Image Cover Sheet

CLASSIFICATION SECRET	**SYSTEM NUMBER** 153054

TITLE

NAVAL RESOURCE PLANNING IN REAL-TIME

System Number:

Patron Number:

Requester:

Notes: Paper # 5 of Multipart document, Sysnum 153057

DSIS Use only:

Deliver to: NL

7th RMC Symposium/Workshop

APPLICATIONS OF ADVANCED SOFTWARE IN ENGINEERING

RMC, Kingston, Ontario, Canada

#153054

NAVAL RESOURCE PLANNING IN REAL-TIME

Bruce A. Chalmers[1]

ABSTRACT

Effective resource management is essential during a military mission. There is a variety of resources to be allocated and scheduled in a timely manner to permit acquiring information to support decision making and to enable exercising command and control to accomplish the mission. Solutions must achieve cooperative, synergistic, and efficient utilization of all resources, while providing increased functionality for the human operator. Achieving these goals requires developing advanced resource management systems.

This paper is concerned with a weapon engagement manager for the Above–Water Warfare. This is a specific resource management system in a situation assessment and resource management component of a naval command and control system. The manager plans, coordinates and directs in real–time defence actions involving use of hardkill and softkill weapons to counter air and surface threats. We review our intelligent agent architecture for the manager that integrates both deliberative and reactive planning. We also describe a probabilistic model for conditionally planning and scheduling surface–to–air missile engagements against anti–ship missiles. Deliberation involves a faster than real–time discrete–event simulation of a number of missile defence plans to select one with the most successful simulation over some look–ahead horizon. Some algorithms for this planner are described and results from using a high–performance parallel architecture simulator to study improvements in computational performance from their parallel execution are also presented.

---

[1] Defence Scientist at Defence Research Establishment Valcatier, Courcelette, Québec, Canada.
Tel: (418) 844–4698, FAX: (418) 844–4538, E–mail:Bruce@cc.drev.dnd.ca.

INTRODUCTION

Effective resource management is essential during a military mission. There is a variety of resources to be allocated and scheduled in a timely manner to permit acquiring information to support decision making and to enable exercising command and control (C2) to accomplish the mission. Solutions must achieve cooperative, synergistic, and efficient utilization of all resources, while providing increased flexibility and functionality for the human operator. To this end, an R&D project aimed at designing advanced real-time resource management systems (RMSs) for naval combat systems has been initiated [1-2]. It is currently addressing three broad issues: first, design, implementation and testing of adaptive software agents for managing defence, computational and communication resources in a naval combat system; second, performance enhancements from concurrent computing; and third, embedding agents in a real-time C2 architecture.

An RMS is an automated real-time system for aiding military personnel in optimizing the use of limited resources. It provides support in and relief from performing critical, complex C2 functions in a highly stressful environment within demanding real-time constraints. An example is a sensor management system for controlling one or more sensors to respond effectively to a changing environment, numerous operator commands, and a variety of functions and missions. Particularly challenging problems arise in domains involving uncertain and dynamic battle environments populated by high-velocity threats of varying densities, rates of arrival and life spans. In the Above-Water Warfare (AWW), for example, a variety of sensor, weapon and communication systems under the control of a warship or naval force must be integrated and managed effectively, both individually and collectively, if they are to be used in a coordinated and directed manner and defensive capability is to be optimized. Moreover, problem difficulty grows with improvements in sensor, guidance and communication technologies that lead to increased complexity in the battle space and with reduced response time due to advances in threat capabilities.

Resource management is a closed-loop process that continually interacts with human operators and an external environment consisting of a number of dynamic entities. Synchronizing actions with occurrences in this evolving world imposes critical timing constraints on its temporal behaviour. A generic RMS is illustrated in Fig. 1. Data from a variety of sources is used to make situation assessments about the state of the environment. These assessments, together with human interaction via a human-computer interface (HCI), as required or as response time permits, drive the planning function for allocation and scheduling of critical resources and response coordination. Human interaction can take the form of commands and/or requests for support from the RMS. This may entail informing or advising the operator by providing recommendations, suggestions, etc., depending on support requirements and on the division of responsibility between the RMS and the operator. Continually monitoring effects of executed actions and recognizing significant occurrences that require new or revised responses closes the loop, leading to adaptive feedback control of the environment. Planning a course of action involves reasoning in time and about time. Decisions may be required periodically (e.g., as a result of sensor input) or aperiodically (e.g., due to sporadic interactions with the operator).

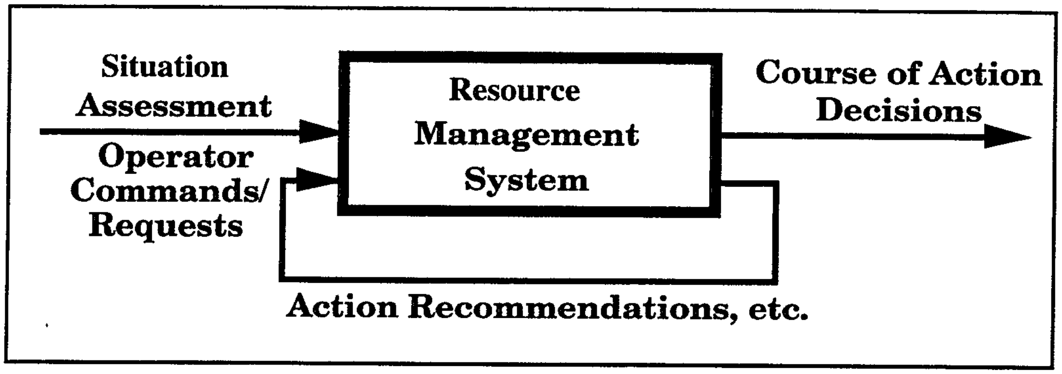

Figure 1: A generic resource management system

The rate of significant occurrences in the world may be time-dependent. The time to formulate and execute a response may therefore vary from one planning episode to the next, as a function of hard and soft real-time constraints depending on situation context. In the AWW, this time is extremely limited; it generally ranges from a few minutes to a few seconds. Resource management strives to optimize the expected "eventual degree of mission accomplishment", from the perspective of the military personnel involved. Internal planning models of the RMS must therefore be consistent with the operator's mental states (beliefs, goals, plans, etc.) to ensure that what is planned to achieve is indeed what is desirable to achieve and that the overall effect of the management system is to improve the performance of the operator in achieving the mission.

The RMS we focus on here is important for naval combat systems. Formulations of various resource allocation and scheduling problems involved have previously been studied [3-5]. Critical resources include the ship's hardkill and softkill weapon systems. A hardkill system is aimed at physically destroying a threat by collision or explosion. Softkill is directed at the control and guidance subsystems of the threat in an attempt to divert it away from its target. Softkill is effected by electronic warfare (EW) systems using a number of electronic countermeasures, including confusion, distraction, deception, and seduction [6]. Specifically, we examine some aspects of the design of an AWW weapon engagement manager (WEM) for countering air and surface threats. This RMS plans, coordinates and directs in real-time defence actions to achieve mission goals at the single ship (self or area defence) or force command (cooperative engagement) levels. Cooperative engagement (CE) involves integrating multiship weapon systems. This assumes a robust network of tactical data links and a computational capability that permit obtaining a force-level picture of the tactical situation. The picture is derived from organic and non-organic information from a variety of sources, including radar, electronic support measures, infrared search and track, identification friend or foe transponder responses, as well as intelligence information from shore and various deployed units. CE is then achieved via tactical control links. We assume for now a centralized or virtually centralized force level C2 architecture. Many details omitted in the presentation below can be found in [1-2].

SOFTWARE ARCHITECTURE OF THE WEAPON ENGAGEMENT MANAGER

Both autonomous and supportive behaviours are required of the WEM. Its autonomous behaviour continually senses and reacts to perceived occurrences. Occurrences include events that take place in the combat environment and actions or activities involving WEM intervention. Its supportive behaviour responds to requests for support or commands from a commander. Our current design couples both feedback and feedforward control to permit dynamic interleaving, and even overlapping, of incremental planning and execution of plans. Feedback control relates to reactive response to immediate change. Reactive planning has a very short term horizon. It is driven by precompiled stimulus-response knowledge that provides almost immediate response depending on the current state of the world (rule-driven). However, it is also guided by deliberative input. Feedforward control involves high-level deliberation based on projecting the perceived history of past and ongoing occurrences over some forecast horizon. Deliberation can be viewed as deep planning over a dynamic look-ahead horizon. It is generally accepted within the artificial intelligence (AI) community that "some balance between quick responses to environmental change and time for deliberation to allow reasoning about what actions to perform seems to offer the best hope for truly adaptive, intelligent agents" [7].

The WEM uses a variety of knowledge, including knowledge about the behaviour of entities in its combat environment, of doctrine and tactics, and of conditions for the feasibility of specific actions as well as their (stochastic) effects on the state of the environment. It continually adapts its reasoning to take account of an evolving picture of the tactical situation, computational resources available for reasoning, response time, and the interdependence between reasoning (planning) and the quality of the information on which it is based (sensing). Resource-bounded reasoning to permit dynamically trading off plan quality against computational cost and response time is therefore required. In addition, incomplete knowledge about enemy intentions and behaviour, and imperfect information due to limited sensor coverage, sensor error, information processing inadequacies, etc., imply that deliberative plans must guide defence actions, not control them.

We now review our software architecture for the WEM. A two-layered architecture is used to hierarchically decompose the functions of the WEM, with various world models distributed among the layers. A fundamental consideration is Saridis's principle [8] of combining "increasing intelligence with decreasing precision", whereby reasoning in one layer is used to influence, but not control, actions in a lower layer which interacts more directly with the world. Figure 2 shows the decomposition of the principal functions (represented by blocks) and information fluents (represented by arrows). Fluents corresponding to interactions with the commander are not shown for simplicity.

The deliberative planner responds to significant changes in the tactical picture that require (re)planning. It computes a plan, subject to engagement doctrine and resource availability, for assigning and scheduling use of hardkill and softkill weapon systems and tracking and guidance systems over a plan horizon within the forecast horizon. Recognizing significant occurrences is handled by the characterizer, using its internal model of the world and information from the projector and the effector. This model permits nonmonotonic and probabilistic reasoning about change and the effects of actions and their effectiveness over the forecast horizon. The projector maintains fluents corresponding to extrapolations over the forecast horizon of the perceived historical states of the world. Projection includes: extrapolating potentially hostile tracks and ship manoeuvres; predicting occurrences of events arising from ongoing engagements, previously committed actions, etc., including outcomes of defence actions and threat strikes on own ships; predicting when potentially hostile tracks will be engageable, from which ships and by which weapon systems on such ships, as well as measures of effectiveness of such defensive actions; predicting effects or restrictions associated with obstructions (parts of a ship's structure, chaff clouds, offboard decoys, etc.), environmental conditions, or operational constraints (EMCON, risk of fratricide, etc.) which prevent threat interception, or which, at least, significantly degrade effectiveness of such actions; and predicting positive and negative interactions [6] that result from concurrent use of hardkill and softkill weapon systems. By communicating with weapon controllers, the effector coordinates and directs execution of plans and uses tactical situation fluents to assess the effectiveness and outcomes of defensive actions.

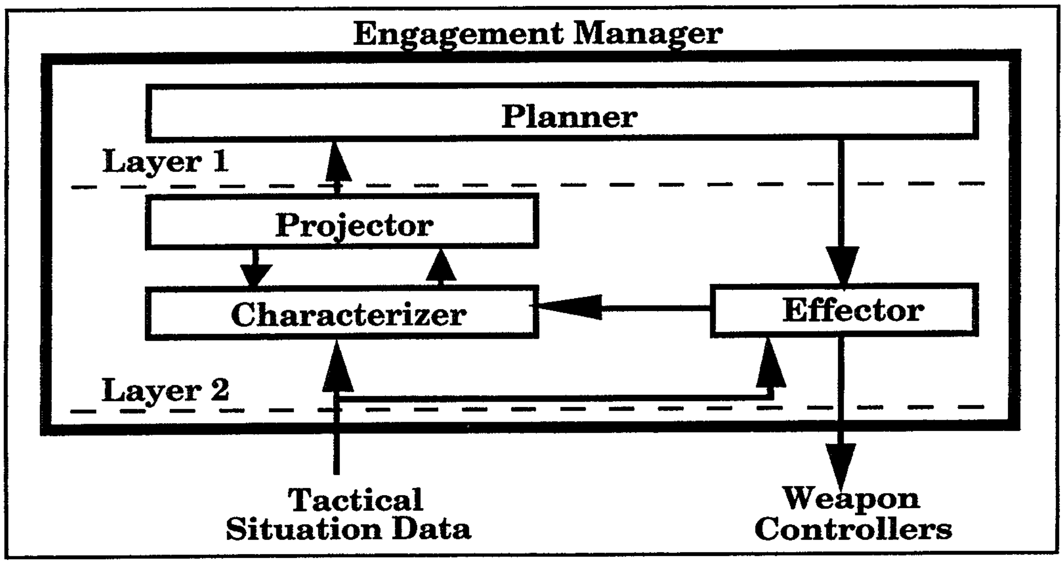

Figure 2: Layered architecture

Deliberation is based on projections by the projector. Such projections will necessarily be defeasible due to statistical error and incomplete or imprecise information. An important implication is that the effector implements reactive plan repair strategies when real-time deviations from projections are "small". Tracks that cannot be so engaged are fed back via the characterizer for replanning. Replanning is triggered by the characterizer signaling the projector to send its latest projections to the planner. This is driven by a number of asynchronous occurrences, including a requirement to extend the window of the latest plan, the commander's request to reshape the defence response, and updates in the tactical picture such as detections of new tracks whose estimated times of first engagement fall within the window of the plan currently being executed.

A weapon controller is cued by the effector. A cue activates a low-level process for using a weapon system. The controller also monitors any servo loops associated with this process. There are two data paths in the architecture leading from the arrival of situation fluents to the generation of cues. The first is via the effector in layer 2 only. This path effects reactive plan modifications due to changes in the tactical picture. The second path is via both layers 2 and 1 and implements deliberative fluent transformations. While response to change along this path is fast, it is not on the same time scale as that on the other.

The requirement that the WEM respond within real-time constraints to occurrences in the combat environment and operator commands necessitate explicit modelling of its real-time behaviour if predictable performance is to be achieved. This aspect has been covered in some detail in [9, 1-2] for the planner. It is based on the idea of a time-dependent planner [10] varying deliberation according to time pressure. It provides service in one of two anytime modes, viz., contract or interruptible. In contract mode, it produces a plan within a specified amount of computing time. In interruptible mode, this time is not specified and it must be capable of returning a plan even if interrupted unexpectedly. One way of achieving the anytime property is to use a rolling plan horizon (Fig. 3) whose size is determined at the time *t* of a service call. Modelling other temporal aspects is ongoing.

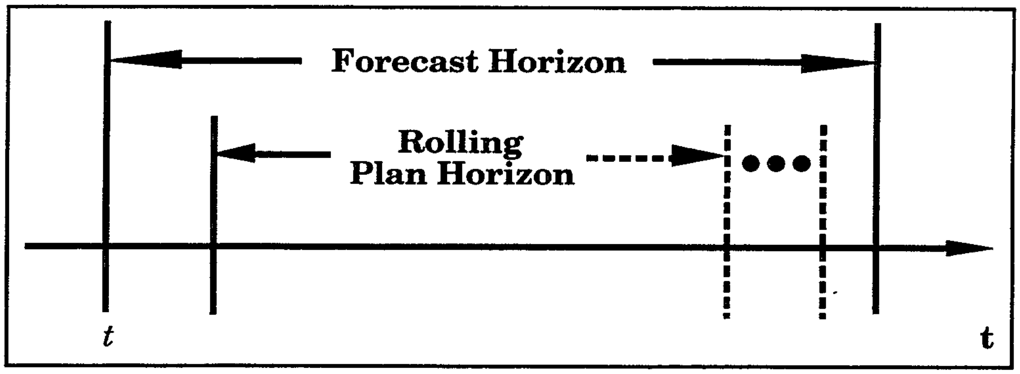

Figure 3: Rolling Plan Horizon

CONDITIONAL PROBABILISTIC PLANNING OF SAM ENGAGEMENTS

We now review an anti-air warfare (AAW) model [9] for planning surface-to-air missile (SAM) firing and guidance actions against anti-ship missiles (ASMs) in the WEM. A plan is a (conditional) schedule of potentially concurrent engagements for the plan horizon. The model has a number of features that distinguish it from ones usually studied in the AI literature which is largely concerned with planning in static, deterministic worlds based on a domain theory for action and on constructive situation calculus representations of planning. In such a representation, planning is essentially characterized as the selection of a sequence of effectively instantaneous actions to achieve complex, higher-level goals defined in some formal language.

The model generates conditional or contingent plans; that is, uncertainty at planning time about engagement outcomes is explicitly accounted for inside the plan by incorporating reactive response (when feasible) to an unsuccessful engagement. A plan is therefore closed-loop, with certain actions conditional on information available at execution time from kill assessment. This contrasts with an alternative approach based on open-loop or unconditional planning, where there are no contingencies and the fact that effectiveness assessment information will become available at execution time is ignored. This alternative has the effect of trading off plan quality for reduced complexity in plan generation; however, it is at the expense of more frequent, time consuming replanning.

Planning is utility-driven; that is, the agent has a utility function expressing the commander's current desirability about world states and an effective plan is one that maximizes the expected utility of the state to which the world evolves. This provides a well-known normative basis for rational choice under uncertainty. The engagement model assumes one or more ships under attack by ASMs. Support ships can concurrently perform various functions of surveillance/tracking, SAM launching for point or local area defence in the missile engagement layer, point defence by other weapon systems in their close-in layer, kill assessment, and radar support (when necessary) for missile guidance. Included are home-all-the-way (HAW) guidance and midcourse guidance with semi-active terminal homing. Missile firing actions are subject to shoot-look-shoot (SLS) doctrine or some related generalization involving salvos (e.g., SSLS); firing policy may even use variable salvo size determined by the commander at planning time.

A dynamical system representation of kinematic entities in the battle environment that over a forecast horizon is reasonably inert is assumed; this is to permit its process models to be estimated or identified with sufficient accuracy to justify their use for deliberation. However, firing actions are capable of influencing inertia, subject to outcome uncertainty. Deliberation proceeds under the assumption of completeness in projections. For example, the model does not currently incorporate expectations of engagement opportunities related to as yet undetected threats (surprise events). A branching model of time is used, represented as a collection of world histories or chronicles, each a potential unfolding of the world's evolution. A history is a mapping from time points to a state space; its value at a time point (realized or projected) is a description of features of the battle world. In the following, the planner is assumed for simplicity to provide high-level plans formed only from cues for missile firing, guidance and execution monitoring. Additional cues for using tracking systems to support engagements are readily incorporated.

A plan P is represented by a rooted labelled directed tree. A node with one direct successor cues a defence action or activity. An action is (essentially) instantaneous. An activity takes place over a time interval; an example is continuous wave or interrupted continuous wave illumination of a threat by a multifunction radar/illuminator (MFR/I) system to home a semi-active missile to an intercept with the threat. An activity is represented by two discrete actions corresponding to starting and ending the activity. An action includes these actions as well as firing actions. A firing action involves a specific ship-threat pairing. Actions/activities may be independent or loosely or tightly coupled; for example, a loose coupling may exist between two engagements competing for terminal illumination over overlapping time intervals; and a tight coupling exists between firing a semi-active missile and its terminal guidance. A node with two or more direct successors corresponds to the stochastic outcome of either a firing action or a threat strike on an own ship; the first type cues execution monitoring of the effectiveness of a firing action; the second signals transition to a degraded defence capability in case of a successful threat strike on a support ship and potential downgrading in importance of a defended asset in case it is a successful strike on a protected ship.

A plan is feasible if the time-stamped action/outcome sequence in any chronicle (path) in the plan tree is realizable. Based on resource availability and control on a planning episode, the planner chooses a most preferred, feasible plan. Given the probability distribution on outcome states, $s \in S$, at the end of the plan horizon and a utility function u on states, possibly depending on the planning episode, the agent chooses, when it exists, an optimal plan, denoted by P^*, according to:

$$P^* = arg\,\max_{P} \; v(P) \;\; , \tag{1}$$

where the optimization in eq. (1) is over all feasible plans and the value $v(P)$ of plan P is the expected projected value of u resulting from its execution, given by:

$$v(P) = \sum_{s \in S} prob(s|P)u(s) \ . \tag{2}$$

SEQUENTIAL ALGORITHMS

Starting from the latest tactical picture, there may be infinitely many histories for the evolution of the world thereafter. In general, therefore, the feasible plan space in eq. (1) is infinite. Obtaining a finite plan space requires narrowing attention somehow to a finite subspace. Our procedure, which leads to a potentially suboptimal plan, currently proceeds in two steps.

Step I:

Generate a finite set A of cues for missile firing, guidance and kill assessment, each cue labelled with its time of occurrence during the current plan horizon.

Step II:

Determine a best plan P^* from the restricted finite set $\Pi(A)$ of feasible plans whose cues occur in A; that is,

$$P^* = arg \max_{P \in \Pi(A)} v(P) \ . \tag{3}$$

Since the intention is to produce an optimal plan, a desirable property of A is that a best plan from eq. (3) is also an optimal plan. Unfortunately, in general, we do not yet know how to achieve this; however, we are able to do so in one important setting obtained by restricting the problem domain; in fact, the heuristic for Step I which we now present has been developed by considering this setting [2].

Step I

Assume that a battle scenario involves m support ships, indexed $i = 1, ..., m$, and n ASM threats, indexed $j = 1, ..., n$. SAMs are midcourse guided with semi-active terminal homing. Suppose that a planning episode has just been triggered. One input is an $m \times n$ matrix $E_0 = (\bar{I}_{ij})_{m,n}$, where each $\bar{I}_{ij}$ is a time set defined as follows: $\bar{I}_{ij}$ is the set of time points within the current plan horizon at which it is feasible for the j-th threat to be intercepted by a SAM fired from the i-th support ship during the current plan horizon. $\bar{I}_{ij}$ is assumed to be a closed and bounded subset of the real line. Assume that each column (row) of E_0 has at least one nonempty entry; otherwise, the corresponding threat (support ship) can be removed from consideration at the current planning episode. $\tau(\bar{I}_{ij})$ denotes the latest time point of the set $\bar{I}_{ij}$, when $\bar{I}_{ij}$ is nonempty; otherwise, $\tau(\bar{I}_{ij}) = -\infty$. Then, $\tau(E_0)$ is defined by $\tau(E_0) = \max_{(i,j)} \tau(\bar{I}_{ij})$. Assume, for simplicity, that ship i must use one of its own fire-control radars to provide terminal guidance to a SAM that it fires at threat j. This requires illuminating the threat for a certain amount of time prior to an intercept at some time point ξ in $\bar{I}_{ij}$. The illumination time is of length c_{ij} time units and the time interval for the illumination is $(\xi - c_{ij}, \xi]$. In this case, the order to launch the SAM must be given at time $f_{ij}(\xi) \leq \xi - c_{ij}$. The inputs to Step I for the illumination timing requirements and these functions are represented by two $m \times n$ matrices $\Gamma = (c_{ij})_{m,n}$ and $F = (f_{ij})_{m,n}$. To simplify matters, an interval of illumination terminates with a kill assessment. In practice, illumination may be independently terminated just prior to kill assessment and the heuristic is readily modified to accommodate this. Assume also that SLS firing doctrine is used; that is, as many SAMs as feasible to counter a threat may be fired, one SAM at a time, with a kill assessment terminating each engagement before initiation, when necessary, of the next. Each engagement is represented by three cues: firing a missile (FM), beginning illumination (BI), and ending illumination and doing a kill assessment (EI-KA). An element a in A is a 4-tuple, $a = (\alpha, \beta, \gamma, \delta)$, where α is the time of the cue, β is the type of cue, γ is the number of the ship to which the cue relates, and δ is the number of the threat involved in the engagement.

The heuristic is based on traversing a dynamically generated tree and a temporal line sweep, backwards in time. Each node v of the tree has an associated $m \times n$ node matrix $M(v)$, each of whose entries is a time set. The root is associated with the initial matrix E_0. A time set in the node matrix of each remaining node is derived by various successive transformations of the corresponding time set in the root matrix. The matrix associated with a leaf node has all of its sets empty; otherwise, the node is an interior node. An edge in the tree corresponds to the three 4-tuples of cues for a specific engagement. A node is expanded when it is visited. Expanding a node involves generating all of its children, as well as visiting each parent-child edge for which it is the parent. Generating a child involves transforming the node matrix of its parent to produce the node matrix of this child. On visiting an edge, its three cues are inserted into the running version of A, provided they have not been previously generated. The tree traversal sweeps temporally over the set of actions for the current plan horizon in a manner that satisfies a monotonicity property: the inequality $\tau(M(v')) \leq \tau(M(v))$ holds for each child v' of an interior node v of the tree. It is for this reason we refer to the enumerative search as a temporal line sweep, backwards in time.

During the traversal, the algorithm maintains the current version of A, as well as a list $\Im$ of node matrices of frontier nodes. It starts by initializing $A \leftarrow \varnothing$ and $\Im \leftarrow \{E_0\}$. It then executes a cyclic step. At each cycle, a node matrix E' in $\Im$ is selected and removed from $\Im$; the corresponding node is then expanded by calling a procedure *expand(* E' *)*. This cycle is repeated until $\Im$ is empty, at which point the algorithm terminates. The specifics of the procedure *expand(* E' *)* are described in the following.

Element Insertion:
Insertion of the cues associated with a SAM engagement of the j'-th threat by the i'-th support ship for an intended intercept at time ξ requires inserting into A the three 4-tuples, $(\xi, \text{EI-KA}, i', j')$, $(\xi - c_{i'j'}, \text{BI}, i', j')$ and $(f_{i'j'}(\xi), \text{FM}, i', j')$, provided these elements have not previously been generated. We denote this procedure by *insert(* ξ, i', j' *)*.

Child Generation:
Let $E' = (I'_{ij})_{m,n}$ be the node matrix of an interior node. Each child of this node is associated with some ship-threat pair (i', j'). The function *generate(* E', i', j' *)* generates the node matrix $E'' = (I''_{ij})_{m,n}$ of the child determined by the i'-th support ship and the j'-th threat by transforming the parent matrix E' according to:

$$I''_{ij} = I'_{ij}\ ,\ i \neq i' \text{ and } j \neq j'\ ,$$

$$I''_{ij'} = I'_{ij'} \bigcap (-\infty, f_{i'j'}(\tau(I'_{i'j'}))]\ ,\ i = 1,\ \ldots,\ m\ ,$$

$$I''_{i'j} = I'_{i'j} \bigcap (-\infty, \tau(I'_{i'j'}) - c_{i'j'}]\ ,\ j = 1,\ \ldots,\ n\ ,\ j \neq j'\ .$$

Node Expansion:
Let $E' = (I'_{ij})_{m,n}$ be the node matrix of a frontier node selected for expansion. To expand this node, a ship-threat pair (i^*, j^*) is first chosen that satisfies $\tau(I'_{i^*j^*}) = \tau(E')$. This node has a branching factor of at most $(m+n-1)$. There is one child associated with each ship-threat pair (i', j') in the set $C(E') = C_1(E') \bigcup C_2(E') \bigcup C_3(E')$, where

$$C_1(E') = \{(i^*,j^*)\},$$

$$C_2(E') = \{(i,j^*) : i \neq i^*,\ \tau(I'_{ij^*}) > f_{i^*j^*}(\tau(E'))\}, \text{ and}$$

$$C_3(E') = \{(i^*,j) : j \neq j^*, \tau(I'_{i^*j}) > \tau(E') - c_{i^*j^*}\}.$$

To complete the node expansion procedure *expand*(E'), the following two steps are then performed for each child of the selected frontier node: first, the function *generate*(E',i',j') is invoked to compute the node matrix of the child associated with the ship-threat pair (i',j') and, if this child is an interior node, its matrix is inserted into the list $\Im$; and second, the procedure *insert*$(\tau(I'_{i'j'}),i',j')$ is executed in order to insert into A the cues for the corresponding parent-child edge.

Step II

This step rests on an encoding of all members of $\Pi(A)$ as solution trees of a finite AND/OR decision tree T. In addition, P^*, defined by equation (3), may be identified with an optimal solution tree in T. T can be explicitly generated as needed. The nodes of T are labelled with a time-stamp and state information on the battle world at that time. Interior nodes of T are of two types, decision or outcome. A decision node is an OR-node. It corresponds to a specific action in A and the two edges emanating from it represent the choice to perform the action or not. An outcome node is an AND-node. It corresponds to the outcome of some previous firing action or a threat strike on an own ship. Such a node has two or more direct successors, depending on the granularity of the stochastic threat destruction and ship damage models used. Emanating edges partition the finite set of potential histories according to the outcome, which is unknown at the time of planning, and they are labelled with outcome probability measures supplied by the damage model. The value $v(P)$ of plan P is recursively computed as a composition of the values of its subtrees, using the values of u on the world states at leaf nodes. A best plan is computed by performing a traversal of T. This procedure can be viewed as a form of simulation-based planning in which a faster than real-time discrete-event simulation of a number of plans is executed to select the plan with the most successful simulation.

PARALLEL ALGORITHMS

Parallel computing is a fast growing area whose purpose is to solve problems which cannot be solved effectively or in a reasonable amount of time by conventional computers.

The speed of a conventional computer is bounded. Furthermore, for certain problems such as NP-hard problems, as is the case for our missile engagement planner [1], it is unlikely that efficient (that is, polynomial time) algorithms will ever be found. Therefore, it is not only natural, but also necessary, to apply parallel computing technology to speed up the computation. A parallel computer consists of a number of processors each possibly having its own local memory. These processors may communicate during a computation in order to exchange data and intermediate results. Interprocessor communication may be done either through a globally shared address-space (shared-memory), or by explicit message passing over a network that interconnects the processors (distributed-memory), or by some combination of these two modes (distributed-shared-memory). A brief description of our multiple-instruction multiple-data (MIMD) parallel algorithms for Steps I and II follows.

As already seen , the sequential algorithms for Steps I and II consist of various tree traversals. We have investigated their parallelization based on performing depth-first tree traversals in parallel. These parallel algorithms have the following general structure [2]:

- generate and distribute one frontier node to each processor of the parallel machine;
- in parallel, each processor then performs a depth-first search using its frontier node as the root of its search to produce a set of cues or a portion of a best plan; and finally,
- the sets of cues or portions of plans computed by each processor are merged in parallel.

Since the initial work distribution among the processors can be unbalanced, various dynamic work sharing schemes are used to redistribute work when a processor is idle.

PERFORMANCE RESULTS

We have tested our sequential and parallel algorithms for planning SAM engagements on twenty randomly generated scenarios involving two ships under attack by eight ASM threats. We concentrate here on their computational performance. These results were obtained using a high-performance parallel-architecture simulator capable of simulating a variety of MIMD multiprocessors [11]. Our parallel algorithms were run on simulated distributed-shared-memory MIMD machines connected by a hypercube network. The sequential and parallel results are summarized in Tables I and II, respectively. Results are averaged over the twenty scenarios. Execution times are given in both processor cycles and wall-clock time, assuming a processor speed of 40 MHz. They indicate that the computational performance of the parallel implementation scales linearly with the size of the parallel machine. They also suggest the feasibility of using advanced planning techniques that can satisfy the real-time requirements of this domain.

Table I: Sequential execution time of combined Steps I and II

	Execution Time (X 1000000 cycles)	Seconds (40 MHz Clock)
AVERAGE	393.3	9.83
MINIMUM	55.9	1.40
MAXIMUM	1805.7	45.14
ST. DEVIATION	452.0	11.30

Table II: Parallel execution time of combined Steps I and II on a hypercube

# of PROCESSORS	**2**	**4**	**8**	**16**	**32**	**64**
X 1000000 Cycles						
AVERAGE	359.2	290.1	182.2	94.8	59.8	31.4
MINIMUM	47.7	38.6	17.1	7.7	6.1	3.2
MAXIMUM	1504.3	1388.6	731.9	338.1	322.8	122.7
ST. DEVIATION	403.5	348.7	217.8	115.7	81.3	37.9
Sec. (40 MHz Clock)						
AVERAGE	8.98	7.25	4.56	2.37	1.50	0.79
MINIMUM	1.19	0.97	0.43	0.19	0.15	0.08
MAXIMUM	37.61	34.71	18.30	8.45	8.07	3.07
ST. DEVIATION	10.09	8.72	5.45	2.89	2.03	0.95

CONCLUSIONS AND FUTURE WORK

This paper has reviewed the design of a real-time resource management system for the Above-Water Warfare. The architecture presented integrates reactive and deliberative planning to permit dynamic interleaving, and even overlapping, of incremental deliberative planning with reactive execution of plans. Results of static open-loop testing of the computational performance of our sequential and parallel algorithms for planning surface-to-air engagements against anti-ship missiles have also been presented. Further work is continuing to implement and extend the management system and to permit closed-loop testing of its various components. In addition, various refinements of the planning algorithms will be investigated, including pruning the world histories examined during planning and use of improved work distribution schemes in the parallel algorithms.

REFERENCES

1. Chalmers, B.A., "An Adaptive Planner in a Layered Real-Time Architecture for an Engagement Manager in a Naval Threat Evaluation and Weapon Assignment System: An Overview", DREV R-4768, July 1994, 56 p.
2. Chalmers, B.A. and Da Ponte, P., "MIMD Algorithms for Naval Resource Planning: Overview and Preliminary Assessment", DREV R-9420, to appear, 106 p.
3. Diamond, M.D. and Carducci, O.M., "Decision Processes for Large Scale Resource Allocation Problems", Proceedings of the 8th MIT/ONR Workshop on C^3 Systems, Cambridge, Massachusetts, 1984, pp. 153-160.
4. Boyer, D.D., Perry, E.L., Price, E.L., and Godrey, S., "Force Level Engagement Control in Naval Anti-Air Warfare", Proceedings of the 1990 Command and Control Research Symposium, Monterey, California, 1990, pp. 204-211.
5. Lee, Y. and Sherali, H.D., "Unrelated Machine Scheduling with Time-Window and Machine Downtime Constraints: An Application to a Naval Battle-Group Problem", Annals of Operations Research, Vol. 50, 1994, pp. 339-365.
6. Thé, L. and Liem, K.D., "Integrated Naval Air Defense: Co-ordinating Hardkill and Softkill Weapons", International Defense Review, Vol. 6, 1992, pp. 567-570.
7. Garvey, A. and Lesser, V., "A Survey of Research in Deliberative Real-Time Artificial Intelligence", The Journal of Real-Time Systems, Vol. 6, 1994, pp. 317-347.
8. Saridis, G.N., "Toward the Realization of Intelligent Controls", Proceedings of the IEEE, Vol. 67, No. 8, 1979, pp. 1115-1133.
9. Chalmers, B.A., "Toward Time-Dependent Planning of Missile Allocation and Engagement Scheduling in a Naval Threat Evaluation and Weapon Assignment (TEWA) System", Proceedings of the 1993 DND Workshop on Advanced Technologies in Knowledge-Based Systems and Robotics, Ottawa, November 1993, pp. 677-684.
10. Dean T. and Boddy, M., "An Analysis of Time-Dependent Planning", Proceedings of the Seventh National Conference on Artificial Intelligence, Minneapolis, Minnesota, 1988, pp. 49-54.
11. Brewer, E. A., Dellarocas, C. N., Colbrook, A., and Weihl, W. E., Proteus: A High-Performance Parallel-Architecture Simulator, Technical Report MIT/LCS/TR-516, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1991, 25 p.