its collaborative capabilities to others. This is why we have to respect also the amount of **private information** to be disclosed. Therefore, we have also studied leaks of private information while forming the coalitions.

Research of the teamwork in a similar domain (evacuation scenarios) was reported in (Tambe, 1997). It was suggested to integrate the already existing software applications in the TEAMCORE wrapper agents. Unlike our acquaintance model that contains just social knowledge, the TEAMCORE wrapper agents also maintain domain specific team plans and the hierarchy of goals. Teams of agents share a team-oriented program, which is a joint knowledge structure that coordinates their activities. In CPlanT, there is no explicit team action plan distributed in agents' acquaintance models. The structure of the coalitions and the team-action plan is a result of the inter-agent negotiation process. However, combination of both approaches where the agents' behavior is coordinated by a team-action plan that results from the agents' negotiation seems to be an interesting topic for further research.

Investigators approaching the problem from the game-theoretic point of view solve the problem of a higher complexity. Whereas in our case, there is a hierarchy structure for each task that is sent to the community and each task is coordinated by a single agent (the coordinator), in (Klusch et. al. 1997) all agents are equal. The agents autonomously analyze their own value. Through negotiations, they try to find out which coalition is the most profitable for them to join. This problem is inherently more complex and causes communication problems in complex communities. There will be several stages of negotiations needed as in many cases optimality of cooperation between two agents may not be reciprocal. In our case, we did not need to solve such a complex problem. On the other hand, in CPlanT we must optimize not only which coalition to join but also which services to provide to the coalition (e.g. team action planning). One may suggest that the game-theoretic approach could be used in the alliance formation phase of our algorithm (see Section 6.2). However, the agents join the system continuously, which makes it rather difficult to maintain the overall optimality of the distribution of alliances.

Besides the contract-net-protocol, there are other negotiation strategies based on classical auctioning mechanisms. While in combinatorial actions, the motivation of an agent is usually to make the biggest profit (or to contribute to a coalition in the best way), in our case, all the auctioneers and the bidding agents collaborate. A bidding agent tries to provide the auctioneer with such a bid that approximates in the best way the resources it can provide, and will help it to suggest the best possible resource allocation. In CPlanT, the agents also do not speculate about whom to work with. As we optimize the private information loss, collaboration within one alliance is always preferred. There is a potential of using the optimization for multiple auctioning mechanism for the team action planning within several overlapping coalitions (Anthony et. al. 2001).

9 Conclusions

The research described in this paper contributes to the coalition formation community by suggesting an alternative, knowledge based approach to the problem. Our research has been driven by the very specific domain of the OOTW. Apart from the classical contract net protocol techniques, we have used the communication strategy based on combination of three techniques: the centralized registration, the acquaintance models and the contract net protocol negotiations.

The agents in the community are organized into smaller, disjunctive groups called alliances. Each agent in the alliance is able to start the negotiation process to form a coalition and to develop a team action plan for a specific task either within the alliance or in collaboration with other alliances. Inside-alliance negotiations explore mainly the social knowledge stored in the acquaintance models, but the CNP technique is used as well (especially in the phase of the team action planning). The inter-alliance negotiations are based just on the CNP principles.

The general complexity of negotiations when forming a coalition in a MAS is of an exponentially explosive nature (Ketchpel 1993, Sandholm, 1995, Shehory, 1998). It has been shown that finding and optimal coalition is an NP complete problem when no specific constraints are imposed. In our case, the negotiation complexity of the coalition formation problem has been significantly reduced because:

– agents are organized into several disjunctive sets (alliances) and the most of coalitions are created just inside an alliance (reduced space of negotiations)
– the coalition leader within an alliance is set randomly (each coalition member has got the same coordination capacity and can manage the negotiation process), they don't compete for the role.
– within an alliance, the negotiation process explores the acquaintance models (social knowledge) in combination with the CNP technique and the pure CNP negotiations are used just in the case of the inter-alliance negotiations.

While the contract net protocol runs rather inefficiently, it keeps the agents from different alliances independent (they do not have to disclose their semi-private knowledge across alliances). This is why, the acquaintance-model based planning has been used exclusively within the alliances.

In our approach, we have not prioritized the requirement for the global coalition optimality, as this is not the main challenge in the OOTW planning. The main issue has been to develop an acceptable plan without forcing the agencies (agents) to make their private knowledge (namely intents and resources) public. This quite specific OOTW requirement enabled to reduce the complexity of the negotiation problem significantly. It has been measured that optimality of the coalition value slightly increases with the number of alliances (the role of the acquaintance model is getting smaller), while the problem complexity with a smaller number of socially knowledgeable alliances is significantly reduced.

Acknowledgement

This research, was supported by AFOSR/European Office of Aerospace Research and Development under contract number F61775-00-WE043 as well as by the Grant No. LN00B096 of the Ministry of Education, Youth and Sports of the Czech Republic.

References

Anthony P., Hall W., Dang V. and Jennings N. R. (2001) "Autonomous agents for participating in multiple on-line auctions" Proceedings of IJCAI Workshop on E-Business and the Intelligent Web, Seattle, WA, 54-64

Cao, W., Bian, C.-G., and Hartvigsen, G. (1997) Achieving Efficient Cooperation in a Multi-Agent System: The Twin-Base Modelling. In: Co-operative Information Agents (Kandzia, P., Klusch, M. eds.), LNAI No. 1202, Springer Verlag, Heidelberg, 210–221.

Kaminka, G. A., Pynadath, D. V., and Tambe, M. (2001) “Monitoring Deployed Agent Teams”, In Proceedings of the Fifth International Conference on Autonomous Agents (Agents-2001), Montreal, Canada.

Ketchpel, S. (1993) “Coalition Formation Among Autonomous Agents”, Proceedings of 5th European Workshop on Modeling Autonomous Agents in a Multi-Agent World, Neuchatel, Switzerland, August 25-27

Klusch, M., Contreras, J., Wu, F., Shehory, O. (1997) “Coalition Formation in a Power Transmission Planning Environment” Proc. 2. Int. Conf. on Practical Applications of Multi-Agent Systems PAAM-97, 21.-23.4, London

Li C. and Sycara K. (2001) “Algorithms for combinatorial coalition formation and payoff division in an electronic marketplace” Tech. report CMU-RI-TR-01-33, Carnegie Mellon University

Mařík V., Pěchouček M., Štěpánková O., Lažanský J. (2000) “ProPlanT: Multi-Agent System for Production Planning”, International Journal of Applied Artificial Intelligence. 2000, vol. 14, no. 7, p. 727-760

Mařík V., Pěchouček M., Štěpánková O. (2001) “Social Knowledge in Multi-Agent Systems”, In (M. Luck et. al, eds.) Multi-Agent Systems and Applications, LNAI 2086, Springer-Verlag, Heidelberg, 2001, pp.211-245

Pěchouček, M., Norrie, D. (2000) “Knowledge Structures for Reflective Multi-Agent Systems: On reasoning about other agents”, registered as Report Number 538, Department of Mechanical and Manufacturing Engineering, University of Calgary, Alberta, Canada,

Pěchouček M., Mařík V., Štěpánková O. (2001) “Towards Reducing Communication Traffic In Multi-Agent Systems”, Special Issue on Virtual Organizations and E-Commerce Applications, Journal of Applied System Studies, Cambridge International Science Publishing, ISSN 1466-7738, Cambridge, UK, Spring, 2001.

Rathmell R. A. (1999) “A Coalition Force Scenario ‘Binni – Gateway to the Golden Bowl of Africa’”, Defense Evaluation Research Agency

Reece, G. A. and Tate, A. (1993) “The Pacifica NEO Scenario”. Technical Report ARPA-RL/O-Plan2/TR/3. Artificial Intelligence Applications Institute, University of Edinburgh, Scotland

Sandholm, T. and Lesser, V. (1997) Coalitions among Computationally Bounded Agents. Artificial Intelligence 94(1), 99-137, Special issue on Economic Principles of Multi-Agent Systems.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., and Tohmé, F. (1999). Coalition Structure Generation with Worst Case Guarantees. Artificial Intelligence, 111(1-2), 209-238.

Shehory O. and Kraus S. (1995) Coalition formation among autonomous agents: Strategies and complexity, Lecture Notes in Artificial Intelligence no. 957, From Reaction to Cognition, (Castelfranchi & Muller, eds.), pp 57-72,

R. Smith (1980) The contract net protocol: High-level communication and control in distributed problem solver. IEEE Transactions on Computers, 29(12):1104--1113

Steinmetz, J. Collins, M. Gini, and B. Mobasher (1998) "An Efficient Anytime Algorithm For Multiple-Component Bid Selection in Automated Contracting" , In Agent Mediated Electronic Trading, LNAI1571, Springer, pp 105-125

Tambe M. (1997) Towards flexible teamwork. Journal of Artificial Intelligence Research, 7:83 – 124

Walker, E. C. T. (1999) Panel Report: Coalition Planning for Operations Other Than War, Workshop at AIAI, Edinburgh

Wooldridge M. (2000) Reasoning about rational agents, The MIT Press, Cambridge, MA,

Services Based Collaboration/Coalition Networks

Ramana Reddy, Professor and Director
Kankanahalli Srinivas, Research Associate Professor
Sumitra M. Reddy, Research Associate Professor
Lane Department of Computer Science and Electrical Engg.
West Virginia University,USA
{rar, srini, smr @ cerc.wvu.edu}
and
Rao Mikkilineni
enSANET, Inc.
Cupertino, California, USA
rao@ensanet.com

Abstract

We propose a novel framework and a scalable architecture that can be used for rapid formation of coalitions and to perform collaborative transactions. Our framework combines the traditional theory of organizations with the theory of signaling from telephone networks to bring about a unifying theory of collaboration to enable dynamic virtual enterprises formed on-demand. The new framework, overcomes many limitations of the traditional frameworks. Our framework known as the PPP/SST denoting the various components of the framework can be realized on the Internet.

1. Introduction

The term *Collaboration* is defined as the act of "working jointly with others" in the Webster's Collegiate Dictionary. It also defines "Coalition" as a "temporary alliance of distinct parties for joint action." While these two words are often used interchangeably, the word *coalition* brings out an important nuance, that of a "temporary alliance" which takes on a great significance in the light of an emerging business paradigm known as *Virtual Corporation*. Virtual corporations (or enterprises) are based on the notion that different parties can come together (i.e., form a coalition) temporarily to accomplish a "mission" and then move on to form possibly different coalitions to accomplish other missions. In this business paradigm one can envision a universe of freelancers offering specialized services to a variety of coalitions on-demand. For example, a training enterprise can form a temporary coalition of freelancing professors to develop a specialized seminar to meet a demand for a topic that may be in vogue. If the demand is ephemeral, so will be the coalition.

To enable this type of opportunistic, on-demand, and temporary coalitions we need to develop a more natural, holistic group-work paradigm that goes beyond the traditional models of Computer Supported Co-operative Work (CSCW) embodied in such products and technologies as NetMeeting [1], whiteboards (1], application sharing [1], erooms [2], Grove Nets [3] and others. All these focus on bringing together, synchronously or asynchronously different parties to communicate and or share documents on the Net. On the other hand collaborations based on Agent Paradigms [4] rooted in Distributed Artificial Intelligence concepts [5] are based on formal communication languages and the presence of "intelligence" at collaborating nodes. We believe these products/technologies and pure agent paradigms do not allow us to fully realize the business potential of ephemeral collaborations. To address this, we propose a new collaboration paradigm based on the idea that an ephemeral collaboration could be accomplished by bringing together a set of network based services through "signaling" mechanisms that facilitate composition of services. In this paradigm, the Person-Agent continuum makes available a service to a coalition on demand as a result of requests (or signals) initiated by the "owner" of a coalition. In this sense, we can consider our paradigm bridging the gap between traditional CSCW and pure agent based collaboration. We call this the PPP/SST Coalition Paradigm. This paper is organized as follows: section 2 describes the PPP/SST framework together with its signaling and interaction model. Section 3 illustrates the various framework components. Section 4 describes the architecture of our framework. Section 5 outlines the infrastructure trends and assumptions for our framework.

2. The PPP/SST framework

The various components of the framework are autonomous entities that reside on a distributed infrastructure. The sequence of interactions among the various components of the framework is illustrated in Figure 1. The coalition episode *starts* with a virtual enterprise initiator creating and sending a Statement Of Purpose (SOP) message denoted by (1) in the diagram, which describes the intent for the coalition to the service mediator object. The SOP message will use a markup language such as XML and use a dictionary and an ontology that are understood by the various components of the framework. In situations wherein the vocabulary is unclear, a metadata service could be used to disambiguate the markup. In the most rudimentary setup, the person in the person-agent continuum could interpret the message and take the necessary action. The same will be true at all other nodes, which receive messages in pursuit of the proposed coalition.

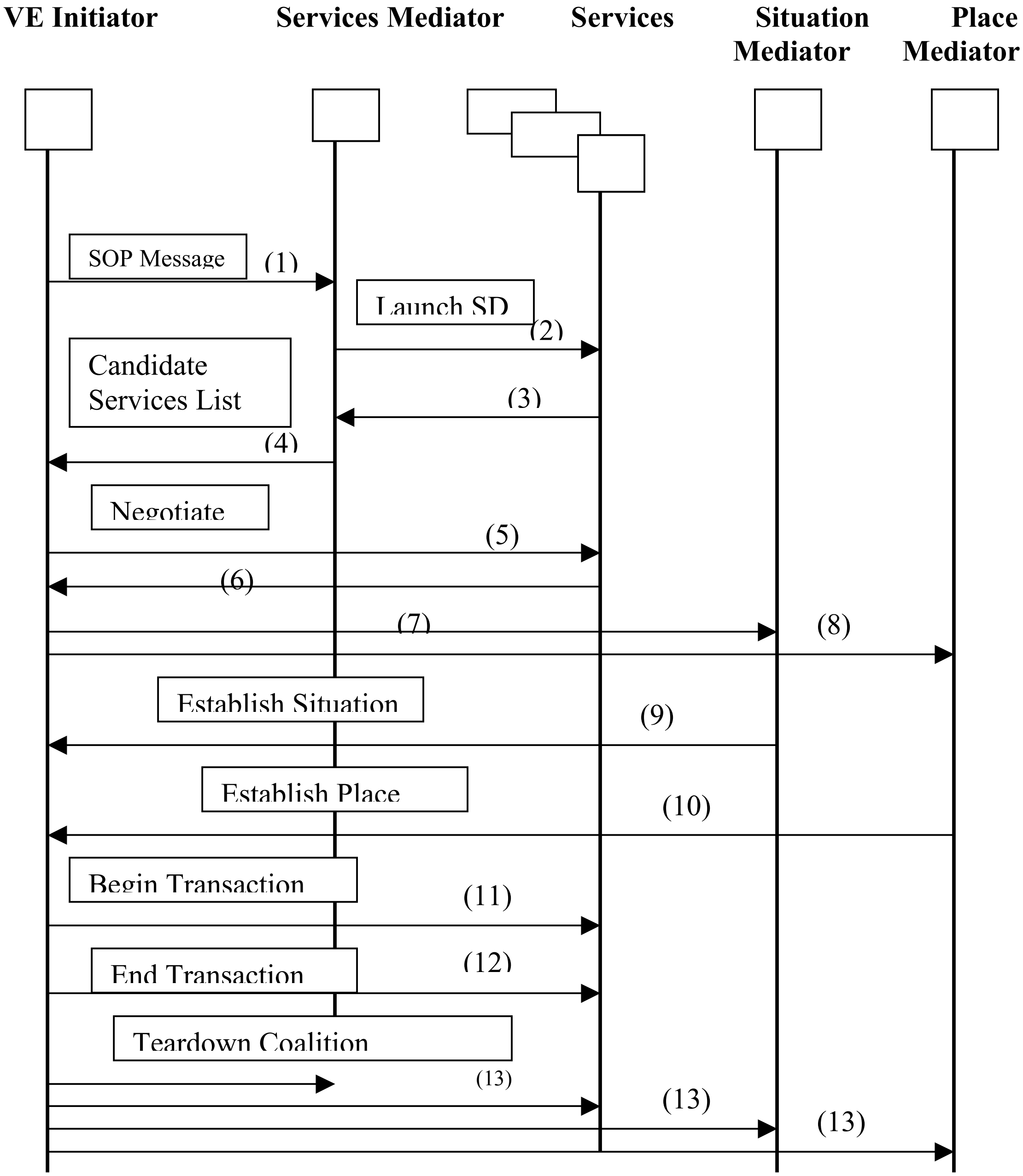


Figure 1. A Logical Message Sequence for Virtual Enterprise Transactions

The assumption here is that the services are described by a service description language such as Web Services Description Language (WSDL) [6]. After the service description is complete, the service registration is performed with a service registry, which is a logical conglomeration of services. The service registration infrastructure is assumed to also consist of proxy agents for services, redirect services (which can redirect requests to appropriate services) and other related services. Upon receipt of the SOP message, the Service Mediator initiates the service discovery process indicated by message labeled (2) in Figure 1. The service discovery process can have multiple bidding and negotiation phases and eventually should result in a set of possible candidate services, which constitute a set of possible coalition partners denoted by message(3) in Figure 1.

After the discovery of candidate services and their availability a negotiated agreement is reached among the coalition partners indicated by messages (5) and (6). Upon completion of the negotiation, a place and a situation are determined similarly by sending appropriate messages to the place and situation mediators denoted by messages (7) and (8). The messages (7) and (8) will contain the preferences and constraints that are imposed by the coalition partners. Messages (9) and (10) are used to establish the situation and place for the coalition, which satisfy the preferences and constraints imposed by the coalition partners as specified in messages (7) and (8). The business process potentially could move from situation to situation, and also from place to place during its lifetime. In a given situation and/or a place a series of transactions take place. These transactions are designed to fulfill the goals described in the SOP message. These transactions are denoted by message (11). Once the halting criteria such as meeting the goal or reaching the end of time allotted for the coalition task are reached, the transactions are ended as denoted by message labeled (12) and a series of teardown messages, labeled (13), are broadcast or multicast to all the members of the coalition. The coalition may be owned by one or more members of the group depending on the nature of the coalition. The entire process is under the control of the coalition owner(s) generating the events that move the coalition forward. This control mechanism could be modeled in a manner very similar to the control mechanism used in Discrete Event Simulation [7] or the control plane in communication networks.

Communication among persons (hereafter we use person to represent the person-agent continuum) is accomplished through message passing. Each situation results in a transcript that is linked with the transcripts of other situations through hyper-linking to form a longitudinal (or episodic) document. This document will be very similar to the Electronic Design Notebooks [8], [9] used in design enterprises. It is clear from this description of the proposed coalition framework, that there is a need to define a number of protocols to deal with description of services, signaling and organization of the transcript, which we assume can be a part of the underlying distributed infrastructure. We believe that extant standards such as BIZTALK [10], WSDL[6], SOAP [11]for services description, and SIP [12] for signaling can be successfully used in realizing implementations of this framework. However, we do not propose to develop yet another collection of standards as we do not feel they are absolutely necessary to realize the potential of this framework. Ad hoc coalitions could be built using agreed upon protocols among partners rather than striving for universal standards.

In the rest of this section we briefly describe the various components of this framework with examples where appropriate. Object descriptions are to be viewed only as representative and not as standard descriptions. To promote clarity of the key concepts we will use the following hypothetical example:

SpeedWay Software Company (SSC) advertises itself as a virtual enterprise which can undertake turnkey software development assignments. It has neither software professionals nor a significant computer infrastructure. It has a small executive staff with the necessary skills to undertake any complex software project. SSC's success is predicated on the discovery and composition of appropriate services from sources anywhere in the world and executing them somewhere in the cyberspace (i.e. the Place), (in fact places such as Sourceforge [13] which has some of the characteristics already exists on the Internet today) according to a script which itself may have been designed by some person serving as a coalition partner through the net. Under this framework, SSC's management is concerned with a series of tasks involving discovery of services, entering into contracts, and generating signaling events to apply previously contracted services until the halting criteria are met. A person providing a service may be modeled as a real person who receives messages in the form of emails or short messages that are understood by humans or could be modeled as an agent who can interpret these messages received according to an agreed upon protocol. In actual operation, we can envision a person as a continuum between a real person and an agent. Further, the agent could be a pure instruction executor or could be endowed with "intelligence" to decide on the action that should be executed autonomously or with the assistance of the agent's owner – a real person. Thus, SSC's operation could be summarized as a series of discoveries, contracts and signaling operations to appropriate persons or agents. This is the essential concept of our framework that bridges the gap between CSCW and pure agent based collaborations.

*3. PPP/SST Com*ponents

We will now describe briefly the general nature of various components of this framework. We do not define actual formats as they could be specified only in actual implementations of this framework.

3.1 Person/Agent

An agent in this framework may be described as an object, which is capable of providing a service and responds to different types of messages listed below:

(a) Registration Messages: This class of messages is intended for the agent to publish its service in a format that is based on a known markup language and a discernable ontology.

(b) Discovery Messages - This class of messages is designed to enable a VE to discover prospective agents that can become coalition partners. Since a universal description of services may not be available in some situations, these messages may likely be handled by a person.

(c) Contract Messages – This class of messages is intended to discover the availability of an agent, and determine the terms of contract and the method of providing services.
(d) Execution Messages – This class of messages is designed to enable a service provider to execute operations that result in the provision of services. These could also be viewed as transactions of the coalition corresponding to messages (11) and (12) in figure 1.

In the case of SSC, we can envision a number of agents such as problem analysts, system designers, programmers, testers, documentation experts and support providers. All of these agents could be drawn from different sources and "assembled" for a particular coalition episode. From an implementation point of view the agent could be totally subsumed by the application layer of the standard TCP/IP stack.

3.2 Purpose

Purpose could be modeled as a goal of the particular coalition episode, which also specifies when the coalition episode will be terminated using a termination or completion clause. The purpose should specify the goals using appropriate markup language and ontology, such that the goals may be easily mapped to services by the service mediator. For example, in the case of SSC, the purpose of an episode could be the creation of a fully documented and tested program artifact that accomplishes a certain goal. The episode comes to an end when the last task (say, acceptance testing) takes place.

3.3 Place

Place could be modeled as a workspace in cyberspace where artifacts created by various agents could be saved and or archived for use by other agents. The concept of eroom [2] is a good example of Place, though it provides primarily project management capabilities. In the general case we could view this as a workspace on a server that is organized as a collection of hyper linked documents. These could be perused, modified, transferred and otherwise manipulated by coalition partners – subjected, to the constraints imposed by the coalition owner(s).

3.4 Situation

Situation could be modeled as a state corresponding to a subtask that is part of a coalition episode. In this model we can think of the coalition episode moving from Situation to Situation as a result of completion of tasks by different agents. This is somewhat analogous to different elements in a process plan that describes a manufacturing operation.

3.5 Signaling

Signaling is an important aspect of telecommunication networks, wherein protocols such as SS7 [14] are ubiquitous. It is an important component of this framework, to bring together coalitions rapidly and to teardown the coalition after the goals of the coalition are reached. This is similar to signaling in the telecommunication world, which is used primarily to set up and teardown calls. Services get executed when the owner of the service is “signaled”.

3.6 Transcript

Transcript is a document linking all the situations that are part of a coalition episode. The actual transcript could be created by a transcription service, which is a part of the coalition. The transcript can be used as a basis for management by the agent charged with management as well as a record that may be needed for legal and /or corporate reasons. In addition the transcript could serve to document “corporate memory” [8] as in the case of the Electronic Design Notebooks.

4. The PPP/SST Architecture

Figure 2 shows the basic underlying P2P computational model of the PPP/SST coalition paradigm. All services needed by a coalition are available on the network edge and can act as clients or servers (depending on the role played by the service in the coalition). Services can also provide a place in the cyberspace for a particular transaction. Figure 3 shows the relationship among different components of the framework. These components are coupled through the networking infrastructure on which message exchange (i.e., signaling) takes place. Because of the inherent flexibility of the P2P computing paradigm, the location of the “place” where a particular task is being executed can be shifted around at will – thus ensuring continuation of the coalition activity in the presence of node failures. In addition, the distributed infrastructure and the protocols should support service-based collaboration in a robust and a secure manner. This is realized by supporting end-to-end Service Level Agreements (SLA) ,Quality of Service (QOS), in addition to new and novel mechanisms for service discovery .Our proposed PPP/SST framework assumes that the underlying networks support the discovery, mediation and compos ability of services. This can be easily achieved in IP(Internet Protocol) based networks such as the Internet.

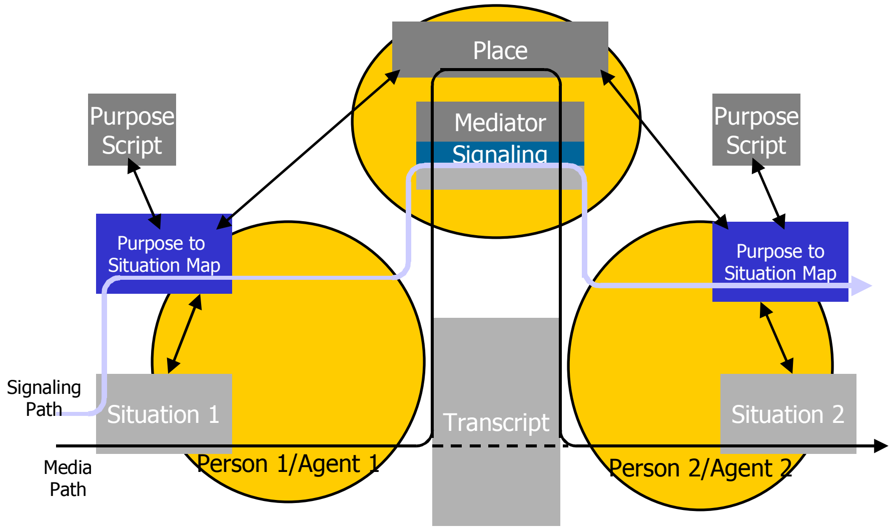


Figure 2. Architecture of PPP/SST

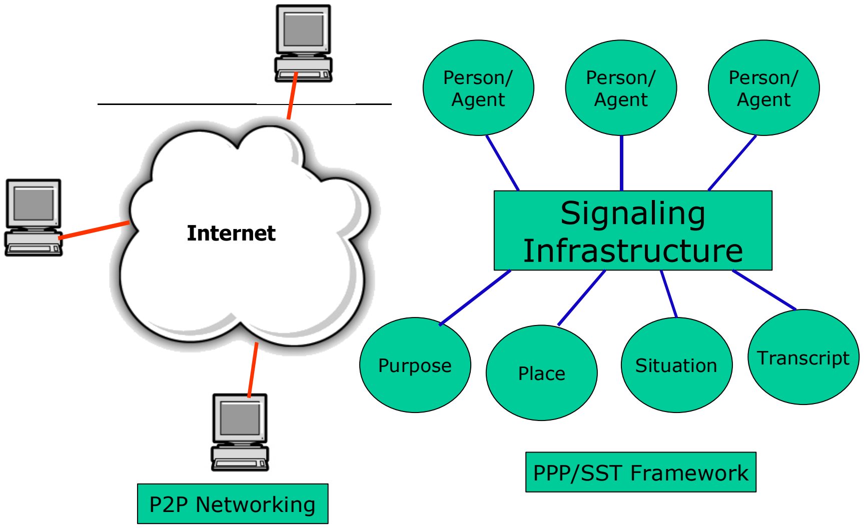


Figure 3. Architecture of PPP/SST

Figure 3 depicts the signaling and media paths in this architecture. Each Situation is attached to a situation map that connects a Situation with a Place where the transaction is being executed. Multiple Situations may be connected to the same Place through their respective mappings. Coordination between Situations is accomplished by executing signaling events that flow through the common place connecting the Situations. Each Place also includes a signaling mediator that ensures mutually contradictory signals are not executed. The transcript that results from this coordinated activity is created and associated with the Place of execution. As the coalition activity continues, these transcripts get hyper linked and periodically consolidated. The disposition of the transcript at the end of the coalition is determined by the agreements reached among partners at the outset.

The architecture presented in this section should be viewed only as representative and not definitive. The paradigm itself could be implemented in a variety of candidate architectures, depending on the capabilities of the enabling technologies. Regardless of the actual architecture, the paradigm ensures that the resulting system is highly adaptive and scalable. In the next section we present trends in the distributed infrastructures for service networks that will have a high degree of impact on the realizability of this framework.

5. Trends in Distributed Infrastructure of Service Networks

In the design of any distributed framework, such as the one proposed here, careful consideration should be given to infrastructure trends and issues, without which the framework will not be realizable. Traditionally, distributed infrastructure, such as the Internet and the telephone networks, have been designed with the intelligence in the core. The intelligence is usually embodied in the routers and switches. Due to the need to support switching and routing at optical speeds, the core is becoming “dumb” and the intelligence is being pushed to the periphery.

One of the benefits of pushing the intelligence to the edge is that it eases the deployment of new value-added services. The emergence of peer-to-peer architectures indicates that many new collaboration architectures will be based on the fact that the intelligence will reside on the edge of the network. The edge intelligence eases the deployment and management of these value-added services to the infrastructure provider such as the Internet Service Provider (ISP), a Network Operating center (NOC) and others.

An important issue in the design and implementation of a framework, such as the one described here is ensuring *trust* among the parties in the coalition and also to ensure the *security of the transactions* that take place in the coalition. To ensure trust mechanisms such as Public Key Infrastructure (PKI), biometric authentication and kerberos like technologies may be part of the underlying infrastructure. The Distributed Denial of Service (DDOS) attack prevention will be one of the crucial assumptions required for coalitions to work.

6. Conclusion

Rapid advances in computational paradigms, protocols, enabling technologies, infrastructures and precipitous drop in the cost and size of computing equipment are giving rise to the dawn of a new era in enterprise theory and practice. Advances in P2P computing, Remote Desktops, Short Messaging Systems, Intelligent Agent frameworks, mark-up languages like XML, Web Services Description Language (WSDL), Simple Object Access protocol (SOAP) and signaling protocols like SIP are some of the key drivers of this trend. The ease with which a person who has a service to offer without being employed by a conventional enterprise is going to rapidly make the employee-enterprise relationship obsolete in a number of disciplines. For example, in the healthcare domain, many radiologists already operate in this mode of service by making themselves available via the Net. Distance education enterprises are beginning to offer the services of *best professors* to students around the world – often through coalitions.

The authors strongly believe this trend towards enterprises, as opportunistic coalitions as opposed to conventional enterprises will gain acceptance rapidly – at least in certain disciplines where information exchange is the key transaction.

7. References

[1] Microsoft NetMeeting http://www.microsoft.com/windows/netmeeting/

[2] erooms http://www.eroom.com/

[3] Grove Networks www.grove.net

[4] Prasannaa Thati, Po-Hao Chang, Gul Agha: Crawlets: Agents for High Performance Web Search Engines. Mobile Agents 2001: 119-134

[5] Readings in Distributed Artificial Intelligence A.H. Bond & L. Gasser (Editors), Morgan Kaufmann, San Mateo, CA, 1988.

[6] WSDL Web Services Description Language http://www.w3.org/TR/wsdl

[7] Discrete Event Simulation Control http://protodesigninc.com/SGmanuals/sansgui_simulation_control.htm

[8] Uejio, Wayne et a l An Electronic Project Notebook from the Electronic Design Notebook (EDN), *In Proceedings of the Third National Symposium on Concurrent Engineering, CALS and CE, Washington DC 1991, pp. 527-535.*

[9] Toye, G. et al. "SHARE: A Methodology and Environment for Collaborative Product Development. In IEEE Proc. Of Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. 1993

[10] http://www.vbxml.com/b2b/articles/biztalk_unleashed/3_whatisbiztalk.asp

[11] Simple Object Access Protocol SOAP http://www.w3.org/TR/SOAP/

[12] Session Initiation Protocol http://www.cs.columbia.edu/sip/

[13] SourceForge www.sourceforge.net

[14] G. Christensen, *SS7 Technology and Applications*, Mobile Streams Publishers, Feb 2000, ISBN 244933 44344.

The DARPA Control of Agent Based Systems (CoABS) Program and Challenges for Collaborative Coalitions

LCDR Dylan Schmorrow, Ph.D.

DARPA, Information Technology Office
dschmorrow@darpa.mil

As the number and availability of information sources (COTS[1], GOTS[2] and military special-purpose) increase, current military command and control systems including those supplemented with commercial off-the-shelf technology are overburdened in an effort to bring the right information to the right participants at the correct time. Military operators bemoan their inability to find mission-critical intelligence and operations information, which must be both manually filtered and routed. Providing better integration, at an information management level, between diverse information systems is a key to providing the information superiority needs as described in "Implementing Joint Visions 2010." To increase the military user's productivity and, by extension, our military capability, we need a next generation of software which is able to help users deal with complex tasking. The system must help the warfighter get needed information, help the user solve difficult problems, route useful information and otherwise enable more informed and rapid. Arising from research in the areas of distributed artificial intelligence and mobile software are computational "agents" designed to provide these capabilities. For the military, software agents will be critical force multipliers that free military personnel from having to do simple tasks which can be automated and assist personnel with difficult tasks. And as our military forces are drawn down, software agents will become increasingly important for retaining our ability to meet crises effectively.

A crucial need for the modern military is the ability to rapidly assemble a set of disparate information systems into a coherently interoperating whole. This must be done without system redesign and may include interoperation with non-DoD governmental systems, with systems separately designed by coalition partners, or with COTS and open-source systems that are not built to a pre-existing government standard. The Control of Agent Based Systems (CoABS) program explores the technical underpinnings of such run-time interoperability of heterogeneous systems, and develops new tools for facilitating rapid system integration in practice. As large-scale integrated systems are deployed, greater stress is placed on the communications infrastructure and on the management of information resources across the system. Techniques developed for agent-based computing, particularly those of mobile agents and agent-communication languages, will help both in the facilitation of this multi-systems integration and in controlling the information flow to alleviate bandwidth saturation and degraded quality of service.

The CoABS program goal – to achieve a comprehensive and scalable approach to software agent interoperability is divided into the following three tasks. (1) Agent Grid. The objective of this task is to develop a set of tools as the basis for upgrading military legacy systems to exploit agent technology using the concept of a "grid adapter." The grid adapter minimizes the integration effort required by focusing on the connection mechanisms instead of the client components. This involves wrapping legacy systems using a middleware approach which is service-based and which includes logging/reporting tools. (2) Agent Interoperability Standards. The objective of this task is to define standards to support agent interoperability, including agent-human interaction, agent-agent communication, agent-software interfaces, and agent management and control. And, finally, (3) Scaling of Agent Control Strategies. This task develops and tests agent control strategies for monitoring, coordinating, controlling, and managing agent collections, ranging from simple tasks involving the cooperation of small agent teams to highly complex interactions involving thousands of individual agents. This task also provides guaranteed behaviors for agents, even in unreliable networks. Areas of interest include knowledge sharing techniques; team formation and coordination through modeling of plans, commitments, and intentions; and computational markets including protocols for auctions and voting.

Success in military operations involves carrying out high-tempo, coherent, decisive actions and information is a key enabler in this process. In addition to the problems of integrating single-service and Joint capabilities into a coherent force, the nature of Coalition (multi-national) operations implies a need to rapidly configure incompatible, "come-as-you-are" or foreign systems into a cohesive whole in an open, heterogeneous, diverse and dispersed environment. DARPA is researching the use of agents within Coalitions, working collaboratively with the 16 partners of an international Coalition Agents Experiment (CoAX) (Allsopp et. al. 2001; Allsopp et.al. 2002).

[1] COTS = Commercial Off-The-Shelf.

[2] GOTS = Government Off-The-Shelf.

It is always perilous to predict the future, but it is also foolish to ignore clear trends that surely will affect the future. Thus it is predicted that there will be increasingly capable physical robotic things, software agents of complex functionality, and humans. The challenge is to bring these three into a synergistic synchronization that makes the whole -- the team -- capability greater than the sum of the parts. It is the thesis of this presentation that to achieve such an objective will require interdisciplinary research and development approaches that exceed such efforts attempted in the past. It is critical that barriers to collaborative efforts, however unintentional, be eliminated. This will require recognition that the goal is real, worthwhile, and to be sought as a military and economic exigency.

During this final year in the CoABS program, the focus will be placed on demonstrating all of the technologies that have been developed during the last five years, as well as the transition programs that anxiously await this technology. Transition programs include Joint Experimentation Programs, Navy Expeditionary Sensor Grid, Air Force Joint Battle Infosphere, and various Army Agent Based programs.

References

Allsopp, D., Beautement, P., Bradshaw, J.M., Carson, J., Kirton, M., Suri, N. and Tate, A. (2001) "Software Agents as Facilitators of Coherent Coalition Operations", *6th International Command and Control Research and Technology Symposium*, US Naval Academy, Annapolis, Maryland, USA, 19-21 June 2001.

Allsopp, D., Beautement, P., Bradshaw, J., Durfee, E., Kirton, M., Knoblock, C., Suri, N., Tate, A. and Thompson, C. (2002) "Coalition Agents eXperiment: Multi-agent Co-operation in an International Coalition Setting*", 2nd International Conference on Knowledge Systems for Coalition Operations*, Toulouse, France, 23-24 April 2002.

I-P^2 - Intelligent Process Panels to Support Coalition Operations

Austin Tate, Jeff Dalton and Jussi Stader

Artificial Intelligence Applications Institute
Centre for Intelligent Systems and their Applications
Division of Informatics, The University of Edinburgh
80 South Bridge, Edinburgh EH1 1HN, UK
a.tate@ed.ac.uk, j.Dalton@ed.ac.uk, j.stader@ed.ac.uk

Abstract. I-X is a research programme with a number of different aspects intended to create a well-founded approach to allow humans and computer systems to cooperate in the creation or modification of some product or products such as documents, plans or designs. I-X may also be used to support more general collaborative activity.

The I-X research draws on earlier work on O-Plan (Tate et.al., 1998; Tate et.al., 2000; Tate et.al., 2002), <I-N-OVA> (Tate, 1996), the Enterprise Project (Fraser and Tate, 1995; Stader, 1996); Uschold, et.al., 1998) and the TBPM project (Stader, 2000) but seeks to make the framework generic and to clarify terminology, simplify the approach taken, and increase re-usability and applicability of the core ideas.

I-X Applications are being studied in a variety of areas. These currently include:

- Coalition Operations (CoAX: I-LEED, I-DEEL)
- Emergency and Unusual Procedure Assistance (I-Rescue)
- Help Desk Support (I-Help)
- Multi-Perspective Knowledge Modelling and Management (I-AKT)
- Contextualised Presentations of Procedures and Plans (I-Tell)
- Collaborative Meeting and Task Support (I-Room, I-Space)

An application of I-X Process Panels within a military Coalition context - part of the Coalition Agents eXperiment - CoAX (Allsopp et.al., 2001; Allsopp et.al., 2002) will be described in this paper.

1 I-X Research Programme

I-X is a research programme with a number of different aspects intended to create a well-founded approach to allow humans and computer systems to cooperate in the creation or modification of some product such as a plan, design or physical entity – i.e. it supports **synthesis tasks**. I-X may also be used to support more general collaborative activity.

The I-X research draws on earlier work on O-Plan (Tate et.al., 1998; 2000; 2002). <I-N-OVA> (Tate, 1996) and the Enterprise Project (Fraser and Tate, 1995; Uschold, et.al., 1998) but seeks to make the framework generic and to clarify terminology, simplify the approach taken, and increase re-usability and applicability of the core ideas.

The I-X research programme includes the following threads or work areas:

1. **I-Core**, which is the core architecture, the underlying ontology of activity and processes termed <I-N-CA>, and the terminology used to describe applications, systems or agents built in the I-X framework.

2. **I-PE**, which is the I-X Process Editor, which is itself an I-X application but is also used to create and maintain the process models and activity specifications used elsewhere.

3. **I-P^2**, which are I-X Process Panels used to support user tasks and cooperation.

4. **I-Plan**, which is the I-X Planning System. This is also used within I-P^2 and other applications as it provides generic facilities for supporting planning, process refinement, dynamic response to changing needs, etc.

5. **I-Views**, which are viewers for processes and products, and which are employed in other applications of I-X. I-Views can be for a wide range of modalities and types of user.

6. **I-Faces**, which are underlying support utilities to allow for the creation of user interfaces (User I-Faces), inter-agent communications (Communications I-Faces) and repository access (Repository I-Faces).

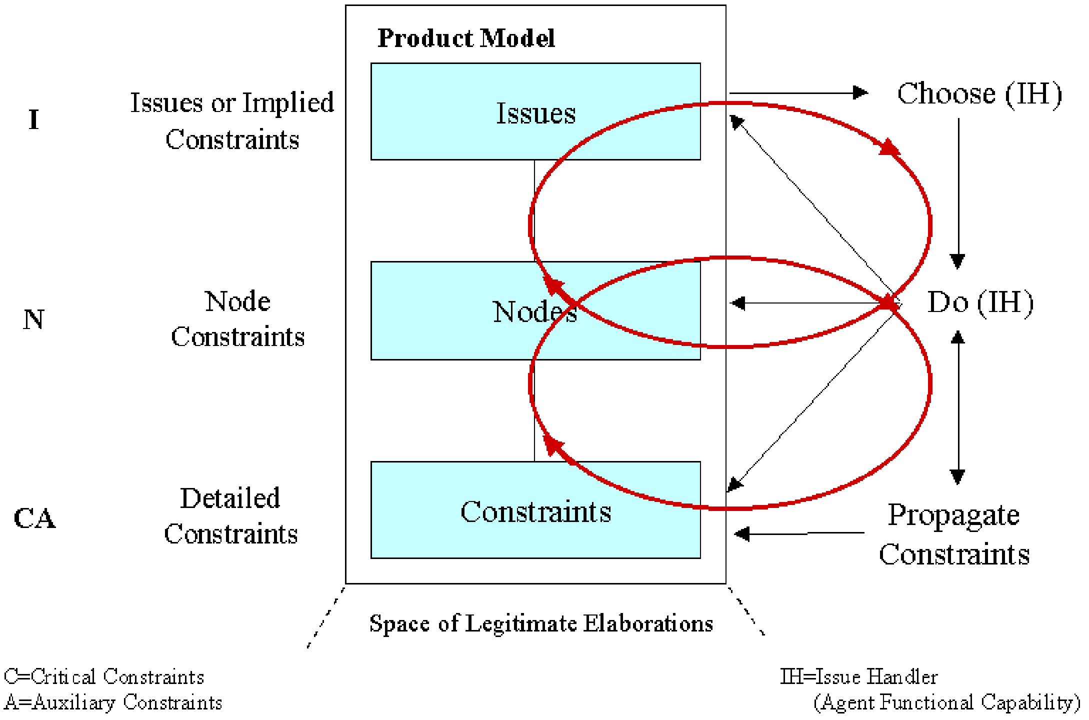
7. **I-X Applications** of the above work areas in a variety of areas. These currently include:
   a. Coalition Operations (CoAX: I-LEED, I-DEEL)
   b. Emergency and Unusual Procedure Assistance (I-Rescue)
   c. Help Desk Support (I-Help)
   d. Multi-Perspective Knowledge Modelling and Management (I-AKT)
   e. Medical Best Practice Procedures or Protocols (I-Medic)
   f. Natural Language Presentations of Procedures and Plans (I-Tell)
   g. Collaborative meeting and task support (I-Me, I-Room and I-Space)

8. **I-X Student Projects**, which are deepening and refining a number of aspects of the I-X research programme.

9. **I-X Technology Transfer**, including work on standards committees, especially for process, plan, activity and capability models.

2 I-X Approach

The I-X approach involves the use of shared models for task directed cooperation between human and computer agents who are jointly exploring (via some processes) a range of alternative options for the synthesis of an artifact such as a design or a plan (termed a product).



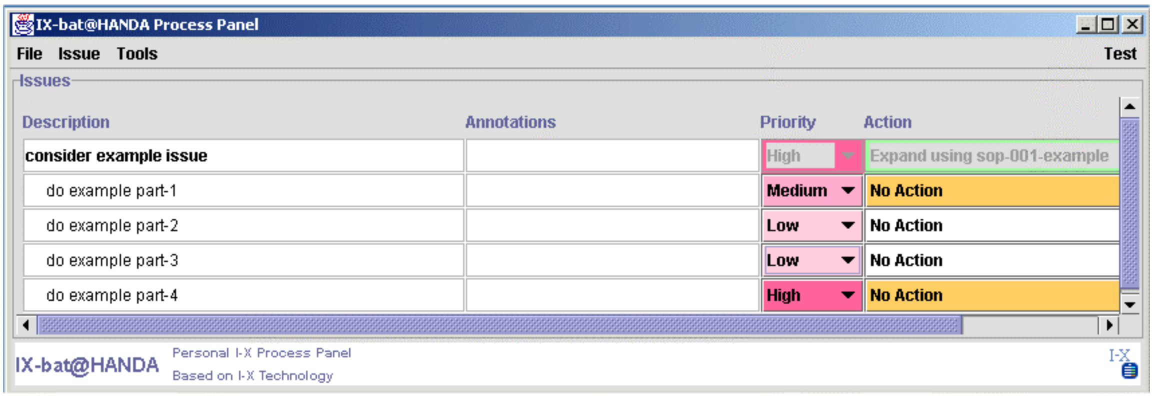
- An I-X system or agent has two cycles:
  - Handle Issues
  - Respect Domain Constraints

- An I-X system or agent carries out a (perhaps dynamically determined) process that leads to the production of (one or more alternative options for) a synthesised artifact.

- An I-X system or agent views the synthesised artifact as being represented by a set of constraints on the space of all possible artifacts in the domain.

I-X also involves a modular systems integration architecture that strongly parallels and supports the abstract view described above.

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



Description	Annotations	Priority	Action
consider example issue		High	Expand using sop-001-example
do example part-1		Medium	No Action
do example part-2		Low	No Action
do example part-3		Low	No Action
do example part-4		High	No Action

The aim of an I-X Process Panel (I-P^2) is to act as a workflow, reporting and messaging "catch all" for its user. It can act in conjunction with other panels for other users if desired.

- Can take ANY requirement to:
  - Handle an issue
  - Perform an activity
  - [later: Add a constraint]

- Deals with these via:
  - Manual (user) activity
  - Internal capabilities
  - External capabilities (invoke or query)
  - Reroute or delegate to other panels or agents (pass)
  - Plan and execute a composite of these capabilities (expand)

- Receives reports and messages and, where possible, interprets them to:
  - Understand current status of issues, activities and constraints
  - Understand current world state, especially status of process products
  - Help control the situation

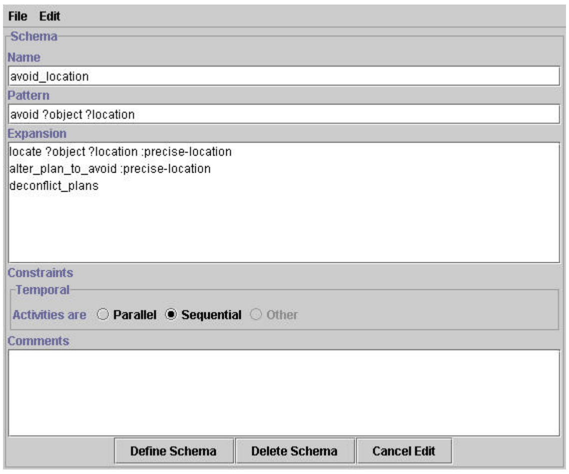
- Copes with partial knowledge

An I-X Process Panel supports a user or collaborative users in selecting and carrying out "processes" and creating or modifying "process products". Both processes and process products are abstractly considered to be made up on **"Nodes"** (activities in a process, or parts of a process product) which may have parts called sub-nodes making up a hierarchical description of the process or product. The nodes are related by a set of detailed **"Constraints"** of various kinds. A set of **"Issues"** is associated with the processes or process products to represent unsatisfied requirements, problems raised as a result of analysis or critiquing, etc. Processes and process products in I-X are represented in the <I-N-CA> (Issues - Nodes - Critical/Auxiliary) Constraints Model of Synthesised Artifacts.

Three example process panels are shown in the figure below. These panels are from a demonstration of agent systems within a military Coalition context – part of the Coalition Agents eXperiment – CoAX (Allsopp et.al., 2001; Allsopp et.al., 2002).

4 I-X Process Editor (I-PE)

The process descriptions used by I-X Process Panels are kept in a domain library. This can be loaded when a panel is started, and can be added to dynamically by a user of a panel.



Schema

Name: avoid_location

Pattern: avoid ?object ?location

Expansion:
locate ?object ?location :precise-location
alter_plan_to_avoid :precise-location
deconflict_plans

Constraints

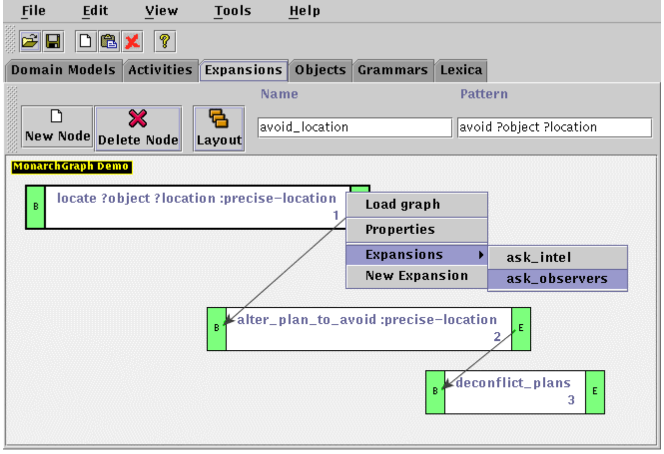
Temporal

Activities are ○ Parallel ● Sequential ○ Other

Comments

Simple View - the process panels contain a simple, form-based domain and process editor (right). This simple editor allows simple task breakdown structures to be specified along with a temporal constraint that the sub-steps should all be sequentially ordered or all kept in parallel.

Advanced View - a more powerful domain and process editor allows for multiple perspectives and views to be used to create rich process models beyond those that can be created with the simple view editor. This can be reached by selecting advanced view from the simple domain/process editor. It is also available as a stand-alone application to maintain a set of domain and process libraries.
The advanced editor consists of a form-based structure editor (not shown), which looks similar to the simple editor but allows the user to specify more complex temporal constraints. Other constraints, like spatial ones or constraints on resources, can also be specified using the advanced view.



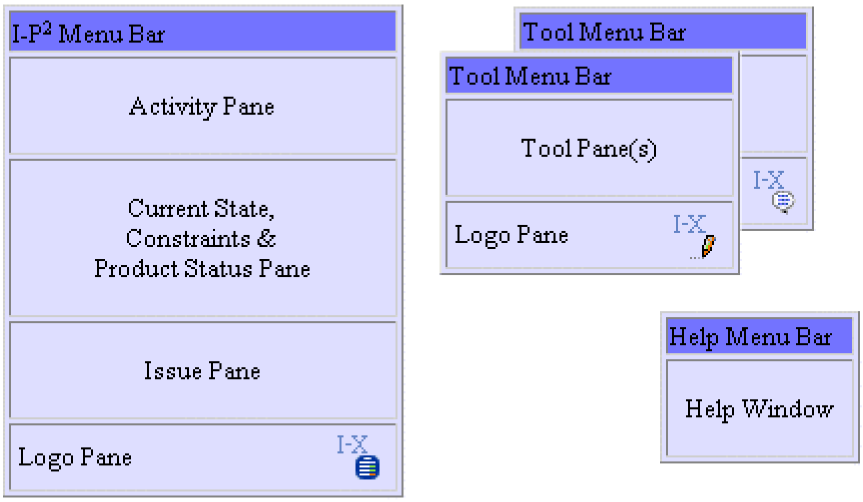
Name: avoid_location

Pattern: avoid ?object ?location

The graphical editor (right) provides an alternative view to the form-based editor. The graphical editor illustrates precedence relationships between the sub-steps of a process. This editor can also be used to specify task breakdown structures via the expansion of nodes in the graph. Full details of the process and its sub-steps can be accessed via the properties of nodes.

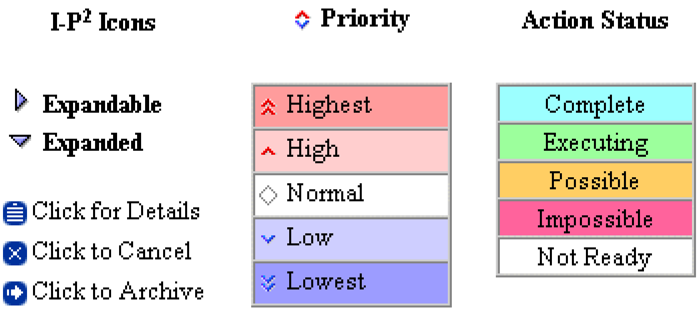
Use of XML and Text Editors - the process and domain models are maintained in XML. You can also modify them using an XML Editing Tool - such as the freely available Microsoft XML Notepad (see http://msdn.microsoft.com/xml/notepad/intro.asp) or a text editor.

5 I-P^2 Generic Approach



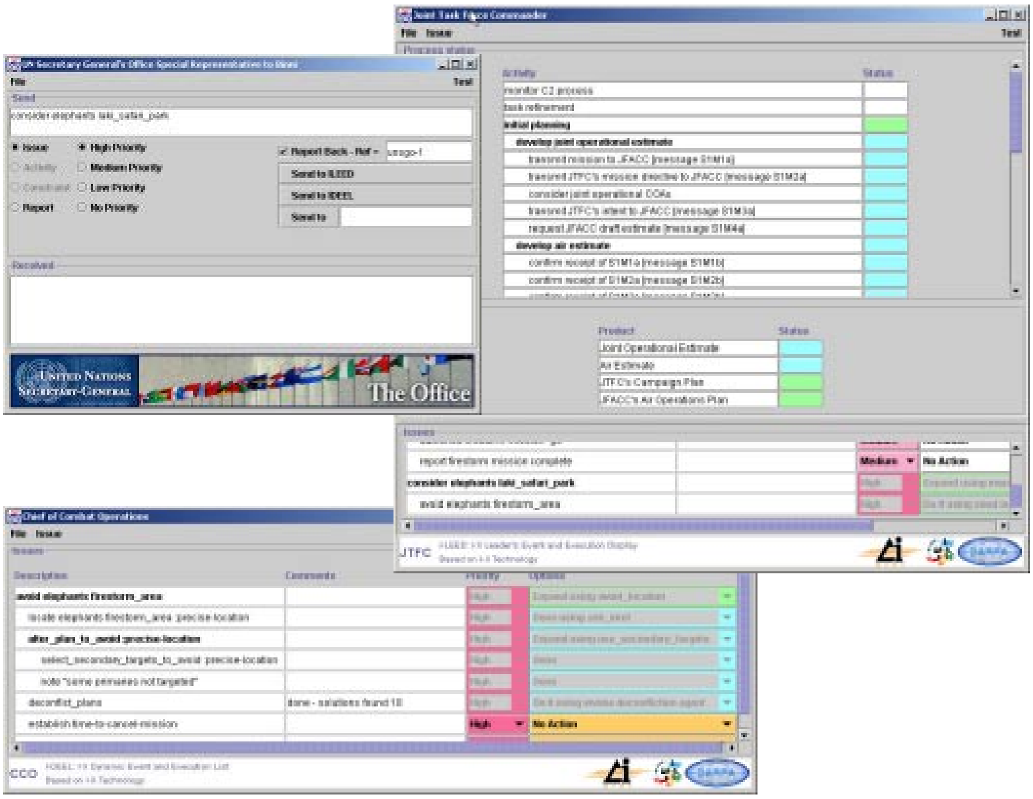
An I-X Process Panel has a menu bar and a number of sub-panels. These can include an activity pane that describes the list of activities to be carried out. Alternative actions to take to perform these activities may be available. A current state pane can be included to describe the current situation, and in particular it may describe the status of the various "process products" being created or modified by the user of the panel. The list of outstanding issues can be included in a pane, and this "to do list" often is at the heart of I-X process panels, and is usually present in all applications. Finally a logo pane can be added to customise the process panel to specific applications.

From the menu bar it is possible to activate a number of "tools" which currently include a free format instant messaging/chat tool, and access to the Process Editor (in Simple or Advanced View varieties). Help is also available from the process panel menu bar.



The process panel includes a number of icons and tabular entries to assist its user to maintain awareness of the current status of activities being executed, process product status and issue status. These are shown in the diagram here.

6 I-LEED and I-DEEL – A Coalition Application of I-X Process Panels


UN Secretary General's Office Special Representative to Binni

File Test

Send

consider elephants lake_safari_park

● Issue ● High Priority
○ Activity ○ Medium Priority
○ Constraint ○ Low Priority
○ Report ○ No Priority

☑ Report Back - Ref = ungo-1

Send to ILEED

Send to IDEEL

Send to

Received

Joint Task Force Commander

File Issue Test

Process status

Activity	Status
monitor C2 process	
task refinement	
initial planning	🟩
develop joint operational estimate	🟦
transmit mission to JFACC [message S1M1a]	🟦
transmit JTFC's mission directive to JFACC [message S1M2a]	🟦
consider joint operational COAs	🟦
transmit JTFC's intent to JFACC [message S1M3a]	🟦
request JFACC draft estimate [message S1M4a]	🟦
develop air estimate	🟦
confirm receipt of S1M1a [message S1M1b]	🟦
confirm receipt of S1M2a [message S1M2b]	🟦

Product	Status
Joint Operational Estimate	🟦
Air Estimate	🟦
JTFC's Campaign Plan	🟩
JFACC's Air Operations Plan	🟩

Issues

report firestorm mission complete		Medium	No Action
consider elephants lake_safari_park		High	Expand using [illegible]
avoid elephants firestorm_area		High	Do it using [illegible]

JTFC I-LEED: I-X Leader's Event and Execution Display
Based on I-X Technology

Chief of Combat Operations

File Issue

Issues

Description	Comments	Priority	Options
avoid elephants firestorm_area		High	Expand using avoid_location
locate elephants firestorm_area precise-location		High	Done using [illegible]
alter_plan_to_avoid precise-location		High	Expand using use_secondary_targets
select_secondary_targets_to_avoid precise-location		High	Done
note "some primaries not targeted"		High	Done
deconflict_plans	done - solutions found 18	High	Do it using [illegible] deconfliction agent
establish time-to-cancel-mission		High	No Action

CCO I-DEEL: I-X Dynamic Event and Execution List
Based on I-X Technology

The CoAX demonstration of the I-X concepts is grounded in a system for supporting event management in a highly dynamic military Coalition environment. Two I-X process Panels are involved in the demonstration. One is called the “I-X Leaders Event and Execution List” – I-LEED – and support the Joint Task Force Commander (JTFC) in the Coalition HQ. The other is called the "I-X Dynamic Execution Event List" – I-DEEL – and support the Chief of Combat Operations (CCO). A third system is provided to act as a source for events and messages to initiate the demonstration. Notionally this acts as the UN Secretary general’s Office Special Representative to Binni – the region where the Coalition mission is taking place (Rathmell, 1999).

The process panels support their respective users in mapping events to actions they decide are appropriate to deal with such events. The panels have some (partial) level of process knowledge in a simple process library, and a way to create / expand task lists / processes on the fly which are dependent on the context or situation that is prevailing at the time. The process panels are designed to be able to be used by any decision-maker operating at different time scales and with appropriate abstraction levels of process description to support people involved in military Command and Control in Coalitions and other operations.

The process panels use the issue-addressing core of I-X to handle issues (derived from externally generated events or user initiated ones) relevant to a Coalition C^2 process within the context of the CoAX Binni scenario. Where these do not match directly to a known capability, the panel seeks (or the user could input) process / task expansions of how to handle these issues and use a very simple expansion engine (a mini-planner termed I-Plan) to match the expanded activities to a range of known capabilities which are performable by the process panel user or by other colleagues or to suitable tasks / solutions which the user could input. Therefore, in a simple way, a process panel can dynamically generate an appropriate response to the issue or event in the current situation - this allows the user to create and interact with a "dynamic event list" to assist with the monitoring of execution outcomes and the resultant actions / changes / new taskings. Links can be created between related tasks (by the user or inferred by the system) and the system can monitor dependencies, etc.

The process panels can identify actions based on known external capabilities to enable the user to "enact" these steps. The process panels can maintain a simple display of the current status of issues and events delegated to the panel and information on how far along in the response process things had proceeded.

7 Summary

This paper has described the application of I-X Process Panels to military Coalition scenarios. Such process panels can be employed quickly and with partial knowledge to connect together “come-as-you-are” participants and systems together, especially in contexts where physical connectivity of systems is too time consuming, or is not allowed due to security constraints. As process and other knowledge is made available improved interoperability can be supported – allowing for more intelligent task and process management in a loose collaborative setting.

Acknowledgements

The I-X project is partially sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Command and Control Directorate under grant number F30602-99-1. The US Government and the University of Edinburgh are authorised to reproduce and distribute reprints 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 express or implied, of DARPA, the Air Force Research Laboratory, the US Government or the University of Edinburgh.

References

Allsopp, D., Beautement, P., Bradshaw, J.M., Carson, J., Kirton, M., Suri, N. and Tate, A. (2001) “Software Agents as Facilitators of Coherent Coalition Operations”, *6th International Command and Control Research and Technology Symposium*, US Naval Academy, Annapolis, Maryland, USA, 19-21 June 2001.

Allsopp, D., Beautement, P., Bradshaw, J., Durfee, E., Kirton, M., Knoblock, C., Suri, N., Tate, A. and Thompson, C. (2002) “Coalition Agents eXperiment: Multi-agent Co-operation in an International Coalition Setting*”, 2nd International Conference on Knowledge Systems for Coalition Operations*, Toulouse, France, 23-24 April 2002.

Fraser, J. and Tate, A. (1995) "The Enterprise Tool Set - An Open Enterprise Architecture", *Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, August 1995.

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.

Stader J., Moore J., Chung P., McBriar I., Ravinranathan M., Macintosh A.. (2000) "Applying Intelligent Workflow Management in the Chemicals Industries"; in *"The Workflow Handbook 2001"*, L. Fisher (ed), Published in association with the Workflow Management Coalition (WfMC), pp 161-181, Oct 2000.

Stader J. (1996) "Results of the Enterprise Project", in *Proceedings of Expert Systems '96, the 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems*, Cambridge, UK, December 1996.

Tate, A. (1996) "The <I-N-OVA> Constraint Model of Plans", *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS-96)*, (ed. Drabble, B.), pp. 221-228, Edinburgh, UK, May 1996, AAAI Press.

Tate, A. (1998) "Roots of SPAR", in "Special Issue on Ontologies", *Knowledge Engineering Review*, Vol.13 (1), March 1998, Cambridge University Press.

Tate, A., Dalton, J. and Levine, J. (1998) "Generation of Multiple Qualitatively Different Plan Options", *Fourth International Conference on AI Planning Systems (AIPS-98)*, Pittsburgh, PA, USA, June 1998.

Tate, A., Dalton, J. and Levine, J. (2000) "O-Plan: a Web-based AI Planning Agent", AAAI-2000 Intelligent Systems Demonstrator, in *Proceedings of the National Conference of the American Association of Artificial Intelligence (AAAI-2000)*, Austin, Texas, USA, August 2000.

Tate, A., Levine, J., Dalton, J. and Nixon, A. (2002) "Task Achieving Agents on the World Wide Web", in "*Creating the Semantic Web*", Fensel, D., Hendler, J., Liebermann, H. and Wahlster, W. (eds.), MIT Press, 2001.

Uschold, M., King, M., Moralee, S. and Zorgios, Y. (1998) "The Enterprise Ontology", in "Special Issue on Ontologies", *Knowledge Engineering Review*, Vol.13(1), March, 1998, Cambridge University Press.

Advanced 3D Visualization Web Technology and its Use in Military and Intelligence Applications

Dusan Toman

DIMENSION 5, Ltd. (Miner3D.com)
Hurbanova 36 SK-920 01 Hlohovec, Slovakia
dusan@miner3D.com

Dr. Alexander Kos

AKos Consulting & Solutions GmbH
Austr. 26, D-79585 Steinen, Germany
alexander.kos@akosgmbh.de

Abstract. Web technologies achieved significant improvements in last years, but many application areas are not yet Web-impacted. Upcoming software products enhance feature sets of Web browsers and make it possible to use systems based on new Web technologies as advanced application framework for complex information retrieval, control, monitoring or analysis systems.

In this paper, we illustrate a new interactive, high-density and information-centric user interface. We show the communication protocol and the architecture of a new 3D Web authoring software. Some applications are discussed with special focus on military and intelligence applications.

Keywords: 3D, Web, user interface, visualization, navigation, real-time

1. Introduction

The boom of the Internet in 90s brought the networking infrastructure to almost every business computer. Even about 60 percent of the households in the United States, the world's most wired country, have Internet access. (Howe, 2001) Today's Web technologies give us many advantages: networking is now system independent, the Internet is nearly everywhere, it is accessible, extremely easy-to-use and even affordable. Many millions of users have the new knowledge and use it daily.

2. User Interface Issues

Unfortunately, the great accessibility is paid by many limitations. Today's Web is built mostly on HTML. The language was designed in early 90s and its main goal was to describe static documents consisting of formatted texts and pictures. It becomes evident now that problems with user interface influence that otherwise very positive feeling.

The expectations were set too high. Is a Web document reading really so different from the old, hard-encoded experience of reading texts written on paper? Static Web pages offer essentially the same content presented on paper, which makes the online experience more like reading in a dusty library than participating in a new medium. (Howe, 2001) The way we use computers today mirrors the way we used to read and write in traditional paper media of the past. Using the simple analogy of the monitor as the paper, the keyboard as the pen, and the act of scrolling as turning a page, we can see how this kind of human interaction with technology is not as advanced as it can be in 21st century.

Mostly because of the reasons mentioned above, many important application areas are not yet Web-impacted, or the real impact is far under our previous expectations. The flat and static nature of HTML pages prevents also from using that, otherwise powerful and effective infrastructure, in series of high-end military and intelligence applications.

It becomes evident that people need much more of freedom for working with data than it is common on the today's Web. They intuitively tend to more natural way of presenting complex information. They feel that some kind of physical-like user interface, where more perceptual abilities of human can be deployed, could communicate information more effectively.

3. Extended Web

Do we remember the early days of television when TV programs looked like radio with announcer's picture? Similar evolution as that of TV is probably ahead of the Internet too. (Howe, 2001) It is necessary to see some technological enhancements to be able developing reasonably advanced Web applications that go beyond simple HTML hypertexts, while still have it quickly to train, easy to install, use and fully deployable. These enhancements will allow users to get real-time, interactive, less mediated experience over the next Network applications.

On the Web browser side it requires to provide additional code able to quickly download and plug into the browser. This code enhances the standard user interface by live and fully interactive graphic space. The new space is preferred to be 3D, giving more space for displaying data and application controls. It is expected the 3D space is intelligent enough and uses included conventions of standard interactions and feedback. This feature enables to provide user with good natural experience without extra coding for specific applications.

On the server and network side, it is necessary to use transparent protocols and languages allowing fast and inexpensive implementation and usage. Technically, using of today's object technologies (COM, .Net, CORBA), standard Internet protocols (HTTP, HTTPS) and languages (XML, HTML) provides a good base for developing of the Extended Web building blocks.

4. Miner 3D SITE

Miner3D SITE software is our development of the visions of the extended Web. It is actually the evolution of our visual data mining software and represents our 15-year experiences with 3D graphics. The system enhances Web browser's user interface and allows building of fully interactive Web pages capable of displaying complex and real-time information.

There is couple of 3D standard-candidates (VRML, X3D…) available also for Web use, but it all defines scenes statically. In fact, it puts emphasize on the look of the graphics, on graphics effects and bells and whistles, and strongly underestimates the information itself. Such scenes are actually hard-coded and thus not suitable for fast downloads and for visualizations of data generated in real-time.

Miner3D SITE software now consists of two main parts:

- The Viewer software, which resides on the Web browser. It creates the 3D graphics window and provides the interactivity and feedback. It is a COM object and allows very easy, fast and transparent downloads. Presently it works only with Internet Explorer on Windows, but versions for Netscape and for Mac are under development.
- Communication protocol used to transfer model properties (visualization rules) and data (content of the visualization). We had to define our own XML-based M3D protocol, which allows us developing of the needed protocol syntax. Saving of network bandwidth is another of positive side effects.

Combining features of the Viewer and of the M3D protocol we are getting live and information-centric nature of the visualization. The scenes may not be defined only at design time anymore. The designer defines actually the visualization rules, while the final look of a scene is defined mainly by the real data returned from server.

The universal design of the software makes it possible to use the same software in various applications, from information retrieval systems, complex site navigation, through real-time monitoring and control systems, to data analysis applications. In the following, we discuss three different applications related to military and intelligence area.

5. Visual Web Searching and Information Retrieval Systems

Archives and databases of defense agencies hold terabytes of data and instant and comfortable access to relevant information is crucial for many of its users. The basic problems of collecting, storing and indexing data has the technology solved yet, but the open issue still remains searching of the data and especially browsing through search results. Also practical Web searching experience says that 70% of all searches are failures (Funke, 2000). The common Web user interface is perfect for displaying just very small data sets, but it becomes difficult to browse hundreds and sometimes even tens of search results. The user interface completely fails in situations when systems return thousands hits or more.

The 3D Web software creates a high-density user interface, capable of displaying hundreds of data points at single computer screen. The interactivity of the visual navigation makes processing of search results faster and easier and provides user also with additional functionality (editing, deleting, selecting, saving result sets…). It is possible to encode dimensions of the information (data fields, columns; i.e. data source, document type, classifications, publication year, author, number of links…) into its 3D graphical attributes (position within the space, color, size, shape...) and deploy human's perceptual abilities to differentiate results and identify relevant information faster.

We used Miner3D SITE to develop a visual Web searching service (http://miner3D.com/search/) as an example of advanced information retrieval systems. The application allows visitor to browse search results returned from a prime Web search engine of his choice (Google, AltaVista, NorthernLight, MSN Search, Google Newsgroups…) within an artificial 3D information space. Results are visualized as graphic objects positioned by its relevancy, colorized according to a document's domain (blue hits = *.com, red hits = *.net, green hits = *.edu) and textured by most important text information (title, domain, text, size). At selected search engines we can use also height of the data objects to carry additional information (Google returns also document sizes, NorthernLight returns percentage of relevancy).

The 3D technology enables to design both completely artificial information spaces without any counterpart in the real world, as well as simplified virtual copies of physical archive rooms or libraries. The design of the information space can be fully customized to real applications. In conjunction with developer's access to full set of data dimensions available in database, it is possible to develop series of information retrieval applications reflecting specific needs of users.

6. Distributed Real-time Monitoring and Control Systems

There are many monitoring and control applications in use over the world and most of them use advanced graphical user interfaces. Usually the applications are standalone software programs and often require using a special hardware, or a complicated installation and setup process, or a lot of training.

A powerful real-time Web 3D environment providing the same functionality through common Web browser would be able to increase tremendously the number of users to virtually unlimited. Various visualization models with different access levels allow designing of complex, hierarchically structured operational, training or educational systems. The 3D Web environment provides an application also with nearly perfect communication infrastructure: exchanging situation data facts, real-time data update feed, issuing of commands and signals can go visually, by email, voice, video or any other Internet compatible channel. Low requirements for hardware with no need for installation and setup (the first visit at an URL performs the installation automatically and transparently) dramatically changes the economy of such systems by reducing assets and operating costs.

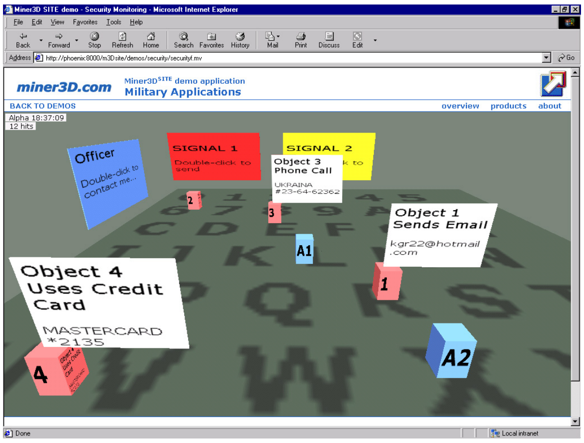

Miner3D SITE demo - Security Monitoring - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Discuss Edit
Address http://phoenix:8000/m3Dsite/demos/security/securityf.mv
Go
miner3D.com
Miner3D^SITE demo application
Military Applications
BACK TO DEMOS
overview products about
Alpha 18:37:09
12 hits
Officer
Double-click to contact me...
SIGNAL 1
Double-click to send
SIGNAL 2
Object 3
Phone Call
UKRAINA
#23-64-62362
Object 1
Sends Email
kgr22@hotmail.com
Object 4
Uses Credit Card
MASTERCARD
*2135
A1
A2
1
2
3
4
Done
Local intranet

Figure 1: Screen shot of a real-time control and monitoring demo 3D Web application

For demo purposes, we created a very simple example of a distributed real-time monitoring and control application (http://miner3D.com/m3Dsite/demos/). A team of agents is monitoring members of a terrorist group. Real-time data feed of updates of objects' positions within a watched area and activity data of all objects is concentrated into single visualization. An operator, analyst or commander can watch the concentrated information, which is provided in a well-readable form. They all can maintain communication, take better decisions faster and can react immediately using the same environment.

7. Financial Transactions Data Analysis Systems

Last months show a growing interest in preventive and analytic operations as an important part of our defense efforts. Analyzing of financial and property transactions, bank operations and stock trades can reveal a potential criminal or terrorist activity.

The problem of such analyses is poor readability of accounting and bookkeeping records. This, combined with overload of raw data, prevents from revealing suspicious operations and many of them remain hidden. Our new visual method of analysis of accounting records materializes the abstract nature of financial transactions and shows also their historical or real-time dynamics.

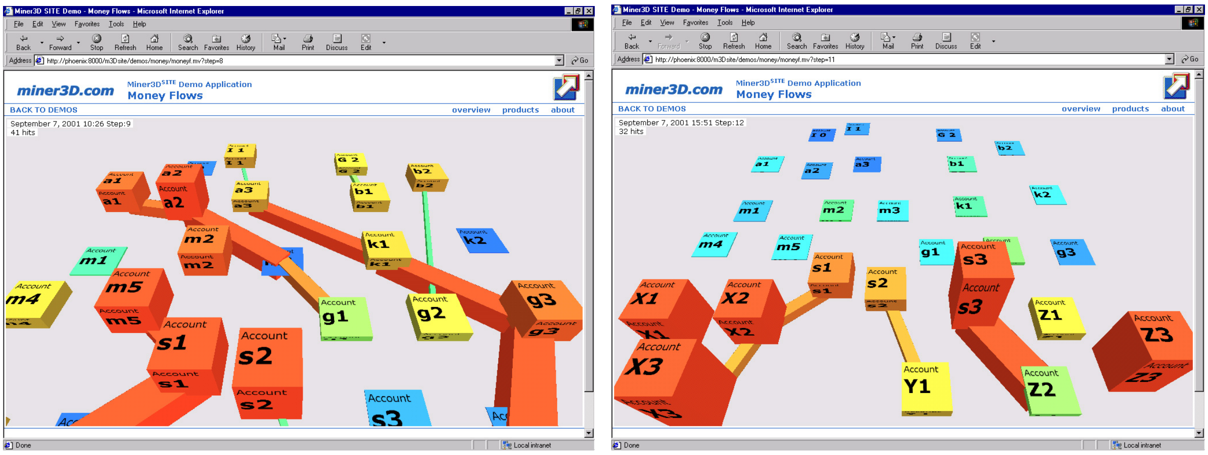

Miner3D SITE Demo - Money Flows - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Discuss Edit
Address http://phoenix:8000/m3Dsite/demos/money/moneyf.mv?step=8
miner3D.com
Miner3D^{SITE} Demo Application
Money Flows
BACK TO DEMOS
overview products about
September 7, 2001 10:26 Step:9
41 hits
Account a1
Account a2
Account a3
Account b1
Account b2
Account k1
Account k2
Account m1
Account m2
Account m4
Account m5
Account g1
Account g2
Account g3
Account s1
Account s2
Account s3
Done
Local intranet
Address http://phoenix:8000/m3Dsite/demos/money/moneyf.mv?step=11
September 7, 2001 15:51 Step:12
32 hits
Account X1
Account X2
Account X3
Account Y1
Account Z1
Account Z2
Account Z3
Done
Local intranet

Figure 2: Screen shots of a financial transactions data analysis demo 3D Web application

In demo application (http://miner3D.com/m3Dsite/demos/), we use fictive data of series of bank transfers between nearly 30 accounts of companies and organizations belonging to several financial groups. The accounts are represented by bars and are positioned to form visual clusters demonstrating the groups. Heights of accounts reflect available sums, while transactions between accounts are visualized as pipes transferring money from one account to another. In several rounds of the whole transaction, cash from 2-3 accounts is transferred to 3-4 target accounts through series of smaller fictive operations used just to hide the real intention of the complex transaction. If the transaction is recorded using traditional accounting methods, it counts hundreds of lines and it takes hours even to experienced analyst to decode it. Our method dramatically reduces time to make a qualified decision and provides analyst also with additional clues and indices.

8. Conclusion

We tried to demonstrate the power and universality of the upcoming Web 3D applications and show its potential for future military and intelligence applications. The 3D Web technology brings new advanced features while maintaining high accessibility, ease-of-use, distribution and installation, as general benefits of Web technologies.

Our research and development will continue in improving of quality of user interface and its conventions, navigation, definition of the communication protocol, attributes of the graphics space (shapes, fonts, textures, images), as well as in the application area by identifying new potential application areas.

References

Miner3D.com (2001), DIMENSION 5, Ltd., http://miner3D.com

Funke, C. (2000) "Information Searching", University Freiburg, Germany

Howe, C. D. (2001) "The End of the Web as We Know It", Forrester Research, USA

A Knowledge-Based Tool for Planning of Military Operations: the Coalition Perspective

Larry Ground
Alexander Kott
Ray Budd

BBN Technologies
5 PPG Place, Ste 310
Pittsburgh, PA 15222
{lground, akott, rbudd}@bbn.com

Abstract. Use of knowledge-based planning tools can help alleviate the challenges of planning a military operation in a coalition environment. We explore these challenges and potential contributions of knowledge-based tools using as an example the CADET system, a knowledge-based tool capable of producing automatically (or with human guidance) battle plans with realistic degree of detail and complexity. In ongoing experiments, it compared favorably with human planners. Interleaved planning, scheduling, routing, attrition and consumption processes comprise the computational approach of this tool. From the coalition operations perspective, such tools offer an important aid in rapid synchronization of assets and actions of heterogeneous assets belonging to multiple organizations, potentially with distinct doctrine and rules of engagement. In this paper, we discuss the functionality of the tool, provide a brief overview of the technical approach and experimental results, and outline the potential value of such tools for coalition operations.

1. Overview

Influential voices in the US military community (Wass de Czege and Biever, 2001) argue for significant computerization of the military planning process and for "...fast new planning processes that establish a new division of labor between man and machine. Staffs will rely heavily upon software to complete the straightforward calculations. Decision aids will quickly offer suggestions and test alternative courses of actions." Although the reasons for introducing such a computerization in the military planning processes are compelling enough even in the context of a single-nation military, many of the same reasons become even more pronounced in a coalition environment:

- The process of planning a military operation remains relatively cumbersome, inflexible and slow even when conducted by a planning staff that trained together extensively in order to achieve common understanding of the collaborative procedures, approaches and ontology. In a coalition context, the planning staff rarely has the benefits of extensive joint training, and comes into the process with significantly different sets of procedures, terminology, and doctrines (Riscassi, 1993).
- The planning process frequently involves significant disagreements on estimation of outcomes, attrition, consumption of supplies, and enemy reactions. Much of these disagreements arise from differences in mental models and underlying assumptions of the process participants. Such differences are further exacerbated in planning performed by a coalition staff (Elron et. al., 1999).
- There is a fundamental complexity of synchronization and effective utilization of multiple heterogeneous assets performing numerous, inter-dependent, heterogeneous tasks. This complexity, heterogeneity and the need for careful coordination and synchronization inevitably grow in a coalition environment, particularly for the ground component.

We argue that using an effective decision aid can, in part, alleviate these challenges. As an example, consider CADET, a tool for producing automatically (or with human guidance) Army battle plans with realistic degree of detail and complexity. In ongoing experiments, it compared favorably with human planners.

Views expressed in this paper are those of the authors and do not necessarily reflect those of the U. S. Army or any agency of the U.S. government.

In brief, the human planner defines the key goals for a tactical course of action (COA), and CADET expands them into a detailed plan/schedule of the operation. CADET expands friendly tasks, determines the necessary supporting relations, allocates / schedules tasks to friendly assets, takes into account dependencies between tasks and availability of assets, predicts enemy actions and reactions, devises friendly counter-actions, estimates paths of movements, timing requirements, attrition and risk. CADET is a generic engine, not specific to any type of assets or tasks. Although currently it is fitted with a US Army-specific task model, it can be readily augmented with models for other forces and nations, a clear requirement for coalition warfare.

Recently, there were several efforts to utilize the planning capability introduced by CADET. For example, US Army Battle Command Battle Lab-Leavenworth (BCBL-L) chose CADET as the centerpiece for its Integrated COA Critiquing and Evaluation System (ICCES) program to provide task expansion for maneuver COAs created with sketching tools and plan developers.

DARPA applied CADET in its Command Post of the Future (CPoF) program as a tool to provide a maneuver course of action. Under the umbrella of the CPoF program, CADET was integrated with the FOX GA system (Hayes and Schlabach, 1998) to provide a more detailed planner, coupled with COA generation capability. Battle Command Battle Lab-Huachuca (BCBL-H) integrated CADET with All Source Analysis System-Light (ASAS-L) to provide a planner for intelligence assets and to wargame enemy COAs against friendly COAs.

The development of Course of Action Display and Evaluation Tool (CADET) began in 1996, at the Carnegie Group, Inc. under the funding available under the Small Business Innovative Research (SBIR) program. With numerous other efforts addressing various aspects of the military decision-making process (MDMP), we sought to concentrate our efforts on the COA analysis phase of the MDMP.

In a setting such as a US Army divisional planning cell, the detailed analysis of a tactical course of action involves a staff of 3-4 persons with in-depth knowledge of both friendly and enemy tactics. Working as a team, they ascertain the feasibility of the COA, to assess its likelihood of success against a particular enemy COA, and to identify the points of the COA requiring synchronized action for participants. The resulting analysis is usually recorded in a matrix format, with time periods for the columns and functional alignment, such as the Battlefield Operating Systems (BOS), for the rows (Field Manual 101-5). Comparable, although not necessarily identical elements exist in decision-making processes of other nations' military establishments, and will be undoubtedly found, formally or informally, in any coalition decision-making.

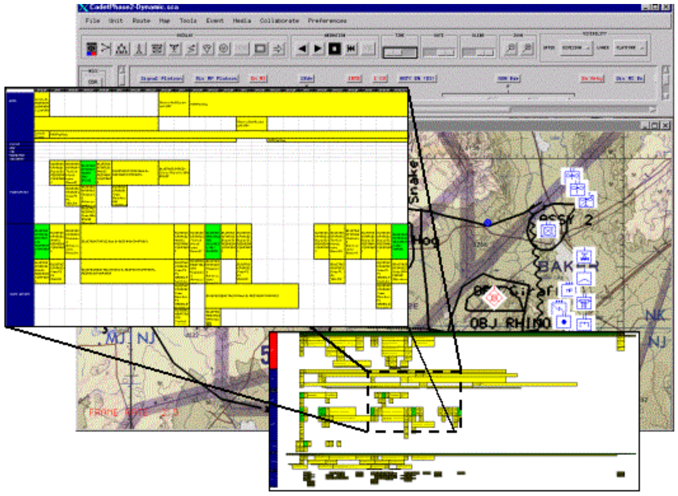


Figure 1 CADET takes a sketch and statement as an input, and produces detailed schedules of hundreds of tasks, usage of resources, risks and losses, actions of the enemy, and routing.

2. Challenges and Capabilities

A planning tool for coalition warfare must provide numerous capabilities to address a number of key challenges. Such capabilities fall into several broad categories:

- Modeling of assets and tasks
- Adversarial environment
- Coordinating team efforts
- Autonomous action

In this section, we explore some examples of such capabilities and their possible relations to coalition operations, from a functional, domain-oriented perspective.

Modeling of assets and tasks

Coalitions bring together military assets with different capabilities and employment doctrines. All too often, a coalition includes members whose assets, capabilities and tactics are not particularly familiar to other members. Thus, any decision aid for coalition planning must allow flexible, inexpensive, and rapid modeling of assets and associated tasks.

Let us consider the evolution of modeling the air assets in CADET as an example. Initially, we started with a very simple modeling that calculated deployment/re-deployment times and time-on-station, but with flat rates applied to resource consumption and timing considerations.

As the modeling evolved, we captured the variations caused by a variety of different aspects of the employment cycle. For example, working with the Battle Command Battle Lab-Huachuca, we performed a detailed breakdown of the sub-tasks involved in readying, launching and positioning a UAV. The possibility of concurrent tasks was factored in where the UAV could be routed to collect intelligence along the ingress/egress route.

The impact of the UAV use on the ground maneuver plan was greater than originally expected. Subject matter experts (SME) had predicted the ground commander would use the UAVs primarily to verify *known* or *suspected* information. Further analysis revealed a prejudice toward UAVs by the older generation based on experience with weather-constrained Army aviation and a tendency to focus on operations within their immediate control. Younger officers, however, employed UAVs as a primary source for intelligence, integrating them fully into the intelligence collection plan.

CADET added a new dimension to the modeling of UAV by showing the demands of continuous coverage. Users had generally planned individual missions or multiple missions. Few of them had considered the full implications of putting continuous coverage on a target. Army attack helicopters address this by using one of three modes: attack by platoon, attack by company or simultaneous attack.

A unit with a limited number of UAVs must factor in travel and recovery time for the cycling UAVs to determine if continuous coverage is feasible. Users were generally discounting the cost of the recovery time (for refueling and preventive maintenance) when calculating the amount of time the UAVs were effectively available for on-site observation.

As this example illustrates, an approach to modeling of assets must take into account at least the following considerations: (a) it must provide for rapid, inexpensive insertion of an initial, coarse but serviceable model; (b) allow for gradual increase in the model's fidelity, with incremental modifications even in a field environment, and (c) recognize and accommodate significant differences between organizations, as well as the ongoing evolution, in approaches to asset's employment.

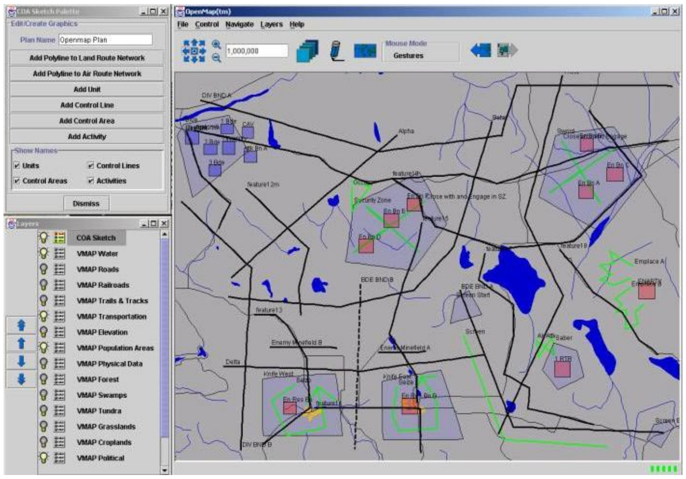


Figure 2 One of the COA-editing tools that have been used as data-entry interfaces to CADET and an example of a sketch produced with the tool.

Adversarial environment

Assumptions and expectations regarding the enemy are particularly challenging in a coalition, where the doctrine of staff officers from multiple nations can differ significantly and the political and strategic aims of the participating nations may be at odds (Riscassi 1993).

Manual wargaming typically depicts the enemy in a situation template, literally a standard tactical formation adapted to a specific piece of terrain in a given situation. Modeling the enemy over time, then, is a matter of taking the standard formations and moving them along the avenues of approach toward the friendly force.

In practice, there are several aspects in considering how the enemy affects friendly actions. In particular, every action taken by either combatant is likely to cause a reaction by the opponent and it might be possible to negate the reaction with the appropriate counteraction. Further, a quick, reliable Conflict Resolution Model (CRM) is needed to determine the effects of each engagement on the combatants.

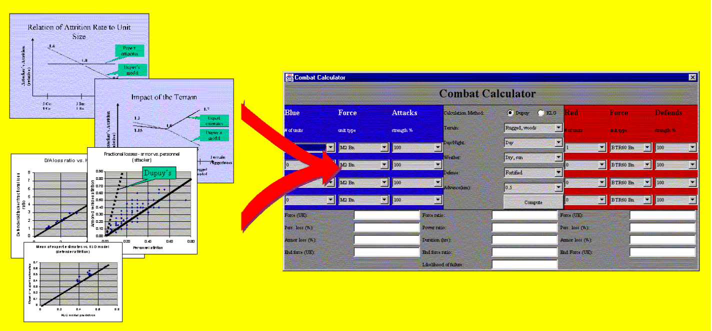
Action/reaction/counter-action

Every action possible by either friendly or enemy units warrants examination for potential reactions. This is augmented with further analysis to determine if there exists a counter-action that can be used to minimize the impact of the reaction or negate its effects completely.

For example, whenever artillery is fired, the opposing force will attempt to locate the firing piece and fire counter-battery fire. The firing unit must either be prepared to relocate or expect to receive incoming fire. The general effect is to reduce harassing and interdicting fires whenever a credible counter-battery threat is present. The potential counter-action is for the firing unit that fired first to conduct counter-battery operations of its own. In fact, US forces have sometimes fired in hopes of drawing the enemy into counter-battery fire for the explicit purpose of destroying the enemy artillery through counter-battery fire. In a coalition environment, a planning tool must allow for multiple and readily adjustable models of such action-reaction-counteraction, to reflect diverse perspectives and expectations of the coalition members.

Conflict Resolution Modeling (CRM)

Although the approach of Dupuy (1990) offers many advantages for application in a system like CADET, the modest demands on the required data being one of them, we found that it produced results that were not in concert with those expected by the users. Having involved expert panels of military officers, both active duty and retired, we modified the equations and coefficients provided in (Dupuy, 1990) to match the expertise and experience of current practitioners (Kott, Ground and Langston, 1999). In a coalition planning process, it may be desirable to be able to either select from a library of multiple models, or to modify rapidly an existing one in a manner that takes into account the perspectives and experiences of the coalition members (Elron et. al., 1999).


Relation of Attrition Rate to Unit Size
Impact of the Terrain
Fractional losses - armor vs. personnel (attacker)
Dupuy's

Combat Calculator

Blue	Force	Attacks
# of units	unit type	strength %
	M2 Bn	100
0	M2 Bn	100
	M2 Bn	100
0	M2 Bn	100

Calculation Method: ● Dupuy ○ KLG
Terrain: Rugged, woods
Day/Night: Day
Weather: Dry, sun
Defense: Fortified
Advance(km): 0.5
Compute

Red	Force	Defends
# of units	unit type	strength %
1	BTR60 Bn	100
0	BTR60 Bn	100
0	BTR60 Bn	100
0	BTR60 Bn	100

Force (UE):
Pers. loss (%):
Armor loss (%):
End force (UE):
Force ratio:
Power ratio:
Duration (hrs):
End force ratio:
Likelihood of failure:
Force (UE):
Pers. loss (%):
Armor loss (%):
End Force (UE):

Figure 3 CADET estimates personnel and weapon systems attrition mimicking the evaluations performed by Army's experts.

Coordinating team efforts

Coordinating timing and movement

Coalition warfare exacerbates the need for careful, thoughtful coordination of temporal and spatial aspects of all tasks within an operation. Field Manual 3-0, the US Army's keystone manual for operations, states "Detailed war-gaming, planning and rehearsals help develop a common understanding of the operation plan and control measures (Field Manual 3-0)." CADET's users can input temporal relationships for high-level activities for a plan. Subject matter expert and user feedback provided us with important information concerning the way a commander conceives the temporal relations between activities. For example, does an attack in an area start when the unit starts moving to the specified area, when the unit attacks the targeted unit, or when the unit enters the specified area?

In CADET, this problem is solved by identifying what we call anchor points for each activity. When the user says that two specified activities should start at the same time, he or she has a specific idea about which derived activities they want the units to be starting at the same time. Users were typically less concerned with the time at which a unit starts moving and more interested in each unit first makes contact with the enemy. For example, when performing a *Seize*, the start anchor point is the first movement in the area being seized. When performing a *Close-with-and-engage*, the start anchor point is the first attack on any target unit. In coalition operations, however, it is likely that officers from different doctrinal backgrounds will have different notions about such anchor points. This is yet another aspect of knowledge engineering in systems like CADET that requires a mechanism for rapid, in-field modifications.

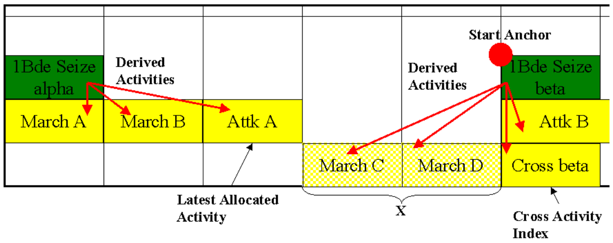


Figure 4 CADET modifies the start of the Seize activity, to occur after all necessary derived activities end.

Coordinating supporting relationships

The common errors encountered in manual COA analysis include failure to fully utilize resources, committing resources to provide support when they are not within range, and over-committing resources.

Clearly, these errors would be even more likely to occur in a COA analysis process performed by a coalition staff. CADET's planning and scheduling algorithm ensures resources are allocated within constraints and are not over-committed. In those cases when the algorithm is unable to find a solution without an over-commitment of resources, CADET identifies the affected activity as questionable (e.g., Fig. 5), but continues the planning process. This allows the user to accept or correct the over-commitment of resources when a more complete solution is available for review and decision-making.

CADET tracks the utilization of resources to allow users to know where resources are not being fully exploited, a capability that can be used to look for places where resources could be applied elsewhere.

CADET looks at the effective range of supporting resources, such as logistics facilities, to determine if they are close enough to achieve the mission. For instance, CADET models the actual movement of support elements between the field trains and combat units. As the combat elements move forward in the offense, and the distances and the time required to perform re-supply increases as well. When it becomes too great to support the planned level of tactical operations, CADET cues the planner to reposition the field trains forward to a closer location. If the trains cannot be repositioned in a timely manner, CADET identifies the restrictions imposed on the combat unit by the reduced level of support. By taking care of such details, CADET can help the coalition staff avoid the typical mistakes of resource management in COA analysis.

Operating in three dimensions

In practice, human planners tend to focus exclusively on the close fight, without due consideration to the full depth of the battlespace. For example, leaders who lack experience with US Army attack helicopters tend to discount their value or leave them out of the equation completely.

A deep attack will normally cause serious attrition for the enemy but carries with it the risk of friendly losses. If Army attack helicopters are lost behind enemy lines, it necessitates a combat search and rescue (CSAR) mission. On the other hand, a deep attack could reduce the enemy strength to the point where the enemy is forced to call off the attack. Whenever assets are available, a deep attack should be considered. The coalition staff officers can take advantage of CADET's ability to analyze air attacks to build in COAs with air assets, where air and ground assets may belong to different coalition members.

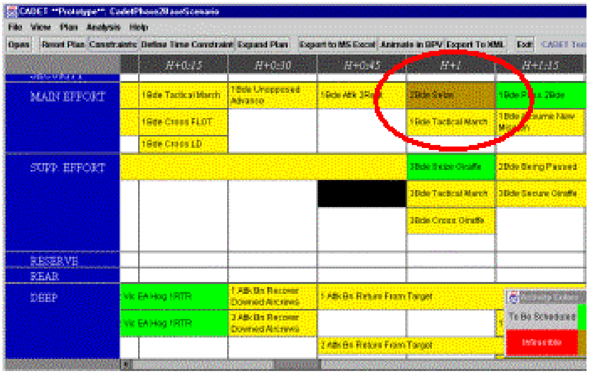


Figure 5 CADET works interactively with the user to identify questionable activities.

Autonomous action

In the context of coalition warfare, even more so than in single-nation warfare, guidance from the commander should often come in the form of his intent or the desired results (Keithly and Ferris, 1999).

Modeling tasks based on intent

The bypass criterion in CADET provides the ability for units to disengage when the opposing force has been attrited to a certain level. However, it does not address the more general situation encountered where actions are initiated with a specific intent in mind. For instance, in *economy of force* operations, the supporting attack will generally not be able to destroy or even to defeat the enemy. Rather, the intent of the supporting attack is to ensure the success of the main effort, regardless of the extent to which the supporting effort is able to defeat the enemy.

Artillery fire commonly has an associated intent. Artillery will be used to suppress, to mask, to defeat, or to destroy. By extending the task set to include the intent, the applicability of the tasks to specific situations was greatly enhanced.

Modeling for deliberate attack is an excellent example of intent and its effect on resource consumption. In CADET, the task is modeled to allow the projection of attrition for attacks that are not attempting to completely remove the enemy (i.e. Attack to Attrit). The effect is a change to attack duration, and ultimately a modification to total defender and attacker attrition. For a planner, the need to hold the friendly strength at or above a certain threshold might be key to the analysis of a particular COA.

Derived actions for subordinates based on higher level tasks

A coalition operation consists of a large number of disparate, unique sub-tasks working to achieve a common goal. To properly model the task requires modeling a variable number of sub-tasks. The timing and interaction of the sub-tasks determines the success or failure of the task. Of particular interest is the assignment of tasks and routes to units that are not fully identified by the user.

A counter-attack is a good example. The commander will attempt to commit the counter-attack force at the time necessary to reverse the trend of the defense. The problem is that the exact speed and route of the attacking force can generally not be predicted in advance. The counter-attack force will be most effective if it is able to strike a flank.

CADET automatically calculates the route and timing for the counter-attack force's movement. In a deliberate planning mode, this allows time to perform route reconnaissance. In a real-time execution-replanning cycle, the ability to rapidly calculate routes and related timing would facilitate identification of the decision point for commitment.

Movement to contact, another good example, represents a significantly harder challenge. The main body deploys a small security force to establish the initial contact, followed closely by a larger security force. The intent is to make the initial contact with the smallest possible force that can develop the situation. The unit making the initial contact attempts to determine the size, composition and intentions of the enemy force. The unit commander must make the initial determination whether to bypass the enemy, avoid contact (if possible), engage directly, or assist the effort of the main body.

CADET uses rules to determine the actions of the security elements. Each individual element follows the rules to decide its actions on contact. These actions ripple through the team. For instance, if the lead security element encounters a particularly strong enemy force that meets the criteria for an attack by the main body, the lead security element will:

- Engage the enemy in direct fire.
- Determine the best route and point for employment for the following security body.
- Determine the possible routes for the main body attack for consideration by the commander.
- Secure the flank opposite the following security body.

The ability to derive the tasks of the subordinate elements as a result of rules-based task expansion and situational analysis is a critical aspect of CADET's planning function. In a coalition environment, this capability helps provide an objective basis for systematically identifying and allocating tasks to assets of multiple members.

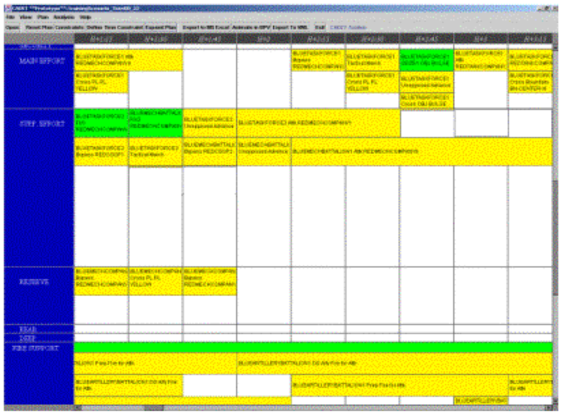


Figure 6 A typical plan/schedule of a brigade-sized offensive operation may include hundreds of significant tasks. A fragment of such a plan is shown here.

3. Technical Approach

Let us consider briefly how CADET addresses some of the technical challenges implicit in the capabilities discussed above.

The integration of planning and scheduling is achieved via an algorithm for tightly interleaved incremental planning and scheduling. The HTN-like planning step produces an incremental group of tasks by applying domain-specific "expansion" rules to those activities in the current state of the plan that require hierarchical decomposition. The scheduling step performs temporal constraint propagation (both lateral and vertical within the hierarchy) and schedules the newly added activities to the available resources and time periods (Kott, Ground and Budd, 2002).

The same interleaving mechanism is also used to integrate incremental steps of routing, attrition and consumption estimate. For estimates of attrition, we developed a special version of the Dupuy algorithm (Kott, Ground and Langston, 1999) that was calibrated with respect to estimates of military professionals, US Army officers. This attrition calculation can be replaced with other methods, when employed in a coalition environment.

The adversarial aspects of the planning-scheduling problem are addressed via the same incremental decomposition mechanism. In particular, the tool automatically infers (using its knowledge base and using the same expansion technique used for HTN planning) possible reactions and counteractions, and provides for resources and timing necessary to incorporate them into the overall plan. In effect, this follows the military action/reaction/counter-action analysis.

In spite of significant functionality, the algorithms of CADET provide high performance. On a modern but not exceptionally fast laptop, a typical run – generation of a complete detailed plan from a high-level COA – takes about 20 seconds. With the coalition planning process taking longer than single-nation planning, which is already considered too slow, the ability to perform multiple, rapid iterations of computerized planning is very important (Riscassi, 1993).

The knowledge base of CADET is structured for simplicity and low cost. In practice, the most expensive (in terms of development and maintenance costs) part of the KB is the rules responsible for expansion (decomposition) of activities. CADET includes a module for KB maintenance that allows a non-programmer to add new units of knowledge or overwrite the old ones. This is critical in a coalition environment, where the knowledge base must be rapidly extended in field conditions, to accommodate assets and rules associated with new coalition members.

From the perspective of integration with other systems, the rigorous separation – both architectural and conceptual - of problem solving components from user interaction mechanisms, allows for integration with a variety of user-interface paradigms and systems. The extensive use of XML enables simple, inexpensive integration with a variety of heterogeneous systems, a significant advantage in environments where members of a coalition bring with them a variety of systems (Thomas, 2000).

4. Experimental Comparisons – CADET vs. Manual Approaches

A recent experiment, one of several series (Rasch, Kott and Forbus, 2002; Kott, Ground and Budd, 2002), involved five different scenarios and nine judges (active duty officers of US military, mainly of colonel and lieutenant colonel ranks). The five scenarios were obtained from several exercises conducted by US Army. The scenarios were all brigade-sized and offensive, but still differed significantly in terrain, mix of friendly forces, nature of opposing forces, and scheme of maneuver. For each scenario/COA we were able to locate the COA sketches assigned to each planning staff, and the synchronization matrices produced by each planning staff. The participants, experienced observers of many planning exercises, estimated that these typically are performed by a team of 4-5 officers, over the period of 3-4 hours, amounting to a total of about 16 person-hours per planning product.

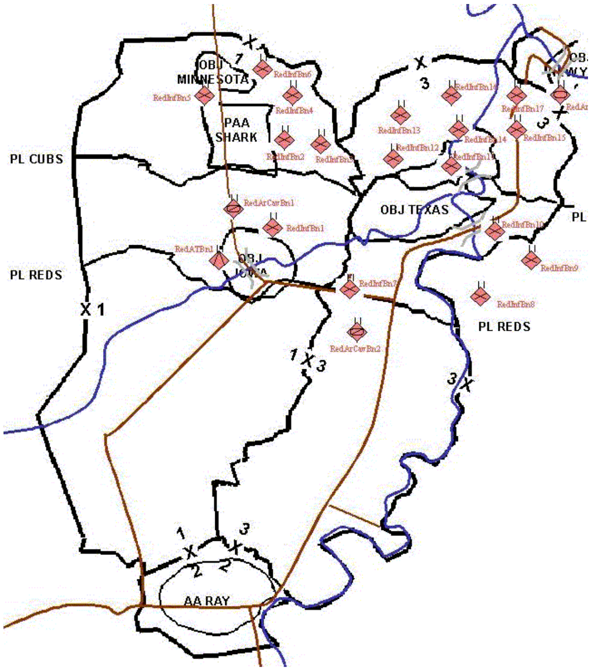


Figure 7 This sketch was used in one of the experiments performed with CADET.

Using the same scenarios and COAs, we used the CADET tool to generate a detailed plan and to express it in the form of a synchronization matrices. The matrices were then reviewed and edited by a surrogate user, a retired US Army officer. The editing was rather light – in all cases it involved changing or deleting no more than 2-3% of entries on the matrix. This reflected the fact that CADET is not expected to be used purely automatically, but rather in collaboration with a human decision-maker. The time to generate these products involved less than 2 minutes of CADET execution, and about 20 minutes of review and post-editing, for a total of about 0.4 person-hours per product. The resulting matrices were transferred to the Excel spreadsheet and given the same visual style at that of human-generated sets.

The products of both the CADET system and of human staff were organized into packages and submitted to the nine judges. Each package consisted of a sketch, statement, synchronization matrix and a questionnaire with grading instructions. The judges were not told whether any of the planning products were produced by the traditional manual process or with the use of any computerized aids. To avoid evaluation biases, assignments of packages to judges were fully randomized. Each judge was asked to evaluate four packages. Each judge was asked to review a package and grade the products contained in the package.

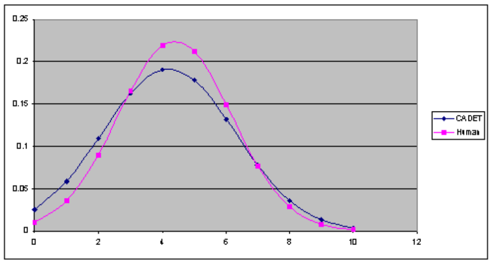


Figure 8 The results of experiments approximated as normal distributions: the judges were asked to grade the products of CADET and manual process on a scale of 0 to 10.

The results demonstrate very little difference between CADET's and human performance. In particular, based on the mean of grades, CADET lost in two of the five scenarios, won in two, and one was an exact draw. Taking the mean of grades for all five scenarios, CADET earned 4.2, and humans earned 4.4, with the standard deviation of about 2.0, a very insignificant difference.

The basic conclusion is clear: the judges gave CADET-produced products (which took typically about 20 minutes to produce) essentially the same level of grades as to the human-produced products (which took on the order of 16 person-hours to produce).

5. The Coalition Perspective: Conclusions and Future Work

A tool like CADET is applicable to a planning process where the planners are tasked with rapid synchronization of assets and actions of heterogeneous assets belonging to multiple organizations from multiple nations and services, potentially with distinct doctrines. The assets that enter CADET's problem solving process do not need to belong to one nation or service. Instead, each asset, e.g., a unit of force, could have its own doctrine, capabilities and rules of engagement (ROE).

The version of the HTN planning paradigm employed by CADET allows that a composite task is decomposed into lower-level subtasks by multiple different methods where the appropriate one is selected depending on which coalition resource would be applicable or assigned to the task. The object-oriented representation of tasks allows economical representation of nation-specific doctrinal variations applicable to the planning and execution of the task.

The integrated planning-scheduling process allows the tool to pick and choose the best coalition force, based on applicability, availability and ROE even if the assets belong to different nations. The mechanisms for flexible human intervention provide opportunities for adjusting system's choices and guiding a system in selecting proper matches between multi-force tasks and resources.

Officers belonging to different nations will need to modify or augment the knowledge base in accordance with their nation's specific doctrine. To this end, the CADET suite includes a mechanism that allows an end-user, a non-programmer, to enter definitions and rules of tasks and store them in a user-specific segment of the knowledge base. Officers can define the knowledge in the field, in real time, even while the coalition is forming and the members are defining the constraints and rules of their participation.

Coalition operations also highlight the need for a tool like CADET to allow collaborative, distributed work. Staff officers will function over geographically dispersed areas, using their adapted version of CADET on a highly portable personal computing device. Each officer on the staff uses his copy of CADET to perform a slice of the overall planning task by (a) considering the partial plans that arrive electronically from other collaborating officers; (b) making reasonable assumptions when actual partial plans are not available; (c) issuing its own partial plans to other officers and highlighting inconsistencies, if any. Although currently CADET functions as a single-user tool, we are considering plans to extend the tool for multi-user, coalition-staff operations.

At this time, CADET shows promise of reaching the state where a military decision-maker, a commander or a staff planner, uses it routinely as part of an integrated suite of tools to perform planning of tactical operations, to issue orders, and to monitor and modify the plans as the operation is executed and the situation evolves. It is not too far-fetched to suggest that such a tool may provide an 80% solution, under most situations, in a fraction of the time required for comparable manual staff planning products.

However, CADET's current state of capabilities also points toward the key gaps that must be overcome to realize the full potential of such tools in coalition warfare:

The coalition planning process is particularly demanding on effective human-machine interfaces that can be used in spite of staff members' differences in training and procedures. Such interfaces remain elusive, especially for complex, multi-dimensional information such as plans and execution of military operations, in high-tempo, high-stress, physically challenging environments. Today's common paradigms – map-based visualizations of spatial information and synchronization matrix for temporal visualization – are not necessarily the best approach, and different methods ought to be explored.

Presentation of the CADET's products requires qualitatively different user interfaces and visualization mechanisms. Our experiments suggests that users had difficulties comprehending the synchronization matrix generated by the computer tool, even though it was presented in a very conventional, familiar manner. Perhaps, the synchronization matrix functions well only as a mechanism for short-hand recording of one's own mental process and is not nearly as useful when used to present the results of someone else's, e.g., a computer tool's, reasoning process.

Ongoing work on CADET technology focuses on closing these critical gaps.

Acknowledgements

The work described in this paper was supported by funding from US Army CECOM (DAAB 07-96-C-D603 - CADET SBIR Phase I, DAAB 07-97-C-D313 – CADET SBIR Phase II and DAAB 07-99-C-K510 - CADET Enhancements); DARPA (DAAB07-99-C-K508 Command Post of the Future); and TRADOC BCBL-H (GS-35-0559J/DABT63-00-F-1247). TRADOC BCBL-L provided additional funding under DAAB 07-99-C-K510.

Dr. L. Rebbapragada of Army CECOM contributed significantly to the concept development along with technical monitoring of the project and valuable guidance. MAJ R. Rasch of the US Army BCBL-L provided important advice. John Langston, LtCol, US Army (ret.), contributed significantly to the understanding of the functional domain and the conceptualization of the Conflict Resolution Model.

References

Dupuy, T. N. (1990) "Attrition: Forecasting Battle Casualties and Equipment Losses in Modern War", Hero Books, Fairfax, Va.

Elron, E., Boas S., and Ben-Ari, E. (1999) “Why Don’t They Fight Each Other? Cultural Diversity and Operational Unity in Multinational Forces”, *Armed Forces & Society*, Fall 1999, Vol. 26, Issue 1.

Field Manual 3-0 (2001) "Operations", Headquarters, Department of the Army, Washington, D.C., June 2001

Field Manual 101-5 (1997) "Staff Organization and Operations", Headquarters, Department of the Army, Washington, D.C.

Hayes, C. C., and Schlabach, J. (1998) "FOX-GA: A Planning Support Tool for assisting Military Planners in a Dynamic and Uncertain Environment", in Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments, TR WS-98-02, R. Bergmann and A. Kott, eds. AAAI Press, Madison, Wisconsin.

Keithly, D. M. and Ferris, S. P. (1999) “Auftragstaktik, or Directive Control, in Joint and Combined Operations”, *Parameters: US Army War College*, Autumn 99, Vol 29, Issue 3.

Kott, A., Ground, L., and Langston, J. (1999) "Estimation of Battlefield Attrition in a Course Of Action Analysis Decision Support System", 67th Military Operations Research Society Symposium, West Point, 1999.

Kott, A., Ground, L., and Budd, R. (2002) "Toward Practical Knowledge-Based Tools for Battle Planning and Scheduling", forthcoming.

Rasch, R., Kott, A., and Forbus, K. (2002) "AI on the Battlefield: an Experimental Exploration", forthcoming.

Riscassi, R. W. (1993) “Principles for Coalition Warfare”, *Joint Force Quarterly*, Summer, 1993, pp. 58-71.

Thomas, T. L. (2000) “Kosovo and the Myth of Information Superiority”, *Parameters: US Army War College*, Spring 2000, Vol 30, Issue 1.

Wass de Czege, H. and Biever, J. D. (2001) "Six Compelling Ideas On the Road to a Future Army," Army Magazine, Vol.51, No.2, February 2001, pp. 43-48