Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

AN AGENT ARCHITECTURE FOR IMPLEMENTING COMMAND AND CONTROL IN MILITARY SIMULATIONS

Colin R. Mason

CORDA Ltd.
Apex Tower
7 High Street, New Malden
Surrey KT3 4LH, U.K.

James Moffat

Defence Science and Technology Laboratory (DSTL)
Farnborough
Hampshire
GU14 0LX, U.K.

ABSTRACT

In models of military operations it is important to include the Command and Control (C2) process in order to achieve a realistic simulation of a military force's behaviour and effectiveness. Inspired by ideas from complexity theory we have developed a representation of C2 based on a decentralised system of interacting intelligent "command agents". In this paper we describe the architecture of our command agents and how this captures the key C2 processes that exist in military headquarters, particularly the G2 and G3 processes of recognised picture compilation, decision making and planning. We describe a re-usable software framework that we have developed, within which we implement the command agents. The architecture and its software implementation enable us to produce command agents that can simulate C2 at any level in the military command hierarchy and in operations across the warfare spectrum - from high intensity combat to Operations Other Than War (OOTW).

1 INTRODUCTION

Our research is concerned with representing the effects of military Command and Control (C2), and determining how these effects can be incorporated successfully into constructive simulations of conflict. Including the C2 process in such models of military operations is essential if the simulation is to provide a realistic model of a military force's behaviour and effectiveness.

Inspired by ideas from complexity theory we have developed a representation of military C2 based on a decentralised system of interacting intelligent "command agents". The essential idea is that a number of interacting agents, behaving in accordance with *small* numbers of *simple* rules, can generate extremely complex emergent collective behaviour.

One of the aims of our research is to identify sets of simple rules and entity interactions that will give rise to emergent collective behaviour that resembles realistic military behaviour. A previous paper (Mason and Moffat 2000) described the progress we have made towards identifying these rules and interactions. This paper complements the previous one by looking in more detail at the design and implementation of our key component – the command agent.

In our approach to modelling the C2 process the command and control of a military operation is carried out by a number of agents interacting with one another within some kind of network. The network is usually, but not necessarily, a hierarchy of some kind, reflecting the organisational structure of the force. Each agent represents a military decision-making entity – a headquarters.

The research reported here is concerned with how to represent the agents in as generic a way as possible. The result of this work is OACIS (Object Architecture for C2 In Simulations) – an architecture for a generic, re-usable command agent.

The remaining sections of this paper describe the essential elements of OACIS. The paper is organised as follows. In Section 2 we highlight the design aims that guided the development of OACIS. Section 3 then describes the context within which we envisage an OACIS command agent operating. In Section 4 we present an overview of the OACIS agent architecture, identifying the main components of the agent. These components are then described in further detail in Section 5. We then discuss our approach to the software implementation of OACIS in Section 6, and describe a software testbed that provides proof-of-principle in Section 7. We summarise this work in Section 8.

2 OACIS DESIGN AIMS

We had three principal design aims for OACIS. Firstly, we wanted to define an agent architecture that captures the key C2 processes, and their interactions, which exist in a military headquarters. Of particular importance are the G2 and G3 processes of data fusion, recognised picture compilation, decision-making and planning. These represent the core processes within a headquarters concerned with evolving perceptions of the outside world from sensor and situation reports (data fusion), developing a mental model of what is going on (the recognised picture) and then deciding what to do next, and formulating a plan to achieve this, given the overall aims of the campaign and the current mission (decision-making and planning).

Secondly, we wanted an architecture that is generic so that it can be used to build agents that operate (a) at any level in the military command hierarchy, and (b) in both combat and OOTW simulation models, whilst at the same time recognising that the details of the C2 processes carried out by a given agent will depend on the actual role played by the agent.

Finally, we wanted to pitch the modelling approach at a level of detail appropriate to constructive simulations of the type used for high-level OA studies. Such studies are the principal target audience for the results of our research.

3 OPERATIONAL CONTEXT

The operational context of an OACIS command agent (CA) is shown below in Figure 1.

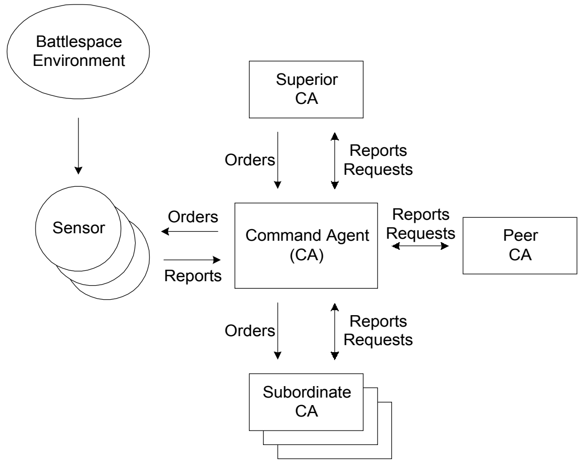

Figure 1: Context of an OACIS Command Agent

In military operations, a typical agent (the larger, central box in Figure 1) will have a superior agent (from which orders are received) and one or more subordinate agents (to which orders are sent). Reports and requests will be exchanged both up and down the command hierarchy, as well as laterally, with peer agents.

Reports received from other agents are one way by which a typical agent gains information about the outside world (the "Battlespace Environment" of Figure 1). An alternative way is via organic sensors, i.e., sensors controlled by the agent itself. A typical agent must be able to task such sensors (via orders) and receive the sensor output (via reports).

This description of a typical agent applies recursively to the superior, subordinate and peer agents. The overall C2 structure is therefore one of a network of agents interacting with one another via exchanges of orders, reports and requests.

4 COMMAND AGENT ARCHITECTURE

The high level structure of an OACIS command agent is shown below in Figure 2. The boxes represent the principal components of the agent. The directed lines joining the boxes show the main information flows between the components.

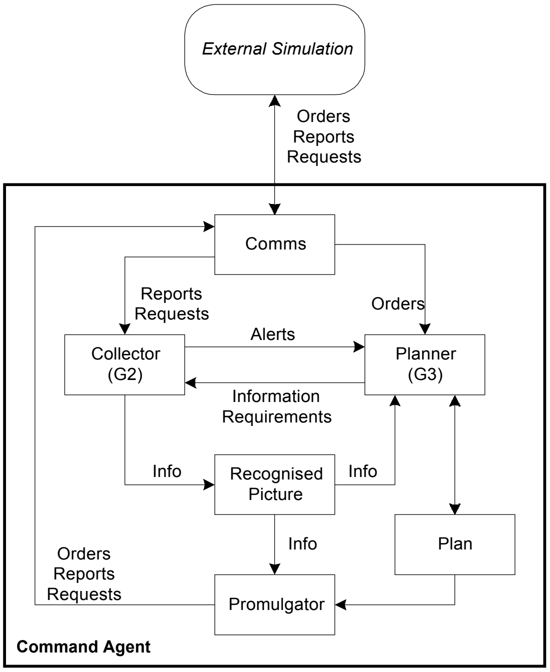

Figure 2: OACIS Command Agent Structure

The agent has six main components:

- Comms – provides communications facilities allowing the agent to exchange information (orders, reports, requests) with other agents.
- Collector – encapsulates the G2 processes, covering data collection, data fusion, construction and maintenance of the Recognised Picture and intelligence assessment.
- Planner - encapsulates the G3 processes, covering decision-making and planning.
- Promulgator - encapsulates administrative processes for managing the output of the agent, e.g., promulgation of the plan. It also handles the reporting cycle and promulgates reports/requests on demand.
- Recognised Picture (RP) – encapsulates the agent's knowledge of the outside world. The Planner does its planning on the basis of the RP.

- Plan - the Planner's output. It defines the tasks that are to be assigned to subordinate agents in order to achieve this agent's mission goals.

These components are described in more detail in the following sections.

5 COMMAND AGENT COMPONENTS

5.1 The Comms Component

The Comms component provides the agent with the ability to exchange various types of information with other agents.

We have developed a communications model based on a network representation. An agent subscribes to one or more communications "nets". For each net to which an agent subscribes, the agent constitutes a node in that communications net. The agent is able to communicate, within limits, with any other agent subscribing to the same net.

The sub-structure of the Comms component, and the mechanism whereby the Comms components of different agents are linked together, is illustrated below in Figure 3.

The main elements involved are:

- messages;
- the Comms component, comprising a comms manager, one or more comms transmitters (Tx) and one or more comms receivers (Rx);
- the comms medium.

We describe these in more detail below.

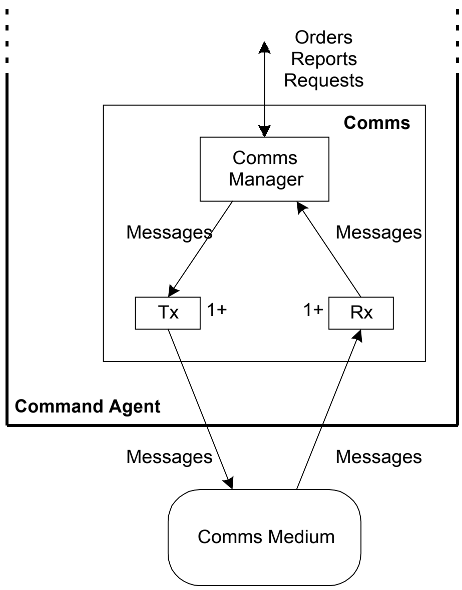

Figure 3: Components of the Comms Model

5.1.1 Messages

Messages are the fundamental entities that the communications process transports from one agent to another in order to achieve information exchange.

The agent that originates a message is responsible for creating the message entity and "injecting" the message into the comms medium via a comms transmitter. The comms medium then delivers the message to appropriate agents via those agents' comms receivers. All messages fall into one of three categories: Orders, Reports and Requests.

Orders task subordinate agents or give warning that tasking is about to change. Orders include various subtypes such as Operations Order, Warning Order and Fragmentary Order. Orders will be issued on an as-required basis, as the situation demands (they are situation driven).

Reports are of two subtypes - status reports and situation reports.

Status reports will report the state of own forces that are commanded by the agent issuing the status report. It is used to cover issues such as current deployments, state of personnel, state of equipment, vehicle casualties, ammunition usage, and so forth. Status reports will be issued:

- at a fixed frequency and time, e.g., every day at 2300 hrs; the frequency will be set by the issuer's superior commander;
- on an as-required basis, e.g., when something of significance happens; the criteria will be specified by the issuer's Standard Operating Procedure (SOP) and/or superior commander, e.g., a battlegroup's SOP might state that immediately after an action, or during the reorganisation phase of an operation, the battlegroup commander shall send a status report to his superior commander;
- on demand or request from another agent; this can happen at any time (see also Requests below).

Situation reports will report the state of everything external to the own forces that are commanded by the agent issuing the situation report. It is used to cover issues such as the state of the environment (terrain, weather, culture, obstacles), the state of the enemy (including contacts and sightings, warnings (e.g., of impending air attack)), and the state of adjacent (own) forces supporting the force commanded by the agent issuing the situation report. Situation reports will be issued:

- at a fixed frequency, e.g., every hour; the frequency will be set by the issuer's superior commander and will be appropriate to the level of command;
- on an as-required basis, e.g., when something of significance happens (e.g., contact with or sighting of the enemy; discovery of a minefield); the

criteria will be specified by the issuer's SOP and/or superior commander;
- on demand or request from another agent; this can happen at any time (see also Requests below).

Requests can be for anything that a commander might require in order to achieve his mission, e.g.,

- reinforcements;
- replenishments (manpower and materiel);
- support, e.g., indirect fire, combat engineers, casevac facilities;
- information, including status and situation reports.

Requests will be issued on an as-required basis, as the situation demands (they are situation driven).

5.1.2 Comms Managers

Each Comms component has a comms manager that is responsible for managing the component's associated equipments, namely, the comms transmitters and the comms receivers. Three types of comms manager are defined:

- the *terminal* comms manager - this type allows an agent to inject messages into, and extract messages from, the comms medium; this is representative of systems such as combat radio.
- the *relay* comms manager - this type does not permit an agent to access received messages; instead, it simply re-broadcasts received messages; this is representative of systems such as comms satellites.
- the *terminal relay* comms manager - this type is a combination of both the above.

5.1.3 Comms Transmitters

Comms transmitters provide one of the two interfaces between agents and the comms medium. A comms transmitter accepts messages from the Comms component's comms manager and injects these into the comms medium. The latter will then propagate the message.

5.1.4 Comms Medium

The comms medium is external to the agent and acts as the interface between comms transmitters and comms receivers of different agents. It accepts comms transmissions (in the form of messages) that have been injected into the comms medium by comms transmitters and propagates these transmissions to the comms receivers of other agents. The comms medium supports two types of signal propagation mechanism:

- Broadcast - this mechanism models free space electromagnetic propagation and simulates comms based on radio links. Agents using comms systems of this type are physically capable of receiving a transmission from another agent if, and only if, they are within the maximum effective transmission range of the transmitter.
- Narrowcast - this mechanism models propagation along direct connections and simulates comms based on landline links or low probability of intercept (LPI) links. Agents using comms systems of this type are physically capable of receiving a transmission from another agent if, and only if, they are directly connected (by a specified link) to the transmitter.

The modelling of message propagation is based on explicit signal and noise source levels, together with a propagation loss equation. When a message transmission is initiated the comms medium calculates the strength of the transmission signal that would be received at each comms receiver that is potentially capable of receiving the message. The comms medium then adds noise to the received signal of each comms receiver to represent the presence of, for example, ambient noise and comms band jammers.

The comms medium then sends a communications "reception opportunity" to the appropriate comms receivers. The reception opportunity consists of the message, the signal level at the receiver and the noise level at the receiver.

Here, the responsibilities of the comms medium end. It then becomes the responsibility of the comms receiver to determine whether or not the signal can be discriminated from the noise, and consequently whether or not the reception opportunity represents a "real" communications reception.

5.1.5 Comms Receivers

Comms receivers provide the second of the two interfaces between agents and the comms medium. A comms receiver accepts a reception opportunity, and its associated message, whenever one of these is sent to the receiver by the comms medium. The receiver determines whether or not the reception can actually be "heard" by thresholding the signal/noise ratio of the received transmission. If the signal/noise ratio is below threshold, the receiver ignores the reception opportunity and the message is discarded. If the signal/noise ratio is above threshold, the receiver accepts the reception opportunity and the latter is converted into a "real" communications reception event. If a "real" communications reception event occurs the receiver passes the message to the Comms component's comms manager.

5.1.6 Interaction With Other Components

The Comms component interacts with three other components of the agent, as follows:

- it passes reports and requests, received via the comms medium from other agents, to the Collector component for further processing;
- it passes orders, received via the comms medium from other agents, to the Planner component for further processing;
- it arranges for orders, reports and requests, generated by the Promulgator component of the agent, to be transmitted into the comms medium and propagated to other agents.

5.2 The Collector Component

The Collector component encapsulates the G2 processes of data collection (in order to achieve the Commander's Critical Information Requirement (CCIR)), data fusion, maintenance of the RP and intelligence assessment. The Collector is also responsible for alerting the agent's Planner component should significant events, or developments in the current situation, occur.

On activation the Collector establishes an intelligence assessment cycle by scheduling, for some time in the future, the next intelligence assessment. The Collector then moves into a wait state in which the Collector awaits the occurrence of certain events to which it will respond.

The Collector responds to four types of event, as follows. After each event the Collector returns to the wait state and awaits the next event.

Set CCIR. The Set CCIR event is generated by, and received from, the agent's Planner component. The event signals that the CCIR have changed. Arguments to the event describe how the CCIR have changed. The Collector, which maintains a description of the current CCIR, updates this description in response to the event. If necessary, new ISTAR asset taskings are issued in order to achieve the new CCIR. The ISTAR assets will subsequently generate reports that will be received and processed by the Collector (see below).

Time of Next Intelligence Assessment. When the time of the next intelligence assessment arrives the Collector performs an intelligence assessment. By intelligence assessment we mean the process of assigning meaning to observed activities and situations (as described by the RP). The assessment is also used to monitor the RP state and to trigger retasking of ISTAR assets if necessary (e.g., to fill in perceived gaps in the picture). The intelligence assessment also alerts the agent's Planner component if anything of significance is recognised in the picture. This is referred to as a "situation alert". Finally, when these activities are complete, the intelligence assessment process schedules, for some time in the future, the next intelligence assessment in order to keep the regular cycle of assessments going.

Alert (Report). At any time the Collector can receive a report, via an alert generated by the Comms component. When a report is received, the Collector fuses information contained in the report into the RP. The situation, as presented by the RP, is then monitored to see if the new information just received has changed the perceived situation significantly. If it has then the Collector alerts the Planner to the new situation, via a situation alert.

Alert (Request). At any time the Collector can receive a request, via an alert generated by the Comms component. The Collector processes requests itself, if feasible. Otherwise the Collector alerts the Planner to the request (via a "request alert") and the Planner assumes responsibility for processing it. For example, the Collector can handle a request for information itself - the Collector just instructs the Promulgator to compile a report from the information known by the agent (the RP) and issue it to the requester. On the other hand, a request for resources (e.g., men and materiel) from a subordinate will probably need the Planner to make a decision, so this request is passed on, via a request alert, to the Planner for consideration.

5.3 The Planner Component

The Planner component encapsulates the G3 processes of command decision-making and planning. The Planner is responsible for creating and maintaining a plan that will enable the agent to achieve the mission assigned to it by higher authority.

On activation the Planner establishes a situation assessment cycle by scheduling, for some time in the future, the next situation assessment. The Planner then moves into a wait state in which the Planner awaits the occurrence of certain events to which it will respond.

The Planner responds to four types of event, as follows. Unless stated otherwise, after each event the Planner returns to the wait state and awaits the next event.

Time of Next Situation Assessment. When the time of the next situation assessment arrives the Planner performs a situation assessment. By situation assessment we mean the process of regularly looking at the situation, as presented by the RP, and checking that the current activities of the agent (and its subordinates) are still appropriate to achieving the agent's mission, i.e., is the current plan still valid? And if it isn't, what to do about it? The situation assessment may include modifying the CCIR. This will involve informing the Collector (via the Set CCIR event - see above), so that ISTAR tasking can be changed accordingly. If the situation assessment concludes that no change to current activities is required (the plan is still valid) then no action is taken here, and activities carry on as before. The Planner reverts back to the wait state and awaits the next event. Alternatively, if the situation assessment concludes that a change to current activity *is* required (the plan has become invalid) then the Planner instructs the Promulgator to issue Warning Orders to all subordinate agents and the Planner moves into the Command Estimate (planning) activity (see below). Prior to leaving the situation assessment process the process schedules, for some time in the future, the next situation assessment in order to keep the regular cycle of assessments going.

Alert (Situation). At any time the Planner can receive a situation alert from the Collector, indicating that the Collector's situation monitoring (intelligence assessment) process has recognised something of significance in the RP. On receipt of such an alert the Planner itself performs a situation assessment, as described above.

Alert (Order). At any time the Planner can receive an order, via an alert generated by the Comms component. Orders are handled differently by the Planner, depending on their type (Operations, Fragmentary or Warning), as follows. If the received order is a Warning Order the Planner instructs the Promulgator to issue Warning Orders to all subordinate agents. No further actions are taken.

If the received order is an Operations Order or a Fragmentary Order the Planner carries out initial processing of the order, including:

- updating the RP with any information contained in the order;
- mission analysis;
- establishing the CCIR, and informing the Collector of these, so that the latter can ensure ISTAR assets are tasked appropriately.

Following these activities the Planner instructs the Promulgator to issue Warning Orders to all subordinate agents. The Planner then moves into the Command Estimate activity.

The Command Estimate is the key command decision-making process embedded in the agent's Planner component. The output of the Command Estimate is the commander's decision, from which a plan of action is constructed that will enable the agent to achieve the mission assigned to it by higher authority. Influences on the Command Estimate process include the following:

- own assigned mission;
- the perceived situation (as presented by the RP), including the enemy and friendly forces (e.g., strength and disposition) and the environment (e.g., ground, weather);
- doctrine (both own and the enemy's);
- likely outcomes (e.g., casualty estimates for personnel and equipment);
- scope and constraints (e.g., Area of Operations, Area of Interest, Area of Influence, Rules of Engagement (RoE), time).

We have developed two models of the Command Estimate process based on different models of human decision-making (Mason and Moffat 2000).

One model is based on the classical (and doctrinal) Generate-Evaluate-Select process. In this the decision-maker develops a number of alternative courses of action (to achieve the mission goal), evaluates each one using some Measure of Effectiveness (MoE) and selects the "preferred" one, judged on the basis of the selection criteria (e.g., maximum MoE). This decision-making approach is appropriate to the deliberate planning process, i.e., the planning activity that is conducted when sufficient time is available to appraise the situation in some detail.

The second model is based on the naturalistic decision making (NDM) paradigm, specifically Klein's Recognition Primed Decision (RPD) model. In this, the emphasis is on situation awareness (understanding what is going on) and the application of the decision-maker's experience of previous, similar situations to cue the rapid selection of an appropriate course of action. This decision-making approach is appropriate to the rapid planning process, i.e., the planning activity that is conducted when time is short or there is an incentive to act quickly.

The Command Estimate leads to a new, or modified, plan that is subsequently issued to subordinate agents by the Promulgator.

Alert (Request). At any time the Planner can receive a request, via a "request alert" generated by the Collector. At present, we have not implemented any Planner response to requests.

5.4 The Promulgator Component

The Promulgator component encapsulates administrative processes for managing the output of the agent, including:

- promulgation of the plan (creation and issue of Operations Orders and Fragmentary Orders) to subordinate agents;
- promulgation of Warning Orders to subordinate agents;
- issue of status/situation reports, within the reporting cycle and on demand;
- issue of requests.

On activation the Promulgator establishes a reporting cycle by scheduling, for some time in the future, the next reporting event. The Promulgator then moves into a wait state in which the Promulgator awaits the occurrence of certain events to which it will respond.

The Promulgator responds to six types of event, as follows. Unless stated otherwise, after each event the Promulgator returns to the wait state and awaits the next event.

Time of Next Report Transmission. When the time of the next report transmission arrives the Promulgator compiles, using information extracted from the RP, appropriate status and situation reports. The reports will be addressed to the agent's superior agent. The reports are passed to the Comms component for transmission out of the agent. Finally, when these activities are complete, the process schedules, for some time in the future, the next report transmission in order to keep the reporting cycle going.

Issue Plan. At any time the Promulgator can receive the Issue Plan event. This is an instruction from the Planner to issue the current Plan. The Plan component (see below) comprises a set of mission assignments, each comprising a list of tasks assigned to a particular subordinate agent. In response to the Issue Plan event the Promulgator is responsible for going through the Plan and, for each mission assignment in the Plan, doing the following:

- creating an appropriate Operations Order addressed to the specified subordinate agent;
- passing the Operations Order to the Comms component for transmission to the appropriate subordinate agent. The Operations Order (eventually) will be received by the subordinate agent's Comms component and will be handled as described in section 5.1 above.

At present, the Operations Order is quite basic, containing only the tasks assigned to the subordinate. The longer-term intention is to model the Operations Order on the standard five-paragraph format (Situation, Mission, Execution, Service Support, Command and Signal).

Issue Warning Order. At any time the Promulgator can receive the Issue Warning Order event. This is an instruction from the Planner to issue a Warning Order to each of the agent's subordinates. In response to this event, the Promulgator creates a Warning Order, addressing it to each subordinate agent, and passes this to the Comms component for transmission out of the agent and to the subordinates.

Issue Status Report. At any time the Promulgator component can receive the Issue Status Report event. This is an instruction from either the Collector or the Planner to issue a status report to one or more specified agents. In response to this event, the Promulgator creates a status report, addressing it to each specified agent, and passes this to the Comms component for transmission out of the agent and to the report addressee(s).

Issue Situation Report. At any time the Promulgator can receive the Issue Situation Report event. This is an instruction from either the Collector or the Planner to issue a situation report to one or more specified agents. In response to this event, the Promulgator creates a situation report, addressing it to each specified agent, and passes this to the Comms component for transmission out of the agent and to the report addressee(s).

Issue Request. At any time the Promulgator can receive the Issue Request event. This is an instruction from the Planner to issue a specified request to one or more specified agents. In response to this event, the Promulgator creates a request, addressing it to each specified agent, and passes this to the Comms component for transmission out of the agent and to the request addressee(s).

5.5 The Recognised Picture Component

The Recognised Picture (RP) component is the information store that contains all of the agent's knowledge of the outside world. The RP represents the agent's perceived state of the world. The RP is the entity upon which the Planner performs its command decision-making and planning activities.

The Planner requires an intelligence view of the battlefield (e.g., the locations and intentions of friendly and enemy forces) in order to plan the manoeuvre of the forces under the agent's command. Depending on the RoE this planning may be to avoid enemy forces or, alternatively, to confront them. The planning will also need to take into account various fixed attributes of the battlefield, such as terrain and culture, and other relatively permanent features, such as minefields.

In order to simplify the Planner's task, and in particular to reduce the number of options that need to be considered, the RP divides up the battlefield into a set of zones, as illustrated in Figure 4. This represents a geographical view of the RP. Each zone represents a tactically significant and distinct area of the battlefield. The zones are non-overlapping and the set of all zones covers the entire battlefield.

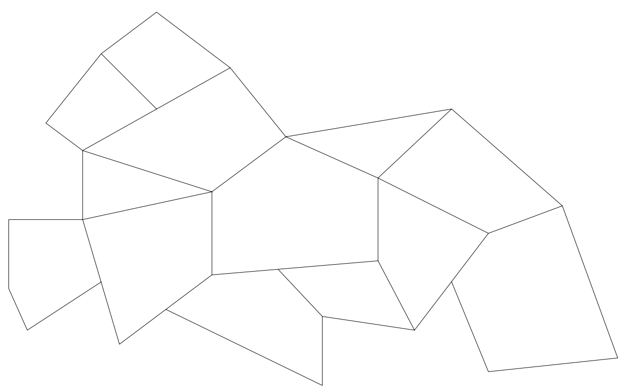

Figure 4: Recognised Picture – Geographical View

The geographical view of the RP (Figure 4) can be converted to a logical network view by considering which zones are reachable from other zones. This is illustrated in Figure 5, which shows the network view of the zone structure shown in Figure 4. Each zone is represented in the logical network as a node whose geographical position is the centre of gravity (weighted in tactical dimensions) of the zone.

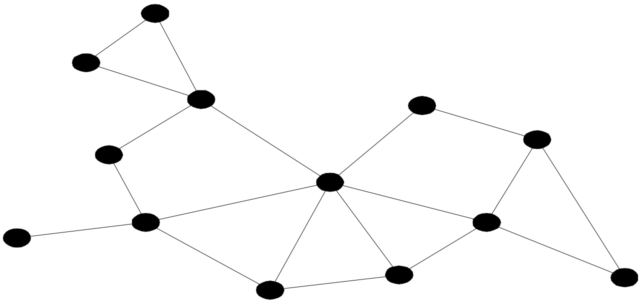

Figure 5: Recognised Picture – Network View

Each zone (node) has a set of attributes which hold the current perceived state of various features of that zone that are changing and which are of tactical relevance to the decision-making and planning activities of the agent. These features include:

- enemy and friendly forces present in the zone, in terms of strength, capabilities, current activities (and in the case of the enemy forces, perceived intent).
- status of the environment in the zone, including topography (climate, infrastructure, terrain type, obstacles, going, key points, NBC conditions) and demography (population, culture, ethnic/religious groupings).

Associated with each data item is the date and time of origin (timeliness) of the data item, and a measure of the uncertainty (confidence/reliability) associated with the data item.

The links in the logical network represent the connectivities between different zones. Each link has a set of attributes holding the current perceived state of link characteristics, such as:

- length;
- trafficability (as a function of entity type);
- capacity (as a function of entity type).

The purpose of the link attributes is to provide information to the planner such that judgements on the relative difficulty (cost) of moving between the connected zones can be made.

5.6 The Plan Component

The Plan component is the structure that holds the results of command decisions concerning the tasks to be assigned to subordinates in order to achieve the agent's mission.

The data structure used for representing the agent's Plan is an implementation of a Gantt chart, illustrated in Figure 6. The Plan comprises a set of mission assignments. These are the rows of the chart. There is one mission assignment per subordinate agent. A mission assignment is an associative pair comprising a subordinate agent identifier (Sub 1, Sub 2, ... Sub n in Figure 6) and a mission. The mission comprises an ordered set of tasks (Tk i,j in Figure 6) that the subordinate agent is required to undertake.

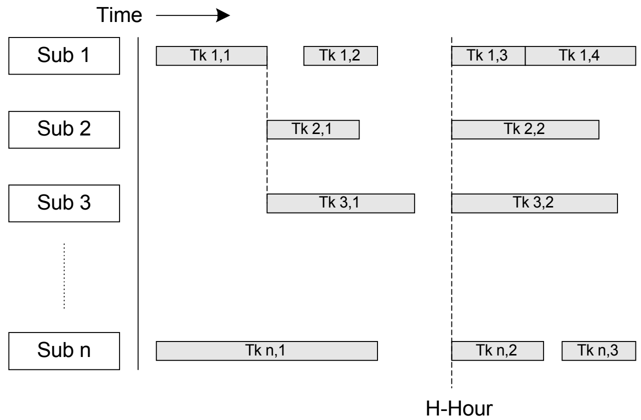

Figure 6: Plan Structure

The implementation of the Plan component enables dependencies between individual tasks to be specified. For example, in Figure 6 tasks Tk 2,1 and Tk 3,1 cannot start until task Tk 1,1 is complete. In addition, it is possible to establish global time points such as an H-Hour or D-Day for co-ordination purposes.

The Plan embodies the concept of mission command. Each task of each mission assignment specifies what is required and why, but not how to achieve it. The subordinate agent, on receipt of a order containing the mission assignment, is responsible for deciding how the mission is to be accomplished, and for accomplishing it.

6 SOFTWARE IMPLEMENTATION

We have implemented the OACIS command agent in software (C++) in the form of an extensible object-oriented framework.

The key design issue that we had to address here was the tension between the generic and specific aspects of C2. For, on the one hand, C2 is fractal (Dockery and Woodcock 1993), with the same C2 process *structure* existing wherever the agent is located in the command hierarchy. Yet, on the other hand, the *content* of individual C2 processes will differ from agent to agent depending on the actual role each agent is playing. This issue is resolved by the Template Method software design pattern (Gamma et al. 1995).

The Template Method design pattern enables us to define the skeleton of an algorithm (process structure) separately from the actual behaviour implemented by individual steps of the algorithm (process content). The invariant part of the problem – the process structure – is captured once, within a set of generic framework object classes, and the parts of the problem that can vary – the process content – are provided by role-specific subclasses. With this framework we can create role-specific command agents using subclasses, derived from the framework classes, which override and specialise the functional implementation of selected C2 processes.

7 MOSES – A PROOF-OF-PRINCIPLE

MOSES (Military OOTW Services-assisted Evacuation Scenario) is one of our research software testbeds. It is an object-oriented simulation, implemented in C++ and running on a PC under Windows NT. It is a simulation of a services-assisted non-combatant evacuation operation (NEO).

MOSES is the proof-of-principle for the OACIS command agent architecture and its software implementation. Our model of the NEO scenario requires 12 distinct command roles to conduct the operation. These roles fit within the command hierarchy shown below in Figure 7.

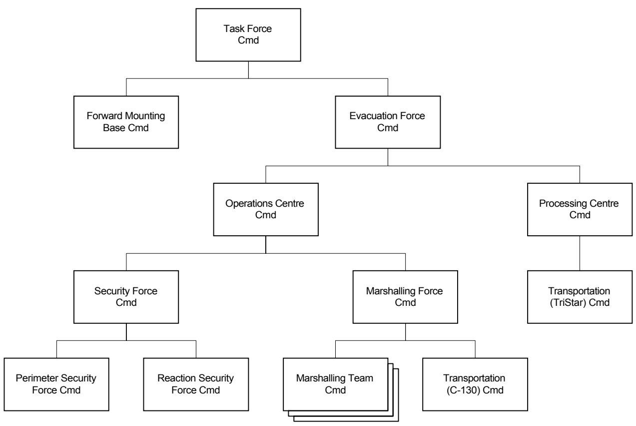

Figure 7: MOSES Command Hierarchy

Using the OACIS architecture and the software framework described above we successfully implemented command agents representing each of these roles.

8 CONCLUSION

In our research we have been investigating ways of representing the military C2 process within constructive simulations of conflict. We have developed an approach based on the idea of interacting agents. In this paper we have described OACIS - our design for the structure, behaviour and software implementation of these agents.

A key strength of OACIS is its implementation as an extensible, re-usable software framework. With this framework we expect to be able to build command agents that represent a wide range of decision-making entities – from the highest military headquarters down to, in principle, an individual human being. Our initial experiences with the MOSES testbed gives us confidence in this belief.

ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support provided to this research by the UK DERA (now DSTL) Corporate Research Programme, Technology Group 11 TA2.1: “C2 in OA Models”.

REFERENCES

Dockery, J. T., and A. E. R. Woodcock. 1993. *The Military Landscape*. Cambridge: Woodhead Publishing Ltd.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

Mason, C. R. and J. Moffat. 2000. Representing the C2 Process in Simulations: Modelling the Human Decision-Maker. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 940–949. New York, New York: Association for Computing Machinery.

AUTHOR BIOGRAPHIES

COLIN R. MASON is a Principal Consultant in CORDA Ltd, the Centre for Operational Research and Defence Analysis, a wholly owned subsidiary of BAE SYSTEMS. He received his BSc and PhD from University College, University of London. He has spent 17 years in CORDA conducting a variety of operational research studies in the land, sea and air domains. His current research interests include object-oriented simulation, modelling human behaviour and decision-making, complex adaptive systems, decentralised systems and distributed intelligence, and Network Centric Warfare. His email address is `<colin.mason@corda.co.uk>`.

JAMES MOFFAT BSc PhD FOR is a Fellow of DSTL, a Fellow of the Operational Research Society and visiting Professor to the University of Cranfield. He has spent over 20 years in defence-related operational research, advising the UK Ministry of Defence on the best balance of investment for defence equipment. He has also led research on stealth technology and sensors. His current research interests are in the improved modelling of C2 in the next generation of models of conflict. His email address is `<jmoffat@dstl.gov.uk>`.