Proceedings of the 1988 Winter Simulation Conference
M. Abrams, P. Haigh, and J. Comfort (eds.)

Air operations modelling in a wargaming environment

Paul E. Rubin, Ph.D. and Joseph L. Sowers, Ph.D.
Computer Sciences Corporation
Defense Systems Division

ABSTRACT

The Enhanced Naval Warfare Gaming System (ENWGS) is a large scale, computer based, interactive wargaming system. It supports curricula and studies at the Naval War College and Tactical Training Groups. The Air Operations (Airops) subsystem comprises the user interface, models, and data structures for representing the deployment and control of Naval air assets within the wargame. This paper will present a discussion of the overall Airops subsystem design and implementation. In an effort to avoid having to present a detailed view of the ENWGS system, much of this discussion will center on data flow, model control relationships and relationships between functional modules within the Airops subsystem.

INTRODUCTION

The Airops subsystem may be considered to be composed of four components: models, model controllers, user system interface, and data structures (and values). The interface between Airops and the rest of ENWGS is primarily through data and the ENWGS event execution mechanism. The primary models within Airops represent the activities of launching, recovering, loading (and unloading), and refueling aircraft. Secondary models provide means of representing events such as aircraft damaged at launch or recovery, designation and use of alert aircraft and other common operations necessary to simulate Naval air operations. Model controllers are those modules which implement the necessary scheduling functions and effect the transition between activities; for example, pre-launch preparation, launch, mission prosecution and recovery. The user interface provides extensive facilities for planning flight schedules, defining and assigning aircraft loads and missions, and the manipulation of aircraft in alert status. An extensive set of reporting capabilities is also provided. Formal reports may be requested by the user and a wide variety of alerts and advisories may be sent to the user depending on events such as low fuel conditions. The fourth major component, the data structures, provides the interprocess communication necessary to correlate the actions of all of the models, controllers, and interface processes.

One of the objectives of the Airops subsystem is to provide a sufficiently accurate representation, yet at the same time not overburden the player with details nor the system with processing. The key, of course, as in any model or simulation, is in recognizing the essential processes and parameters. For example, the time needed to arm an aircraft before launch is essential. However, within the context of a wargame, it is sufficient that there merely be a delay. Variations due to type of ordnance or specific type of aircraft are not considered; however, depending on the model level chosen, stochastic processing may be used, more accurate time computations and environmental conditions may be considered. Human factors may be added to time delay. For example, while the actions of any given crew member are not considered, placing an airbase into 'surge' state will shorten the required time delay used in arming and launching an aircraft.

USER INTERFACE

Player access to and interaction with the Airops subsystem starts with and revolves around the airplan. In principle an airplan is a schedule of a day's set of launches and recoveries for an airbase. Figure 1 shows a sample of the Define Aircraft input form. An airplan is uniquely identified via the base and the day. A day's airplan is organized into cycles with each cycle consisting of a set of events, or flights. A cycle represents the usual mode of flight operations on board an aircraft carrier where launches and recoveries are effected in a cyclic manner. The player defines the starting point for the day's airplan and the cycle duration to be used. With the exception of some special events, all launches and recoveries are scheduled according to the following rules:

1. All recoveries for cycle (n - 1) are completed prior to the start of launches for cycle n.

2. All launches for cycle n are completed prior to the start of recoveries for cycle (n + 1).

For each event in the airplan, the player enters the following data:

1. A unique identifying cycle/event number. This number consists of a two digit cycle number and a three character event number which in turn consists of a letter (from A - N) and a two digit number. All cycle/event numbers must be unique within a given airplan. This uniqueness is enforced by the USI software. The cycle number determines the (scheduled)

launch time for the flight. If the airplan has been defined to begin at time T with a cycle duration of C hours, then cycle N will be scheduled to launch at time

$$t = T + (N - 1) * C$$

2. Type of aircraft. A flight event is not limited to a single type of aircraft. Additional aircraft types may be scheduled for a flight via the 'More AC Types' overlay.

3. Number of aircraft.

4. Aircraft load. This is an optional field. The aircraft load specifies what ordnance or cargo the aircraft are to carry on their mission. Aircraft loads are defined for each type of aircraft.

5. Number of cycles. This field determines the mission duration. The scheduled recovery time, r, for a flight is

$$r = T + (N - 1 + M) * C$$

,where T, N, and C are as defined above and M is the number of cycles.

6. Alert Level. If the flight is not to be launched, but instead is to be 'on alert', this field specifies the delay that will be encountered when the launch is assigned. Three alert levels are defined for ENWGS - 5, 15, and 30 minutes.

7. Hotspin. Upon recovery, a flight is usually refueled and the aircraft returned to the mission capable (available for assignment) pool. If hotspin is indicated, however, the recovered flight is refueled, re-armed, and immediately launched.

8. Launch/Recover Out-of-Cycle. This field allows the player to launch and/or recover a flight at specific times other than those dictated via the airplan cycle structure.

9. Mission. The player has the option of assigning a mission to the flight from the following set:

a. CAP (Combat Air Patrol)
b. AEW (Airborne Early Warning)
c. ECM (Electronic Counter Measures)
d. Tanker
e. Strike
f. SUCAP (Surface CAP)
g. ASW (Anti-Submarine Warfare)
h. Minelaying
i. Minesweeping
j. Escort
k. Reconnaissance
l. Logistics

After initial airplan definition, the player has available an extensive set of reporting and editing functions. A request to report airplans will present the requestor with a list of currently defined airplans. Selection of one of these results event and by cycle, or an entire airplan may be purged. The processing software verifies that the requestor is entitled to perform the requested operation. In any case, only those events that have not proceeded through launch may be purged. Virtually any field of the airplan which pertains to a specific event may be changed. An exception is the cycle/event number itself. Most fields are modifiable up to the start of launch processing. Once aircraft have been put into alert status via the airplan form, they may be further manipulated via launch and upgrade forms. The Launch Alert Aircraft form provides the means of employing alert aircraft. A mission for these aircraft may be supplied at the time they are assigned to launch. In addition the player may specify hotspin at this time. Since the act of launching alert aircraft creates a new airplan event, the player must also specify the intended mission duration, either by supplying a specific recovery time or by number of cycles. The specific aircraft to be launched may be selected in one of three

```
                              DEFINE AIRPLAN

DIRECTIONS:  Enter AIRBASE ID, DATA, CYCLE START,  and CYCLE DURATION.
             Use LAUNCH/RECOVER OUT OF CYCLE if you want to deviate
             from cycle.

AIRBASE ID                          DATE:
DIVERT AIRBASE ID                   CYCLE START (HHMM)
                                    CYCLE DURATION (HHMM)

CYL/                     MORE                            LCH/RC  ON
EV   AC                  AC    MISSION  NBR  NBR  AC     OUT OF  ALERT HOT
NBR  TYPE                TYPES NAME     AC   CYLS LD     CYCLE   (5,15,SPIN
                         (Y/N)                           (Y/N)    30)
```

Figure 1.

ways. The simplest method is to launch an entire event by supplying the cycle/event number of an existing alert event. In this case, all of the aircraft of the specified events will be launched. Alternatively, aircraft type, number, and aircraft load may be given. In this case the system will find and assign the requested aircraft to the new event from existing alert events. Finally, the player may combine both sets of input in order to launch a subset of the aircraft from a specific event. The player will be advised if this directive cannot be acted upon; for example, in the case where there is an insufficient number of the specified aircraft available on the base.

AIROPS MODELS

There are several major models needed to simulate air operations. In discussing them, it will prove to be useful to consider the models from the point of view of a flight of aircraft scheduled for a mission from an airbase; that is, a simplified view is to consider a flight of aircraft as a GPSS transaction moving through the 'blocks' comprising the airops models.

Pre-flight Preparation

The first major model a flight encounters is pre-flight preparation. This encompasses the arming and fueling of the flight of one or more aircraft. As noted previously, detailed loading and fueling is not necessary - a time delay can be applied and the flight held in this preparation state until sufficient game time has passed for these actions. The actual time delay varies with number of aircraft in the flight, the number of aircraft capable of fueling/arming simultaneously, backlog from prior delays and other factors. For the simplest model level, this may be represented as

$$t = T*L_f *(0.5 + CEIL(N/S_{max})$$

,where T is the characteristic time needed to arm this type of aircraft, Lf is 0.5 if the aircraft is not to be armed and 1 if it carries a weapon load, N is the number of aircraft in the flight and Smax reflects the number of aircraft which can be prepped in parallel. This time factor is modified by values relating to weather, surge or post-surge base conditions, hotspin and number of other aircraft crowding the deck (density). The time delay may - depending on the nature of the particular ENWGS wargame being played - be varied randomly within predetermined limits. The pre-flight preparation model contains a large number of validation tests and advises, or alerts, the player of problems during preparation. A simple example is when the player has requested the launching of a flight of five F-14A aircraft and only four are available. Similar tests are done on weapon and fuel availability. Once a flight has completed pre-flight prep; that is, the appropriate weapon data structures and fuel levels have been set and linked to flight data structures and the time delay has been satisfied, then the flight moves into the launch model.

Launching

The launch model has the primary duty of setting up the data structures used to control the flight while it 'is in the air.' Once again, considerable validation is done as well as holding the flight for a time delay corresponding to the time needed to move each aircraft onto the catapult or runway, and launch the flight. As before, the time delay is influenced by the state of many dynamic variables. Should an airbase have four runways a certain time delay could be anticipated for a given number of aircraft (holding all other variables fixed); however, should three of these be damaged the time delay will be longer for the same flight. Weather factors, surge or post-surge conditions will influence the time delay, but the individual aircraft will not be considered as a discrete entity unless the entire flight is composed of a single aircraft. The launch model's final step - once the computed time delay has been reached by the game clock - is to call all the required routines to place the appropriate graphic symbol on the tactical display and inform the player that the flight has launched. As soon as the flight is 'airborne' the next model will be accessed - that of mission prosecution.

Mission Prosecution

The mission model encompasses many different lower-level models ranging in sophistication; the most basic model simply causes the flight to stay 'above' the airbase, the most complex involves the use the ENWGS Tactical Control Directives (TCD) - an expert system which will make decisions during mission prosecution depending on a set of rules. Mission types range from Combat Air Patrol (CAP) to Anti-submarine missions (ASW) to Strike missions. Each mission involves specific details and procedures which are simulated by a corresponding model and, as before, will generally last for either a specified time span or be terminated by some conditional event. That is, the mission could be defined to have a duration, for example, of three hours or could be terminated by running low on fuel. The player also, by means of the user system interface, has the option to terminate the mission by either taking direct control of the flight or by causing it to recover immediately. Dependent on the type of mission, the next process modeled by the Airops subsystem will usually be the recovery model. This involves computing flight time and direction to the designated landing base, causing the flight to move (updating the tactical screen with its location during the flight) and finally to recover (land) on the airbase. As always, a great deal of validation is incorporated within the model. It is quite possible for a flight of aircraft to intercept their recovery base and during the flight, the flight deck (or runways) become damaged or destroyed. In this case, the model may recover the flight to another airbase if one is available which can accommodate the type of aircraft; that is, a flight of jet aircraft cannot land on a base which has only helicopter pads, the flight deck of a carrier cannot recover aircraft which are not equipped with tail hooks. Options within the recovery model allow flights to hotspin as discussed under model control. Both the launch and recovery models consider the situation of a locked deck (density = 1) where the flight deck has so many aircraft that movement becomes impossible. Once again, surge conditions, weather factors, number of aircraft are accounted for by the time delay used in landing the flight of aircraft. Once recovery has completed for a flight, the next stage begins - that of the post-recovery turnaround.

Turnaround

Post-recovery turnaround is similar to the pre-flight preparation model with the additional requirement of a time delay associated with disarming the aircraft. In fact, as will be detailed in the section describing the model controllers, aircraft in post-recovery turnaround compete with those in pre- flight preparation. Most of these models are acting in parallel. Some flights are launching, some are recovering, others may be executing their scheduled missions and still others may be in post-recovery turnaround (competing for resources with aircraft being prepared for launch). It becomes a major effort to control the actions of all of these essentially simultaneous events. This is especially true in the ENWGS situation where a human operator is an integral part of the simulation. For example, all of the above actions may be taking place and the player decides to direct recovery of a given flight to a base other than its 'home' base; the sequencing and activities must be merged smoothly within the structure of the designated recovery base's existing airplan (if any). It is for this reason that a critical portion of this discussion describes the actions of the model controllers.

MODEL CONTROLLERS

The Airops control functions may be separated into three broad classes -airplan scheduling, mission and recovery scheduling, and queue maintenance. Airplan scheduling deals with the initial processing of an airplan and the orderly transitions from launch to recovery modes of airbase operations. Mission and recovery control functions provide the means to effect flight recovery at the required times. This functional area also provides the interface between the Airops subsystem and the mission prosecution models of ENWGS, including TCD's. Finally, queue maintenance handles the orderly entry and exit of aircraft to and from the several queues that are utilized between aircraft states.

Airplan Scheduling

Airplan scheduling processing is shared between two control modules: the flight ops monitor (FOM), and the airboss. In general the FOM determines what should be happening next and the airboss then performs the necessary control functions to effect the required processing, such as causing aircraft to move to and from queues and executing the required models. The FOM handles the following scheduling functions:

1. Initial airplan processing.

2. Initial scheduling of events added to an airplan subsequent to initial processing.

3. Initial processing of changes to previously defined events.

4. Effecting the transitions between launching and recovering base operations.

The airboss module is the primary driver of the actual models. It effects the following actions:

1. Initiation of next flight for pre-flight preparation.

2. Initiation of next flight for launch or recovery.

3. Initiation of next flight for post-recovery refuel for non-failed aircraft.

4. Initiation of next post-recovery refuel for failed aircraft.

The relationship between the entry points of these two controllers and the models can best be discussed with reference to Figure 2, which shows the primary prep/launch/recover/turnaround control flow.

The main entry point of FOM is activated upon completion of a Define Airplan Form. The primary function of this entry point is to determine when it will be necessary to start preparation of the aircraft for this cycle's entire set of launches. The prep entry point of FOM is then scheduled for this time. If there is insufficient time, preparation will start immediately. The prep entry point first puts each flight for the current cycle onto the prep list (queue). The prep_next entry point of airboss is then executed. This function of airboss removes flights from the prep list, one at a time, and then calls the prep model. One of the key features about carrier based flight operations which is represented in ENWGS is that there is a limit to the number of aircraft that can be worked on at any one time. This saturation parameter is an airbase characteristic. Before removing a flight from the prep list and initiating the model, it checks that the saturation level has not been reached. If this level has been reached, further prep list processing is discontinued. At the completion of each pre-flight preparation and post-recovery

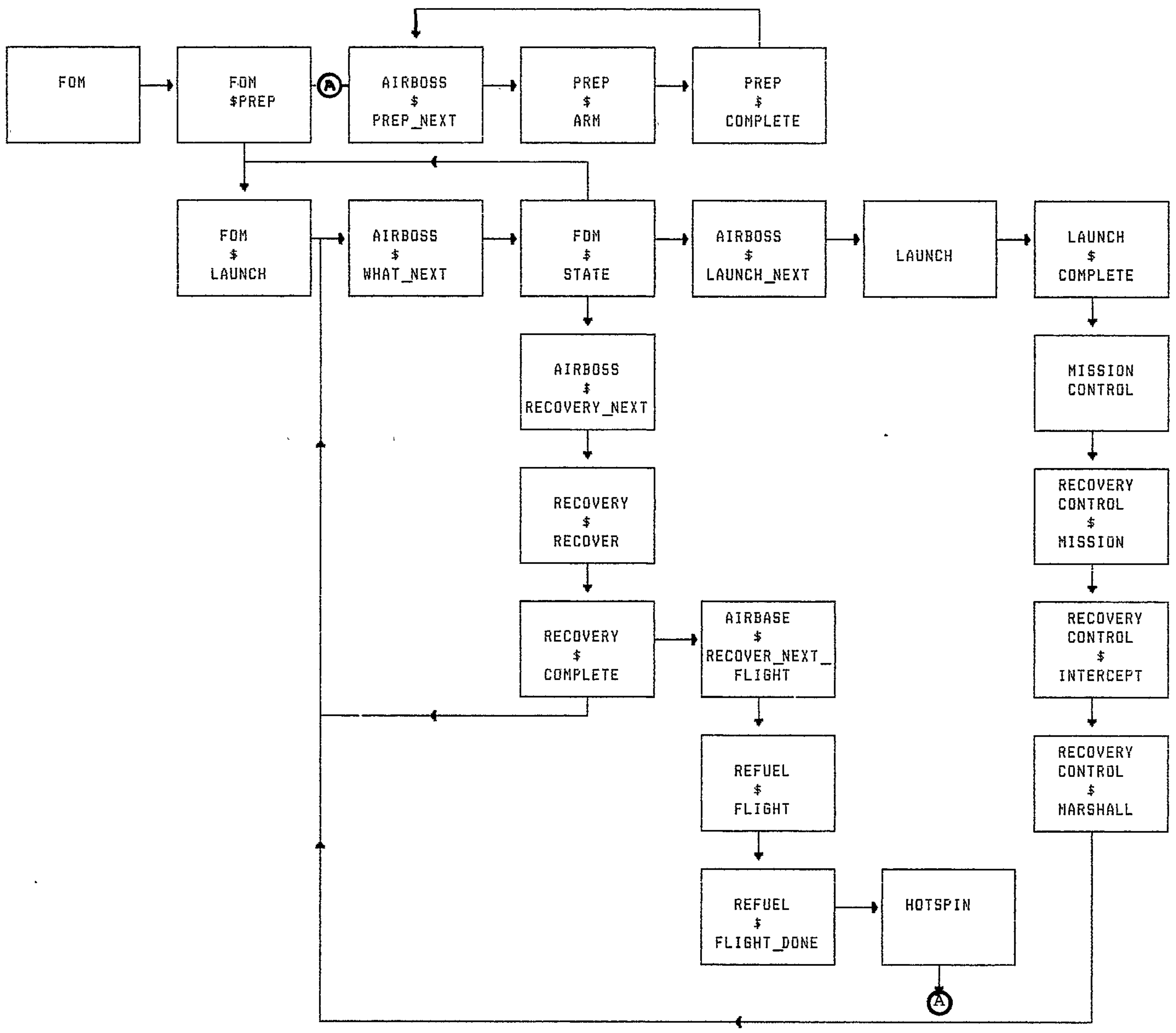
FOM
FOM $PREP
A
AIRBOSS $ PREP_NEXT
PREP $ ARM
PREP $ COMPLETE
FOM $ LAUNCH
AIRBOSS $ WHAT_NEXT
FOM $ STATE
AIRBOSS $ LAUNCH_NEXT
LAUNCH
LAUNCH $ COMPLETE
AIRBOSS $ RECOVERY_NEXT
MISSION CONTROL
RECOVERY $ RECOVER
RECOVERY CONTROL $ MISSION
RECOVERY $ COMPLETE
AIRBASE $ RECOVER_NEXT_FLIGHT
RECOVERY CONTROL $ INTERCEPT
REFUEL $ FLIGHT
RECOVERY CONTROL $ MARSHALL
REFUEL $ FLIGHT_DONE
HOTSPIN
A

Figure 2.

refueling (non-failed or failed), the prep_next entry point of airboss is again activated. It should be pointed out that any aircraft undergoing pre-flight preparation or refueling count towards the saturation level. When any of these activities is completed, it may have become possible for one or more new flights to enter one of these activities. Therefore, the complete phase of each of these respective models call the prep_next, refuel_next_flight, and refuel_next_down entry points of the airboss, in the indicated order. The order as shown implements a priority order to these activities, wherein preparation for a new flight takes precedence over turning around an old flight which takes precedence over refueling (previously) down aircraft.

After initiating the above preparation sequence, FOM schedules its launch routine for the cycle's launch time. Finally, the scheduling algorithm of FOM's main entry is used to schedule preparation activities for the next scheduled cycle.

The launch function of FOM initiates the sequence of launches for a given cycle. It establishes the identity of the current cycle, which is maintained as part of the airplan for use by the other models and controllers. If the base is not currently recovering aircraft, FOM sets the base operations state to launch and executes the what_next entry of the airboss. Otherwise, the what_next entry of airboss is executed without setting the operations state.

The what_next function of the airboss determines whether a launch or a recovery should be the next activity and initiates it. What makes this a non-trivial task is the necessary handling of priority events and late launches. Any launch of alert aircraft, the launch of a flight as a result of hotspin, and any flight scheduled as out-of-cycle is considered a priority launch. Such launches take precedence over all other ending launches and recoveries. At the conclusion of a priority event, the previous state of base operations is restored and the usual sequence of events proceeds. If priority events are not involved, the state function of FOM is invoked to determine the next activity and operations state. At this point, consideration is given to the handling of late launches. A late launch comes about because flights were backed up for pre-flight preparation due to saturation level limits. These flights therefore arrive on the launch list after the launch entry point of FOM has processed this queue. The FOM state function determines if a late launch may be necessary when it finds both the launch and recovery lists empty. This function examines each event in the current cycle to see if it is in prep or ready to launch. The earliest preparation complete time from the set found is then used to (re)schedule the FOM launch entry point, if it has not already been scheduled.

Mission and Recovery Control

Mission and recovery control functions are initiated by the launch model when it has determined that a flight has completed its launch processing. Although 12 different missions are available from the Define Airplan Form, they divide into three classes for modeling purposes. CAP, AEW, ECM, and Tanker missions are all considered local missions. These mission involve going to a point, loitering for a period of time, and returning to base. Strike missions are implemented via an interface to the engagement subsystem of ENWGS. All of the remaining missions are implemented as TCD's as discussed in Rubin and Buser (1988) and in Buser and Rubin (1988). Recovery control for these two latter types of missions is embedded in the mission processing itself, and is therefore beyond the scope of the present work. Local missions and their recovery, however, are controlled within the Airops subsystem. The mission control module shown in Figure 2 provides the interface to the other mission processors and also to the recovery control module for local missions.

Recovery control sequencing operates in several phases needed to account for a number of possible situations. Once a flight scheduled for a local (CAP, AEW, etc) mission has completed launch and is under the control of the mission-control model, one of the controller's first actions is to schedule an automatic recovery based on the scheduled recovery time for the flight. The scheduled auto recovery will compute the turnaround point based on flight speeds and the possibility that the recovery base is different from the launch base. This is needed to take into account the possibility that the flight will not be able to reach its assigned mission station. As an example, one could imagine that a aircraft assigned a CAP station 200 miles away from its launch base with a speed of 100 knots (helicopter) has - because of prep/launch delays - only one hour until recovery. Assuming that the recovery is to be at the launch base and the speed out is the same as the speed returning, then it is clear that the turnaround will occur in a half- hour after launch at a distance of fifty miles. Should the flight reach its station, the usual case, then the mission-control module will cancel the scheduled auto-recovery and recompute the time at which recovery is to be initiated. It does this by scheduling the execution of the mission entry point of the recovery control module and then continues to control the prosecution of the current mission.

When the game clock reaches the scheduled time and recovery control becomes active, the mission entry point tests to make sure the recovery base is still available and computes the distance and flight time required to reach it. The controller then schedules the activation of the next phase of the recovery process by scheduling the entry point intercept for the scheduled recovery time less the flight time. When intercept becomes active, its function is to compute the bearing to the recovery base and cause the flight to change speed and direction toward the base; setting up various conditional events and schedules the entry point marshall which is nominally a holding pattern above the recovery base. Once the marshalling point is reached, the marshall entry point verifies the continued availability of the base and calls the airboss recover_next entry - which in concert with what_next will cause recovery and eventual post-recovery turnaround.

Queue Maintenance

Transitions between states are handled via a queuing mechanism. The base manager module performs the actual adding and deleting of aircraft to the various queues. This approach 'hides' the details of the queue data structures from the individual models and controllers. Five queues are maintained within the Airops subsystem. They are:

1. Preparation - aircraft waiting to be prepared prior to a launch.

2. Launch - aircraft that have completed pre_flight preparation and are waiting to be launched.

3. Recovery - aircraft that have returned to the base and are waiting to land.

4. Refuel (flight) - aircraft awaiting post-recovery refueling.

5. Refuel (down) - aircraft that have been repaired from a post-recovery failure and are now awaiting refueling.

Each of these queues are implemented as linked lists and are handled as priority queues. The use of priority queuing is forced by hotspin, out-of- cycle launches and alert aircraft. Keeping in mind that due to delays in launch, prep, and recovery because of the dynamic nature of an ENWGS game any given airbase may be doing all of the activities described at the same time on the game clock. Hotspinning flights, for example, must be able to move to the front of the prep queue; scrambled alert aircraft must be able to access the runways/catapults without waiting for recovering or scheduled launching flights. Much of the complication of the controlling process is due to constraints of this nature, priority queuing, FOM and airboss must try to cope with all eventualities.

Other Data Structures

The priority queues described above are a very small part of a subsystem which is largely data driven; as noted, they are primarily for the purpose of controlling a sometimes complicated and hectic scheduling problem. Once the flight is 'airborne' a considerable number of other structures must be maintained to control as many of the possible conditions which might arise in the wargaming environment. The linked list is the basic workhorse structure, but most of the large number of lists are interlinked providing a network of information available to the gaming processes. Starting from the basic index of an airbase in its list, the airops components can derive flight schedules, weapon capability, fuel capacity and all other necessary data needed to obtain as much realism as is possible.

SUMMARY

In the preceding discussion, we have tried to show how a subsystem which models (primarily carrier) air operations is designed and meshes with a large, comprehensive wargaming system. Each of the four major components of the airops subsystem - models, model controllers, user interface and data structures have been described. In particular, the sequencing of activities from pre-flight preparation to launch through recovery and turnaround have been discussed in some detail to illustrate the interactions between these components and the surrounding 'universe' of ENWGS.

All of this activity is foreground; ENWGS background activities are still maintaining fleet manuevers, battle activity, communications, logistics and most other aspects of Naval wargaming. Although some aspects of real life carrier based air operations are absent from ENWGS, such as wind over the flight deck, we feel that the current subsystem does provide the ENWGS player with a level of realism that is pursuant to the objectives of various wargaming exercises.

ACKNOWLEDGEMENT

We wish to acknowledge our coworkers within the airops subsystem. While many individuals working on ENWGS in have contributed greatly to this effort, we would like to note in particular the efforts of Mona Freilich, Stephen Dowling and Robert Blacker.

The work reported in this paper was developed for the Department of the Navy, Space and Naval Warfare Systems Command, under Contract Number N00039- 84-C-0025.

REFERENCES

Buser, J. F. and Rubin, P. E. (1988). User considerations and their impact on an expert systems building tool for war gaming. In: AI Papers, 1988. (Ranjeet J. Uttamsingh, ed.) Simulation Series. Volume 20. Number 1. Society for Computer Simulation International. San Diego, California. 97-102.

Rubin, P. E. and Buser, J. F. (1988). Development of an expert system environment for use with a wargaming system. In: Artificial Intelligence and Simulation: The Diversity of Applications. (Troy Henson, ed.) Society for Computer Simulation International. San Diego, California. 155-162.

BIOGRAPHY

PAUL E. RUBIN holds a BS from Rensselaer Polytecnnic Institute, and an MS and Ph.D. from Drexel University, all in Physics. He is currently the head of the Build, Design and Integration group for tne Enhanced Naval Warfare Gaming System (ENWGS) at Computer Sciences Corporation. His professional interests include computer based wargamming, simulation, and applications of AI to such systems.

Computer Sciences Corporation
Defense Systems Division
304 West Route 38, Box N
Moorestown, NJ 08057

JOSEPH SOWERS holds a PH.D. in Physics from Temple University. He is a senior computer scientist for the Enhanced Naval Warfare Gaming System (ENWGS) at Computer Sciences Corporation. Prior to joining CSC in 1987, he was with Rutgers University for eighteen years. His interests lie in simulation of physical phenomenon, compiler design and computer languages.

Computer Sciences Corporation
Defense Systems Division
304 West Route 38, Box N
Moorestown, NJ 08057