AD-A084 830 ARMY COMBINED ARMS STUDIES AND ANALYSIS ACTIVITY FOR--ETC F/G 19/1
AMMUNITION RESUPPLY MODEL. VOLUME I. METHODOLOGY AND USERS MANU--ETC(U)
MAR 80 D J REMEN, R B CLARKE, J FOX

UNCLASSIFIED TR-2-80-VOL-1 NL

1 OF 1
AD
A084830

END
DATE
FILMED
6-80
DTIC

UNCLASSIFIED

MARCH 80

TR 2-80

ACN 36801

LEVEL II

6

ADA084830

AMMUNITION RESUPPLY MODEL METHODOLOGY AND USERS MANUAL VOLUME I

Technical Report TR 2-80

UNITED STATES ARMY COMBINED ARMS CENTER

COMBINED ARMS
STUDIES AND ANALYSIS ACTIVITY

Approved for public release
DISTRIBUTION UNLIMITED

STUDIES AND ANALYSIS DIRECTORATE

80 5 22 023

M008200 80—50—30 Apr 8C

UNCLASSIFIED

Technical Report TR 2-80
March 1980

Directorate of Studies and Analysis
US Army Combined Arms Studies and Analysis Activity
Fort Leavenworth, Kansas 66027

6

AMMUNITION RESUPPLY MODEL.

VOLUME I.

METHODOLOGY AND USERS MANUAL.

9 Final rept.

10

by
Mr. Donald J./Remen
MAJ Robert B./Clarke
Mr. James/Fox

11 Mar 80

ACN 36801

12 74

Approved by:

Robert T. Reed

Robert T. Reed
Colonel, Armor
Director

14 TR-2-80-VOL-1

411 772

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report TR 2-80 / 2. GOVT ACCESSION NO. AD-A084830	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Ammunition Resupply Model Volume I Methodology and Users Manual	5. TYPE OF REPORT & PERIOD COVERED Final
	6. PERFORMING ORG. REPORT NUMBER TR 2-80
7. AUTHOR(s) Mr Donald J. Remen, MAJ Robert B. Clarke, Mr James Fox	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE March 1980
	13. NUMBER OF PAGES 72
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report)
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

War Game
Jiffy Game
Ammunition Resupply
Force Structure Trade-off
Division 86

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report is one of two, produced to document the Ammunition Resupply Model (ARM). The model was designed to simulate those activities associated with ammunition resupply-demand, reload, resupply-in parallel with the play of the Jiffy war game in the evaluation of a division size force. The purpose of ARM is to assess the capability of a given TOE structure to respond to logistical demands placed upon it by various numbers of ammunition-expending weaponry. The other volume of the report is the Programmer's Manual which consists of the FORTRAN code of all the subroutines.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

ABSTRACT

This manual is one of a set of two, produced to document the Ammunition Resupply Model. As an outgrowth of the Division-86 study effort, the Ammunition Resupply Model (ARM) was designed to simulate those activities associated with ammunition resupply in parallel with the play of existing war games. Its purpose is to assess the capability of a given TOE structure to respond to the logistical demands placed upon it by various numbers of ammunition-expending weaponry. This report contains a discussion of model methodology, data base development, interface requirement with the war game, and the operators guide. The other volume of the report is the Programmer's Manual, which consists of descriptions, logic flow diagrams, and the FORTRAN code of all the programs and routines associated with the Ammunition Resupply Model.

FOREWORD

In general, existing logistics models tend to address resupply requirements in aggregated terms, such as tons per man per day or rounds per tube per day. Although this approach has considerable merit for evaluating large force structures engaged in sustained combat, it is inadequate for addressing logistics impacts of organizations engaged in short, intense conflict scenarios.

Ammunition expenditures emerging from high level (as opposed to high resolution) war games have traditionally been either unconstrained or based on a percentage of an "anticipated" daily resupply capability. Because of this, support analyses have not been the product of a concurrent logistics simulation utilizing the same scenario, but have been based on evaluations made after game completion. This method can paint a false picture of a combat unit's effectiveness. The logistics system, especially its ability to resupply critical commodities such as ammunition and fuel, must be evaluated during the course of the simulated battle.

The study directive for the Division-86 study called for a Force Structure Trade-off Analysis (FSTA) of various division alternatives. The tool for this FSTA effort was the Jiffy war game. To derive meaningful insights into the effects of the ammunition resupply assets contained in the different force structures and their impact on the combat effectiveness of the various units within the division, ammunition resupply must be evaluated in some detail. Such an evaluation must include simulating the time-consuming resupply process that places ammunition on individual weapon systems, as well as the movement of the different units' transportation assets to secure additional ammunition. It is this concept that provides the basis for the Ammunition Resupply Model (ARM), a concept that reflects the real-world factors that affect ammunition resupply. ARM was, therefore, developed to work in parallel with Jiffy in conducting a total FSTA of the Division-86 alternatives.

The concept for ARM was developed in Oct-Nov 1978, with the methodology and logic flow charts being completed in Dec 1978. The actual coding of the model was accomplished from Dec 1978 through Feb 1979, and the model was operational in May 1979. This report provides the detailed documentation of the methodology and operator procedures.

The authors of this report wish to acknowledge Harry Jones of the Model Design, Development and Validation Branch of COA for his assistance in programming several of the operating routines. Our thanks also to Mr. Ken Pickett, Dr. Dave Bash, and Mr. Harvey Taylor of Methodology and Quality Assurance Branch for their help in providing some initial file structure organization and programming logic flow charts. Special thanks are given to Mrs. Elizabeth Etheridge, who served as Technical Editor for this report, and the girls in the Word Processing Center East, who typed the report.

TABLE OF CONTENTS
Technical Report TR 2-80

TABLE OF CONTENTS (Cont'd)

TABLE OF CONTENTS (Cont'd)

LIST OF FIGURES

Ammunition Resupply Model
Technical Manual

1. INTRODUCTION.

a. Purpose. The purpose of this report is to provide a detailed description of the Ammunition Resupply Model (ARM). ARM was developed to operate in parallel with the JIFFY war game (but it is adaptable to most other combat models). ARM has been used with JIFFY, to study the ammunition resupply capability of alternative division organizations developed for evaluation in the Division 86 force structure tradeoff analysis (FSTA) efforts.

b. Scope. This report discusses the methodologies and data used in ARM along with the output (reports) generated. Included in this report as appendix A is the Operators Manual, which outlines the procedure to be followed by the model operator in the execution of ARM.

2. OVERVIEW.

a. General. ARM is an event oriented, time sequenced computer model developed to simulate the various functions associated with ammunition resupply from the Corps Storage Area (CSA) down to the individual weapon systems. Its purpose is to assess the capability of a given TOE structure to respond to the logistical demands placed upon it by various ammunition expenditures developed by JIFFY (and is easily adapted to any other model). It places these expenditures as demands on the resupply network. For a detailed description of JIFFY see references 1 and 2. ARM forces the network to replace rounds on individual weapon systems at unit level and send unit trucks back to designated resupply points to fill up and return. The functions each truck must perform are broken into a series of discrete events (subroutines). The model takes each truck through a series of these event subroutines (with operational availability and interdiction considered) in which actions are completed and times accumulated. Helicopter resupply, interactive command decisions, and tactical realism can be incorporated during the game.

b. Gamer Functions. The manual functions associated with ARM are the finalizing of the data base from acquired map data for each game played and the interactive operation of the console. The map-acquired data needed to finalize the data base are the locations of all units, the locations of the Ammunition Transfer Points (ATP) and Ammunition Supply Point (ASP), and the road distances from the units to the servicing ATP and ASP. The operating instructions are contained in appendix A.

c. Game Resolution. ARM is a high resolution game that is capable of playing a brigade or division size force. Through the development of a second data base it can play a second division, thus having application to corps level war games. Within the data base the unit resolution corresponds to that of JIFFY; that is, maneuver units are played at battalion level, artillery units at battery level, and aviation units at battalion or company level. Within the units, individual weapon systems are reloaded, and ammunition dedicated trucks deliver the ammunition to the various weapon systems and make runs to resupply points. A status of all units can be obtained at the end of a specified period of battle.

d. Stand Alone Model. ARM was developed as a stand alone model in order to retain the flexibility to support other attrition models. Ammunition resupply parallels the combat that generates demands. Resupply functions take place concurrently with combat and continue long after a particular battle has been fought. ARM is an event sequencing, time stepped model that schedules events to occur in future time. Consequently, it should not be integrated into the attrition model that ARM is supporting.

3. ASSUMPTIONS AND LIMITATIONS.

a. Maneuver units will have the opportunity to accomplish reload once during a specified period (4 hours) of combat.

b. Artillery units will reload when low on ammunition, which is likely to be once each hour during the battle.

c. Aviation units will reload upon the return of the aircraft to the FARP.

d. Air defense artillery units will reload once during a specified period (4 hours) of combat.

e. Ammunition trucks are dedicated to carrying specified types of ammunition. Limited dual loading takes place.

f. When a weapon system is lost all ammunition on the system is lost.

g. When a loaded truck is interdicted the load is lost.

h. Helicopter emergency resupply will support only 155mm artillery batteries.

i. Helicopter resupply will originate from the ASP.

j. The division slice of corps heavy lift helicopters will not exceed 10 CH47s.

k. The division slice of corps transportation assets for ammunition resupply will be one medium truck company of 60 tractors and 120 trailers. This company will provide ammunition through-put from the Corps Storage Area (CSA) to the ATPs and provide some replenishment to the ASP.

1. The model addresses the movement of ammunition from the Corps Storage Area (CSA) forward to the individual weapon systems.

4. DATA BASE DEVELOPMENT.

a. General. The data base is the nucleus of ARM. It contains a unit file that describes various attributes of each unit identified in the force file of the combat game. There are ATP and ASP files that describe stockage levels of ammunition and the operating characteristics of each. A truck file that describes each ammunition truck in the various units as well as the stake and platform (S&P) trucks of the corps medium truck company is a necessary ingredient. The data base also contains an ammunition file that identifies different truck load quantities by type of ammunition and the load times at the ATP or ASP for the respective quantity of rounds. The information to make up the data base came from the TOEs of the units played in the war game as well as from FM 101-10-1, the Ammunition Initiative Task Force (AITF) Report, the Ammunition Transfer Point (ATP) Test Report, and several TRADOC associated schools.

b. Definition and Description of Data Files.

(1) IUNIT File. The IUNIT file consists of a 75 by 69 array in which each unit is listed using up to 69 attributes to describe each. Most of the attributes are used to describe up to five types of ammunition that a unit might use.

(a) Attribute 1 identifies the type of unit. There are presently eight different unit codes as shown in figure 1.

1	Tank task force
2	Mech task force
3	Armed cav sqdn
4	155 arty btry
5	8 inch arty btry
6	GSRS btry
7	DIVAD gun btry
8	Cbt avn bn

Figure 1. Unit Type Codes

(b) Attributes 2 and 3 identify the servicing ATP and ASP of the unit, while attributes 4 and 5 provide the distance from the combat trains or assembly area to the ATP and ASP. Attribute 6 identifies the unit's position on the map, and attribute 7 contains the name of the unit such as TF4A as used in the attrition model. The name of the unit in ARM must be identical to that in the war game generating the expenditure data.

(c) Attributes 8 through 67 are used to describe the ammunition associated with the unit's major weapon systems. Attributes 8-19 describe the first ammunition type as follows:

1. Attribute 8 identifies the first ammunition type. The model can handle 20 different types of ammunition at present, those types listed in paragraph A-7-22 of Annex A-7.

2. Attribute 9 lists the number of weapons alive that use the ammunition identified in Attribute 8.

3. Attribute 10 shows the number of weapons that are short ammunition at the end of a CI.

4. Attribute 11 lists the number of rounds short of this type ammunition.

5. Attribute 12 shows the current ammunition supply on the weapons, which is equal to the basic load on the weapons alive minus the rounds short.

6. Attribute 13 identifies the routine resupply level; i.e., that level of ammunition stockage on the weapon at which resupply would be initiated given an opportunity to resupply.

7. Attribute 14 identifies the critical resupply level; i.e., that level of ammunition stockage on the weapon at which resupply must be initiated in order to sustain firing.

8. Attribute 15 is the basic ammunition level (per weapon). It lists the stockage of ammunition on the weapon system itself and/or the track carrier associated with an artillery tube.

9. Attribute 16 is the ammunition on trucks. It lists the number of rounds bulk loaded on the unit ammunition trucks. It represents the bulk loaded portion of the true basic load.

10. Attribute 17 is the number of weapons killed at the end of the critical incident (CI). This is an input from the war game each CI and is used to reduce the original number of weapon systems alive.

11. Attribute 18 is the number of weapons short ammunition during the CI. It is used to determine reload requirements per weapon based upon total rounds short.

12. Attribute 19 is the total rounds fired through the whole CI. It is an input from the war game each CI and is used as the demand on the resupply network.

(d) Attributes 20-67 describe the other four types of ammunition used by a particular unit. In describing a GSRS unit only attributes 8-19 would be used since only one type of ammunition is fired.

(e) Attribute 68 is the number of helicopters assigned. It applies only to 155mm artillery units and is used to determine whether or not a unit has received the maximum of two helicopter resupply missions this CI.

(f) The last attribute (69) contains a Ø or a 1. A Ø indicates that the unit receives a single pulse demand per CI; that is, reload of rounds fired is accomplished once during the CI. A 1 indicates that the unit receives multiple pulses during the CI. For example, for artillery units the total CI demand is divided in fourths, with 1/4th the demand being resupplied each hour. A sample of a unit file is shown in figure 2.

(2) ITRUCK File. The ITRUCK file consists of a 560 by 7 array, which allows the use of 560 trucks, each of which is described by seven words as follows:

(a) Attribute 1. Truck type - six truck types are used in the model as shown in figure 3.

Code		TRK TYPE
1	-	10 Ton
2	-	5 Ton
3	-	5 Ton w/1 1/2 T. Trl.
4	-	10 Ton w/15 T. Trl.
5	-	22 1/2 Ton Stake and Platform
6	-	Helicopter, CH 47

Figure 3. Truck Type Codes

(b) Attribute 2. Mission type - identifies the truck as a unit truck or a truck used on the CSA to ATP link, or ASP to ATP link. There are presently five mission type codes as listed in A-V-23 of annex A-V.

(c) Attribute 3. Status type - identifies the status of the truck based upon the status codes shown in figure 4.

CODE	DEFINITION
1	IN UNIT QUEUE
2	IN ATP QUEUE
3	IN ASP QUEUE
4	IN TRANSIT
5	UNIT TRUCK GOING FROM ATP TO ASP
6	TRUCK AWAITING REPAIR
7	TRUCK DEAD (INTERDICTED)

Figure 4. Truck Status Codes

****** UNIT DATA ******

UNIT	TYPE AMMO	ATP WPNS ALIVE	ASP WPNS	SHORT	ATP DIST RNDS	SHORT	ASP DIST CURFCHI	UTM BRL	GEL	DIFFY NAME CAL	TRK AMMO	NO. HELD	[illegible]	[illegible]
1	1	1	1		31		101			TF1A		0	0	
	1	44		0		0	2376	54	0	54	720	0	0	0
	2	6		0		0	72	12	0	12	108	0	0	0
	2	13		0		0	52	4	0	4	108	0	0	0
	11	6		0		0	498	83	0	83	560	0	0	0
	15	6		0		0	36	6	0	6	0	0	0	0
2	1	3	1		51		106			TF2A		0	0	
	1	44		0		0	2376	54	0	54	720	0	0	0
	2	6		0		0	72	12	0	12	108	0	0	0
	2	13		0		0	52	4	0	4	108	0	0	0
	11	6		0		0	498	83	0	83	560	0	0	0
	15	6		0		0	36	6	0	6	0	0	0	0
3	1	3	1		18		73			TF3A		0	0	
	1	44		0		0	2376	54	0	54	720	0	0	0
	2	6		0		0	72	12	0	12	108	0	0	0
	2	13		0		0	52	4	0	4	108	0	0	0
	11	6		0		0	498	83	0	83	560	0	0	0
	15	6		0		0	36	6	0	6	0	0	0	0
4	1	3	1		22		77			TF4A		0	0	
	1	44		0		0	2376	54	0	54	720	0	0	0
	2	6		0		0	72	12	0	12	108	0	0	0
	2	13		0		0	52	4	0	4	108	0	0	0
	11	6		0		0	498	83	0	83	560	0	0	0
	15	6		0		0	36	6	0	6	0	0	0	0
5	1	2	1		19		86			TF5A		0	0	
	1	44		0		0	2376	54	0	54	720	0	0	0
	2	6		0		0	72	12	0	12	108	0	0	0
	2	13		0		0	52	4	0	4	108	0	0	0
	11	6		0		0	498	83	0	83	560	0	0	0
	15	6		0		0	36	6	0	6	0	0	0	0
6	1	2	1		23		76			TF6A		0	0	
	1	44		0		0	2376	54	0	54	720	0	0	0
	2	6		0		0	72	12	0	12	108	0	0	0
	2	13		0		0	52	4	0	4	108	0	0	0
	11	6		0		0	498	83	0	83	560	0	0	0
	15	6		0		0	36	6	0	6	0	0	0	0
7	1	1	1		20		96			TF1B		0	0	
	1	14		0		0	756	54	0	54	240	0	0	0
	2	19		0		0	216	12	0	12	324	0	0	0
	2	42		0		0	150	4	0	4	324	0	0	0
	11	9		0		0	495	83	0	83	640	0	0	0
	15	9		0		0	40	6	0	6	0	0	0	0

Figure 2. IUNIT File

(d) Attribute 4. Owner number - the ARM number for a particular unit; e.g., 10, 33, 45.

(e) Attribute 5. Ammo mix number - the number that identifies the type of ammunition hauled by the truck (see figure 5).

(f) Attribute 6. Percent loaded - the percent of ammunition on the truck at a given point in time.

(g) Attribute 7. Time since last failure - contains the time since the truck last had a failure. It is used to determine the next time the truck will break down by subtracting from it all subsequent movement times. It is established for all trucks at the beginning of the game by multiplying the mean time between failure of each type of truck found in ITYPE by a random number between 0 and 1, thus distributing the time since the trucks were last repaired.

(3) ITYPE. This is a 6 by 6 array used to describe each of the six types of trucks as follows:

(a) Attribute 1. Secondary road night speed in km/hr - used in determining arrival time of a unit truck traveling to the ATP or ASP at night.

(b) Attribute 2. Secondary road day speed in km/hr - used in determining arrival time of a unit truck traveling to the ATP or ASP during the day.

(c) Attribute 3. Highway night speed in km/hr - used in calculating the arrival time of an S&P truck traveling at night.

(d) Attribute 4. Highway day speed in km/hr - the speed of an S&P truck on a highway during the day.

(e) Attribute 5. Mean-time-between-failure (MTBF) in minutes - used to address operational availability of the trucks.

(f) Attribute 6. Mean-time-to-repair (MTTR) in minutes - used to determine the time that a truck will return to duty once it has broken down.

(4) IRSTME. A 20 by 3 file that records resupply time data for 20 types of ammunition. Each ammunition type is described as follows:

(a) Attribute 1. Weapon set-up time - the time it takes a weapon system to prepare itself to take on ammunition once the ammunition truck arrives in the area of the weapon.

(b) Attribute 2. Load time per round the average time to uncase a round and store it on the weapon system.

(c) Attribute 3. Travel time to weapon - the one-way travel time from the truck assembly area or combat trains to the weapon system location. It is computed based on the approximate distance that the trucks are likely to be from the weapons systems they support and the travel speed of the truck. The travel speed is calculated as the average speed for 50 percent cross country and 50 percent secondary roads. All travel speeds were calculated based upon mobility data contained in Mobility Performance of 1/4- to 10- Ton Tactical Trucks and Cargo Carriers in the HIMO West Germany Study Area (TACV Study) produced by US Army Engineer Waterways Experiment Station, reference 3.

(5) IASP. This is a 4 by 41 word array that describes the Ammunition Supply Point. The model can handle four ASPs, each of which is identified by 33 attributes. The other attributes are currently empty. This file contains the current stockage of ammunition played in the model and the number of servers available to load unit trucks. There is a separate queue for GSRS trucks and one for all other trucks. Since GSRS trucks have their own self loader it is assumed that no assistance is required from the ASP crew. There is also an attribute that keeps track of the number of trucks in each queue. See paragraph A-V-5 of annex A-V for a description of each attribute.

(6) IATP. This is a 4 by 30 array that describes four separate ATPs. Each ATP is defined by the 30 attributes listed in paragraph A-V-3 of annex V. Under the ATP concept, it would handle high demand-high tonnage ammunition with 155 HE, 155 ICM, tank and TOW being the recommended stockage. In ARM the ATP can handle five types of ammunition, the four listed above and 155mm powder. The powder is handled separately since the S&P trucks from the CSA will be commodity loaded; i.e., only one type of load per truck, no cross loading. On the other hand, S&P trucks originating at the ASP could be combat loaded; that is, loading a trailer with two-thirds white bag and one-third green bag powder.

(7) IMIX. This is a 40 by 23 array that records information on 40 different combinations of ammunition mixes. For each of the 17 ammunition codes used in ARM the IMIX file contains one or more mix numbers. Each mix number represents the number of rounds of that type ammuniton that can be hauled on a particular type truck. For example, IMIX 22 represents a load of 96 rounds of 155mm ICM on a 5-ton truck with trailer. Attributes 22 and 23 of IMIX 22 provide the load time for this quantity of rounds at the ATP and ASP. A sample of the IMIX matrix is shown in figure 5.

(8) Other. Other files or data commons required are defined in annex A-V.

c. <u>Building of Data Files</u>.

(1) Main data files. Although editing of the data base can be accomplished through use of the main ARM program driver, it is desirable

****** AMBO DATA ******

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	460	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	50
2	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	70
3	0	46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	65
4	0	0	0	192	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45	70
5	0	0	0	0	192	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	60
6	0	0	0	0	0	96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	60
7	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	60
8	0	0	0	0	0	0	0	108	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60
9	0	0	0	0	0	0	0	0	108	0	0	0	0	0	0	0	0	0	0	0	0	0	60
10	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0	0	0	55
11	0	0	0	0	0	0	0	0	0	0	280	0	0	0	0	0	0	0	0	0	0	0	55
12	0	0	0	0	0	0	0	0	0	0	0	2800	0	0	0	0	0	0	0	0	0	0	55
13	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	55
14	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
15	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	55
16	449	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	55
17	0	144	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	55
18	0	0	0	440	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	55
19	0	0	0	0	440	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	55
20	0	0	768	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	55
21	0	0	0	96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	55
22	0	0	0	0	96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	55
23	159	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	55
24	0	0	0	0	0	0	0	0	0	0	240	0	0	0	0	0	0	0	0	0	0	0	55
25	0	0	0	60	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
26	0	0	0	0	0	0	0	0	0	0	160	0	0	0	0	0	0	0	0	0	0	0	55
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	55
28	0	0	0	0	0	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	0	0	55
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2000	0	0	0	0	0	55
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55

Figure 5. IMIX File

and more advantageous to build the initial data base through the use of a separate edit program called into play by the call program HJEDIT. To edit an existing data base or create a totally new one it is first necessary for the operator to be logged in on a CRT under a 100K password. The next step is to attach a version of HDARMDATA as T1. HDARMDATA is the permanent file name of the master data base. The complete sequence of events to be performed in order to edit the data base is as follows:

(a) Attach HDARMDATA as T1
    Attach, T1, HDARMDATA,ID=TRACOMD

(b) Call HJEDIT
    Call, HJEDIT,ID=TRACOMD

(c) CUE: EDIT DATA FILES? (Yes/No)

(d) ANS: "YES" or "Y"

(e) ENTER VARIABLE NAME --

(f) ANS: IUNIT 10 -- CUE: Enter the name of any one of the various files to be edited, such as IUNIT 10. This will allow the operator to make any changes to the attributes of IUNIT 10. For a list of all the data files and their attributes see annex V to appendix A.

(g) CUE: . .

(h) ANS: L 8 19

This lists the attributes 8 through 19 that describe the first ammunition type for unit 10.

(i) RES: IUNIT 10

ATTRIBUTE	8	1
ATTRIBUTE	9	36
ATTRIBUTE	10	0
ATTRIBUTE	11	0
ATTRIBUTE	12	1944
. . . .		
ATTRIBUTE	19	0

(j) CUE: . .

Any attribute can be changed by writing C for change, I an integer representing the attribute to be changed, and J an integer representing the new value. The entry would look like this: C 9 30, which means change attribute 9 to read 30. If the same values are to be entered into several units, it is possible to make these changes all at once by identifying all the units as follows: IUNIT 1, 10; Res: any changes made to an attribute will be recorded in units 1 through 10. More discussion on the commands that can be used in editing the data base can be found in annex I to appendix A. Having completed making all the required changes to the data base the word "end" is entered after the CUE: . . is given.

(k) CUE: . . END

(l) CUE: EDIT DATA FILE? (YES/NO)

(m) ANS: "N" or "NO"

(n) CUE: UPDATE ARRAYS?

(o) ANS: NO, Response NO is given here since it is assumed that all changes to the data base have been made. A YES response would allow a more rapid change of selected attributes over multiple variables.

RES: The edited data base is written to Tape 1 and the program terminates.

(p) The original T1 should be returned and TAPE 1 copied to a new local file T1 in preparation for the next operation to complete building the data base. TAPE 1 can be returned.

(2) Truck queue.

(a) General. After developing the data base is it necessary to assign all trucks to their respective queues (units). In the ITRUCK file of the data base the trucks were assigned owner numbers coinciding with the ARM IUNIT numbers. In order for the model to find these trucks it is necessary to put the trucks in their respective queues. All trucks assigned to owner 10 must be put into queue 10. Trucks assigned to an ATP, which have on board the initial stockage of ammunition at the ATP, must be put into the right queue at the various ATPs. Trucks that will be set in motion at the beginning of the game should not be placed in a queue. This is any truck assigned a status code 4.

(b) Operator routines.

1. Continuing from the previous operation, paragraph 4c(1)(0) where TAPE 1 was copied to a new T1, the next command is call HJTRKQUE.

2. CALL, HJTRKQUE,ID=TRACOMD

3. CUE: Modify Truck Queues? (Yes/No)

4. ANS: "Y" or "Yes"

5. ANS: Command Examples:

GET 3 From 35
Put 3, 10 in 105
List 105
Take all out
End

6. CUE: . .

7. ANS: Enter the commands to accomplish the task. If all the queues are empty then the operator can begin putting the various trucks in their respective queues; i.e.,

P 70, 79 in 10--This puts trucks 70 through 79 in queue 10, unit 10.

P 501, 512 in 101--This puts trucks 501-512 in queue 101 at ATP 1.

If an old data base is being modified to reflect a new organization it may be necessary to remove certain trucks from their old unit queues before they can be put in their new unit queues. Trucks must be removed from a queue one at a time (G 80 of 11) unless the operator enters "Take all out", which will initialize all queues to zero. After entering all the trucks in their respective queues, enter END.

8. CUE: . .

9. ANS: END

10. CUE: Print Out Contents of Queues? (Yes/No)

11. ANS: "YES" or "Y"

12. RES: Contents of all queues will be printed to output. A sample of the queue listing is shown in figure 6.

13. CUE: Modify Truck Queues? (Yes/No)

14. ANS: "NO" or "N"

15. CUE: Print Out Contents of Queues? (Yes/No)

16. ANS: "NO" or "N"

17. RES: New queue listing and all data base files are written to TAPE I and program terminates. TI should now be returned and TAPE 1 cataloged as the new Master Data Base for this version of ARM. Further discussion of the subroutine TRKPUT can be found in annex III of appendix A. The actual computer program for TRKQUE and associated subroutine can be found in Volume II, Programers Manual.

QUEUE 7 TRUCKS
50 51 52 53 54 55 56 57 58

QUEUE 8 TRUCKS
59 60 61 62 63 64 65 66 67

QUEUE 9 TRUCKS
68 69 70 71 72

QUEUE 10 TRUCKS
73 74 75 76 77 78 79 80 81 82
83 84 85 86 87 88

QUEUE 11 TRUCKS
89 90 91 92 93 94 95 96 97 98
99 100 101 102 103 104

QUEUE 12 TRUCKS
105 106 107 108 109 110

QUEUE 13 TRUCKS
111 112 113 114 115 116

QUEUE 14 TRUCKS
117 118 119 120 121 122

QUEUE 15 TRUCKS
123 124 125 126 127 128

QUEUE 16 TRUCKS
129 130 131 132 133 134

QUEUE 17 TRUCKS
135 136 137 138 139

QUEUE 18 TRUCKS
140 141 142 143 144

QUEUE 19 TRUCKS
145 146 147 148 149 150 151 152 153 154
155

QUEUE 20 TRUCKS
156 157 158 159 160 161 162 163 164 165
166

QUEUE 21 TRUCKS
167 168 169 170 171 172 173 174 175 176
177

FIGURE 6. Truck Queues

5. METHODOLOGY.

a. General. The general concept for the ARM simulation is based on a description of the ammunition resupply policies within a division along with the ammunition through-put provided by the corps support units. A review of these policies reveals a series of discrete events (see figure 7). Ammunition expenditures during a battle generate a demand for more ammunition. The ammunition trucks of the various units execute reload activities to replenish expended rounds on surviving weapons. When unit trucks become empty, they are sent to a rear ammunition resupply point for another load. The ammunition resupply point must have its stockage replenished in order to continually support the combat units. Essentially, ARM processes the ammunition expenditure developed by a war game and places this expenditure as a demand on the resupply network. It forces the network to replace rounds expended at unit level by requiring unit ammunition trucks to reload the surviving weapon systems and, when empty, to go to a designated resupply point, fill, and return. The model takes each truck through a series of discrete events (subroutines) (operational availability and interdiction considered) in which actions are completed and times are accumulated.

b. Major Events. The ammunition resupply network was broken down into four major events, each with a number of subroutines. The major events are demand, reload, resupply, and replenishment. The connecting links among the last three events are the ammunition trucks of the various units and their movements. Figure 8 is a graphic view of the event subroutines and should be referred to while studying the description of each subroutine.

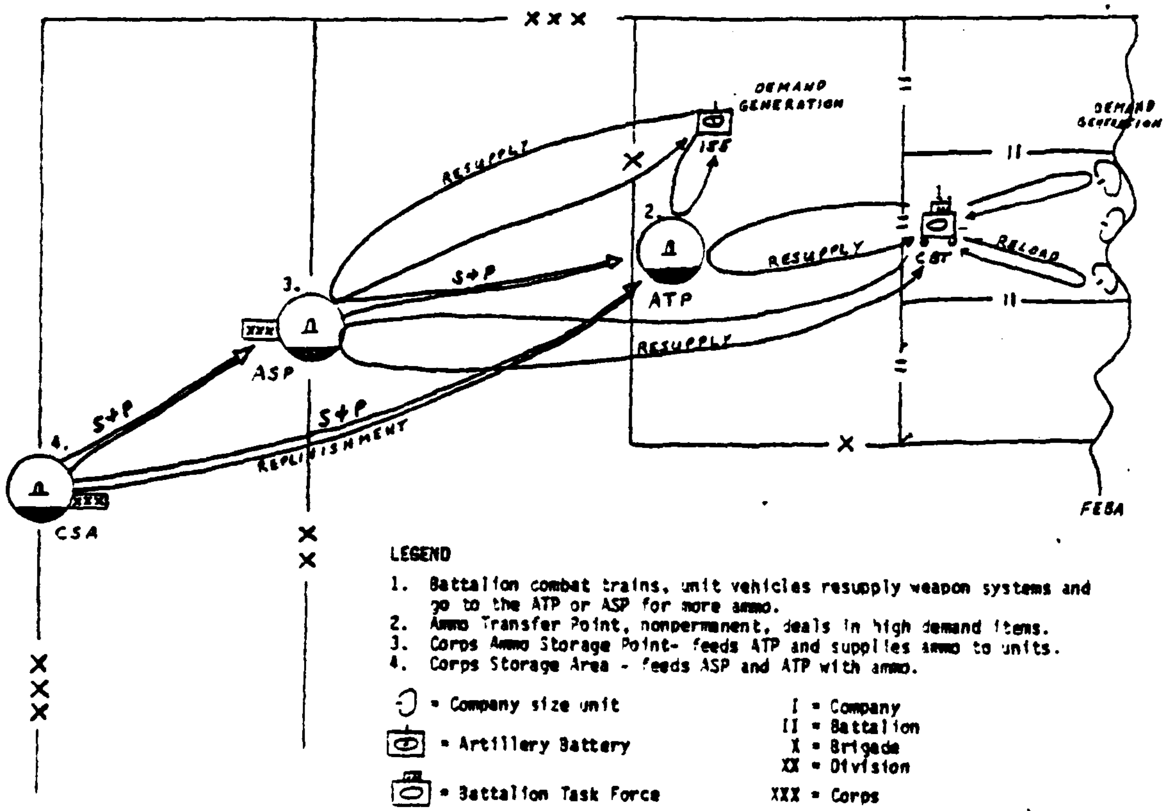
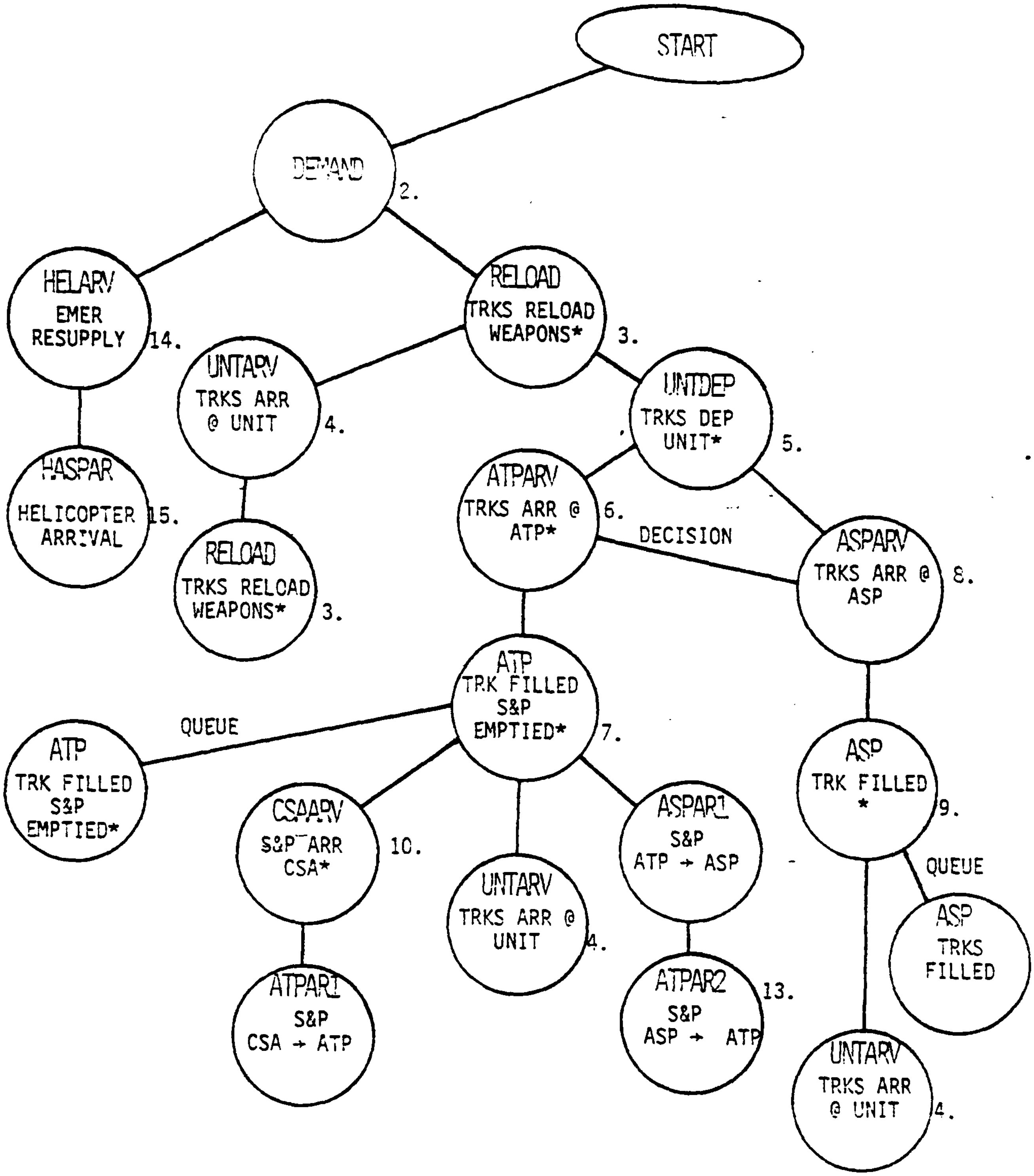


Figure 7. Ammunition Resupply Network



*OPERATIONAL AVAILABILITY AND INTERDICTION CHECKED

Figure 8. Model Event Sequence.

c. Demand. The demand for ARM is provided as an input to the model from some opposing forces war game. The input lists by unit the number of each weapon systems remaining alive, the number of survivers that actually fired, and the total rounds of ammunition fired by the survivers. Rounds fired by systems that were killed have been subtracted out. Within ARM the subroutine DEMAND performs the function of taking any unsatisfied demand of a unit (demand that has not been satisfied when the DEMAND event is called) and combining it with newly generated demand. If the unit is artillery, the total demand for each type of ammunition is divided by H to reflect the expenditure during each hour of the H hour critical incident of the war game for which ARM was developed. The subroutine then scans the trucks at the unit to find one with the right ammunition. In the particular case of 155mm artillery units, DEMAND compares the actual demand (expenditure) against the sum of the current supply and the ammo-on-trucks. If the difference is unusually high (e.g., no trucks at the unit and current supply low) it is compared against the critical resupply level. If the critical resupply level has been reached, DEMAND will schedule an emergency reload event by helicopter (HELARV). If a truck was found with the correct ammunition on board a RELOAD event is scheduled. Upon completion of the DEMAND event, the unit's demand will have been updated and a reload event scheduled. For artillery units the DEMAND event will reschedule itself to occur again 60 minutes from the present time.

d. Reload. For a given unit for which reload has been scheduled the subroutine RELOAD performs the actual replacement of rounds of ammunition expended by rounds carried on unit trucks. First a type of ammunition is selected, then the unit queue is searched for a truck hauling that type of ammunition. If a truck is found, the following calculations are made: (1) the number of weapon systems that can be reloaded by this truck - a function of the demand and the truck's load, (2) the total reload time, and (3) the return time to the combat trains or truck assembly area. The reload time is calculated by the following equation:

$$Rtime_{ijk} = 2 * Trvtme_k + w(A_i + B_j\ (\#Rds/Wpn))$$

where $Rtime_{ijk}$ = the time required to complete reload for weapons i with round j at unit k.

$TRVTME_K$ = travel time from combat trains or truck assembly areas to the weapon positions

W = number of weapons that can be loaded by a truck (depends on truck load)

A_i = set up time per weapon i (time for weapon to prepare itself to take on ammunition)

B_j = reload time per round (obtained from IRSTME file)

Having completed reload of the weapons the truck is scheduled to return to the combat trains or assembly area if there are rounds on board or it is scheduled to go pick up another load. Therefore, RELOAD will schedule a unit arrival (UNTARV), returning the truck to the unit to wait for another demand, or a unit departure (UNTDEP), sending it after another load.

e. Unit Arrival (UNTARV). The subroutine UNTARV brings the truck back to the unit combat trains, in the case of maneuver units, or assembly areas for artillery units. Upon arrival the ammo-on-truck, an element of the unit status, is updated. Since this event is scheduled from other subroutines a check is made for unsatisfied demand of the particular type of ammunition carried on the truck. If there is an unsatisfied demand, a RELOAD event is immediately scheduled; otherwise, the truck waits for another RELOAD event to occur.

f. Unit Departure (UNTDEP). If upon completion of a reload event a truck is empty, a unit departure (UNTDEP) event is scheduled. This subroutine checks the type of ammunition the truck was carrying against the four types of ammunition stocked at the ATP. If a match in ammunition codes is found, an arrival time at the servicing ATP is calculated and an ATP arrival (ATPARV) event scheduled. The arrival time is based upon the distance from the unit to the ATP and the average speed of the truck. If no match in ammunition codes is made then an arrival time at the ASP is calculated and an ASP arrival (ASPARV) event scheduled.

g. Operational Availability. Every time a truck moves, a check is made of its operational availability in subroutine OPERA. Each truck has its own clock, which keeps track of the hours of operation since it last failed. At the start of the game all trucks are initialized with a time since last failure. This initialization is the product of the mean-time-between-failure and a random number between 0 and 1. Each time a truck moves, the time length of the move is subtracted from the time remaining until the next failure. When the time remaining becomes zero the truck status is changed and its movement delayed by the mean-time-to-repair for the particular type of truck. OPERA is called from any event subroutine that involves truck movement.

h. Interdiction. The interdiction subroutine (INTRDK) determines whether a truck about to execute a move will be interdicted and, if so, assesses a time delay for providing a replacement truck. For interdiction purposes, the combat zone is subdivided into two zones. Zone one extends from the line of contact to the brigade rear boundary. Zone two consists of the area from the brigade rear boundary to the corps storage area. Unit trucks forward of the ATP are considered to be in zone one, where they are susceptible to being hit by artillery fire. Unit trucks moving from the

ATP to the ASP and all S&P type trucks are considered to move in zone two, where they are subject to attacks by aircraft. The attrition model provides the number of trucks killed by artillery fire during the battle being simulated. In order to determine which trucks are interdicted it is first necessary to take the total number of trucks killed, as given by the attrition model, and multiply it by the percentage of all trucks that carry ammunition. This number is then entered into a data file as the total number of zone one trucks to be interdicted this battle period. In order to spread this number over as many units as possible, another number between 15 and 30 is selected at random, which is used as a controlled cycle number that we will call y. Each time a truck is scheduled for move it is sent through INTRDK, where a counter is maintained. When the yth truck enters INTRDK it is the one interdicted, and a time delay is assessed before a replacement truck can be provided. The counter is then reset to 0 and started again. This procedure is continued until the total number of trucks that were to be interdicted has been reached. This method comes very close to representing reality since one would expect that units that fire often are more susceptible to receiving counterfire and therefore lose more trucks. In ARM the units that fire more require the unit trucks to move more frequently. The number of trucks to be interdicted in zone two are selected by the military gamer. The number is usually less than the number of trucks interdicted in zone one.

i. ATP Arrival. The ATP arrival (ATPARV) subroutine (#6 in figure 3) checks the ammunition mix number of the arriving truck and the quantity of rounds needed. It then checks the quantity of that ammunition on hand against the total demand for that type of ammunition. The demand is determined by checking the number of trucks in the queue waiting for that particular type of ammunition. If there is sufficient ammunition and if the ammunition type is one of artillery, a further check is made of the total artillery demand against the powder on hand. If there is insufficient ammunition or powder to provide the truck with a load once it reaches the head of the line, it is sent to the ASP. Otherwise, the truck waits to be served. The first truck to arrive generates an ATP event, other trucks arriving wait for the next available server.

j. ATP Event. The ATP event subroutine (#7 in figure 3) simulates the activities that take place at the ATP. The arrival of a unit truck triggers the first ATP event; subsequent ATP events are scheduled automatically at the end of the reload time for that truck. The simulation begins with one active server for each queue (maneuver and artillery). Provisions exist for an idle server in one queue to assist the other, and for activating an additional server should one or both queues become lengthy. Initially, the ATP subroutine removes a unit truck from a queue, determines the desired type of ammunition, then searches the queues of replenishment S&P trucks to find a truck with the right ammunition and the least amount. A match having been found, the unit truck is loaded, and the

S&P trailer load is decremented by the number of rounds removed. If the unit truck needed artillery ammunition then a check would have been made of the amount of powder available. After the projectiles are loaded the powder truck would be decremented the same number of rounds. The unit truck is then scheduled for a unit arrival (UNTARV) having successfully passed operational availability and interdiction. When the S&P trailers become empty they are sent back to either the CSA or ASP depending upon their initial origin. Thus, the ATP subroutine schedules arrivals at the CSA, ASP, or unit, and reschedules itself.

k. ASP Arrival. The ASP arrival (ASPARV) subroutine (#8 in figure 8) places an arriving truck into one of two queues, the GSRS queue or the "all other" queue. Since the GSRS trucks have a self-loading capability it is assumed that they can load themselves at the ASP, thereby allowing all the servers assigned to the ASP to handle other resupply requirements. The arriving truck is put in the proper queue. As servers become available trucks are removed from the queue by the ASP event.

l. ASP Event. The ASP event (ASP) subroutine removes a truck from the queue, determines the type of ammunition required, calculates the load time as a function of the ammunition type, and schedules a unit arrival for the truck. Again, operational failure and interdiction are checked before the unit arrival time is calculated. As long as there are idle servers, trucks are removed from the queue for servicing, thus simulating concurrent service of as many trucks as there are servers. Upon reloading a truck the current stockage is updated. Each ASP event schedules a subsequent ASP event for the next truck in the queue.

m. CSA Arrival and ASP Arrival 1. ARM replenishes ATP stocks from both the CSA and ASP using basically the same methodology. These subroutines simulate the arrival at either place of an S&P tractor, either alone or with an empty trailer, and the subsequent dropping of the empty trailer and picking up of a full trailer load of ammunition. It is assumed that the full trailer has already been loaded and all the tractor has to do is take on some fuel and hook on to the trailer. Therefore, a short reload time is used. Both subroutines subsequently schedule the arrival of this truck at the proper ATP, having checked for operational failure and interdiction. The rounds removed are added to a counter of total rounds removed from both the CSA and ASP. The CSAARV subroutine can also schedule S&P trucks to arrive at the ASP to simulate replenishment of the ASP stock.

n. ATP arrival 1 and ATP Arrival 2. These two subroutines place the S&P trucks in two separate queues at the ATP, one for trucks coming from the CSA and the other for trucks from the ASP. It also updates the current supply of ammunition at the ATP and changes the truck status to being in the queue at the ATP.

o. Helicopter Arrival. The helicopter arrival (HELARV) subroutine is scheduled from DEMAND whenever the current supply of HE or ICM ammunition at a 155mm artillery battery becomes less than or equal to the critical resupply level. HELARV simulates the pickup of a mixed load of HE and ICM ammunition at the ASP by a CH47 helicopter and the subsequent delivery to the designated battery. When the event actually takes place, the current supply of each of the two ammunition types at the battery is updated, showing the receipt of the ammunition. It also decrements the number of helicopters serving the unit and schedules the helicopter's arrival back at the ASP.

p. Helicopter ASP Arrival. This subroutine (HASPAR) brings the helicopter back to the ASP and updates the number of helicopters available for resupply missions. The helicopters are subject to operational failure in the same manner as the trucks, but are not interdicted.

q. Initialization. Initialization consists of those actions taken by the console operator to put ARM into operation. A detailed description of procedures to be followed can be found in Appendix A, ARM Operators Guide.

6. TYPES OF OUTPUT. At the completion of a specified period of simulated combat, ARM is designed to provide a complete status report on the ammunition resupply network of the division. This is accomplished by means of an audit trail of all events and a number of reports generated by the subroutine called REPORTS. From the REPORTS subroutine it is possible to select one or all of the following reports:

1. TRUCK STATUS
2. UNIT STATUS
3. SINGLE ATP REPORT
4. ASP REPORT
5. CSA REPORT
6. MULTIPLE ATP REPORT
7. AMMO REMOVED FROM ASP
8. TRUCKS THAT HAVE BEEN KILLED OR BROKEN

a. The truck report provides a complete status of each truck. It identifies the owning unit, specifies its location; i.e., on the road or at the unit, identifies the type of ammunition being carried and percentage of the total load on board, and specifies the next time the truck will fail. The truck report is normally printed in conjunction with the unit status reports. A sample of a truck status report is shown in figure 9 along with the parent unit.

b. The unit report provides the current status of each unit, reflecting unit identification, servicing ATP and ASP, and the respective distances to the ATP and ASP. Additionally, it identifies each ammunition type used by the unit. For each type it specifies the associated weapon,

UNIT STATUS — UNIT DATA — WPN DATA

C12FA
SER ATP 2 12 KM
SER ASP 1 72 KM
NO HELO 0

AMMO-CODE	WPN-TYP	WPN-ALIVE	CUR-SUP	RNDS-SHORT	PCBL-W	ON-TRKS	NO WPN SH / NO SH EA		TOT-DMD
4	155HE	5	195	0	100	0	0 /	0	248
5	155ICM	5	254	181	58	0	5 /	36	505
6	155SMK	5	35	0	100	0	5 /	0	11
7	155CLGP	8	56	0	100	96	0 /	0	0

TRUCK STATUS REPORT FOR UNIT C12FA

TRK	NM	STATUS	MIX	PCLOAD	NXFAIL
246	1	1	4	47	3527
247	1	5	4	0	3149
248	1	4	4	0	3572
249	1	7	5	54	4717
250	1	1	5	0	4518
251	1	4	5	0	1604
252	1	4	5	100	476
253	1	5	5	0	3080
254	1	1	6	56	1530
255	1	1	7	100	2380

KEY:

o Unit Data. Column heading definitions can be found in the description of the unit data file.

o Truck Status Report
- TRK - Truck Number
- NM - Mission
- Status - See Figure 4
- Mix - Type of ammunition hauled
- PCLOAD - Percent of load on truck
- NXFAIL - Next time the truck will fail.

o Weapon Data
- No WPN SH - Number of WPNS Short Ammo
- No SHEA - Number of rds Short per WPN
- TOT-DMD Total Demand

Figure 9. Unit Status Report

the number of surviving weapons, the current supply of ammunition on the weapons, number of rounds short, percentage of basic load on the weapons, and the number of rounds of ammunition on the trucks located at the unit. Also specified for each ammunition type is the total demand for the past period of combat. A sample of a unit report is also shown in figure 9.

c. The single ATP report provides information on each ATP such as number of servers (forklifts) active in each queue and the number of trucks waiting to be serviced. It also specifies for each ammunition type handled by the ATP, the current demand, the amount on hand, and what the initial stockage was. A sample of an ATP report is shown in figure 10.

ATP STATUS

ATP NO 3

QUEUE	ARTY	MU
SERVERS ACTIVE	1	1
TRUCKS IN Q	1	2

AMMO-CODE	CUR-DMD	AMT-O/H	BASIC-LVL
1	96	320	448
2	36	144	144
3	160	1344	1536
4	160	1920	1320
5	0	1200	1320

Figure 10. ATP Report.

d. The ASP report specifies the number of active servers (forklifts and cranes), the number of trucks waiting to be loaded, and the current supply of each type of ammunition.

e. The CSA report only registers the number of rounds of ammunition drawn out of the CSA since the beginning of the war game.

f. The multiple ATP report and the report on ammunition removed from the ASP are similiar to the single ATP report and the CSA report.

g. The last report (Type 8) lists each truck that has been interdicted or is being repaired. The report identifies the owner of the truck, what type it is, and the type of ammunition it was carrying (see figure 11).

```
TRUCK NUMB   8 OF UNIT 1F1A       WHICH IS TYPE    1 CARRYING AMMN  11 IS DEAD
TRUCK NUMB 139 OF UNIT 11ACR      WHICH IS TYPE    2 CARRYING AMMN   3 IS DEAD
TRUCK NUMB 148 OF UNIT 11ACR      WHICH IS TYPE    2 CARRYING AMMN  23 IS DEAD
TRUCK NUMB 183 OF UNIT FA11ACR    WHICH IS TYPE    1 CARRYING AMMN   5 IS DEAD
TRUCK NUMB 192 OF UNIT FA21ACR    WHICH IS TYPE    1 CARRYING AMMN   5 IS DEAD
TRUCK NUMB 194 OF UNIT FA21ACR    WHICH IS TYPE    1 CARRYING AMMN   6 IS DEAD
TRUCK NUMB 202 OF UNIT A11FA      WHICH IS TYPE    1 CARRYING AMMN   5 IS DEAD
TRUCK NUMB 216 OF UNIT C11FA      WHICH IS TYPE    1 CARRYING AMMN   4 IS DEAD
TRUCK NUMB 223 OF UNIT C11FA      WHICH IS TYPE    1 CARRYING AMMN   5 IS DEAD
TRUCK NUMB 229 OF UNIT A12FA      WHICH IS TYPE    1 CARRYING AMMN   5 IS DEAD
TRUCK NUMB 233 OF UNIT A12FA      WHICH IS TYPE    1 CARRYING AMMN   5 IS DEAD
TRUCK NUMB 238 OF UNIT B12FA      WHICH IS TYPE    1 CARRYING AMMN   4 IS DEAD
TRUCK NUMB 249 OF UNIT C12FA      WHICH IS TYPE    1 CARRYING AMMN   5 IS DEAD
TRUCK NUMB 264 OF UNIT A13FA      WHICH IS TYPE    1 CARRYING AMMN   6 IS DEAD
TRUCK NUMB 266 OF UNIT B13FA      WHICH IS TYPE    1 CARRYING AMMN   4 IS DEAD
TRUCK NUMB 270 OF UNIT B13FA      WHICH IS TYPE    1 CARRYING AMMN   5 IS DEAD
TRUCK NUMB 271 OF UNIT B13FA      WHICH IS TYPE    1 CARRYING AMMN   5 IS DEAD
TRUCK NUMB 284 OF UNIT C13FA      WHICH IS TYPE    1 CARRYING AMMN   6 IS DEAD
TRUCK NUMB 294 OF UNIT A14FA      WHICH IS TYPE    1 CARRYING AMMN   9 IS DEAD
TRUCK NUMB 304 OF UNIT B14FA      WHICH IS TYPE    1 CARRYING AMMN   9 IS DEAD
TRUCK NUMB 316 OF UNIT D14FA      WHICH IS TYPE    4 CARRYING AMMN  10 IS DEAD
TRUCK NUMB 359 OF UNIT C22FA      WHICH IS TYPE    1 CARRYING AMMN   9 IS DEAD
TRUCK NUMB 360 OF UNIT C22FA      WHICH IS TYPE    1 CARRYING AMMN   9 IS DEAD
TRUCK NUMB 369 OF UNIT A21FA      WHICH IS TYPE    1 CARRYING AMMN   9 IS DEAD
TRUCK NUMB 379 OF UNIT C21FA      WHICH IS TYPE    1 CARRYING AMMN   8 IS DEAD
TRUCK NUMB 500 OF UNIT
*** WHICH IS TYPE     5 CARRYING AMMN  19 IS DEAD
TRUCK NUMB 524 OF UNIT
*** WHICH IS TYPE     5 CARRYING AMMN  16 IS DEAD
TRUCK NUMB 528 OF UNIT
*** WHICH IS TYPE     5 CARRYING AMMN  18 IS DEAD
TRUCK NUMB 534 OF UNIT
*** WHICH IS TYPE     5 CARRYING AMMN  20 IS DEAD
TRUCK NUMB   2 OF UNIT 1F1A       WHICH IS TYPE   1CARRYING AMMO MIX   1 IS BEING REPAIRED
TRUCK NUMB 137 OF UNIT 11ACR      WHICH IS TYPE   2CARRYING AMMO MIX   3 IS BEING REPAIRED
TRUCK NUMB 166 OF UNIT 21ACR      WHICH IS TYPE   2CARRYING AMMO MIX  23 IS BEING REPAIRED
TRUCK NUMB 167 OF UNIT 21ACR      WHICH IS TYPE   2CARRYING AMMO MIX  23 IS BEING REPAIRED
TRUCK NUMB 191 OF UNIT FA21ACR    WHICH IS TYPE   1CARRYING AMMO MIX   5 IS BEING REPAIRED
TRUCK NUMB 220 OF UNIT C11FA      WHICH IS TYPE   1CARRYING AMMO MIX   5 IS BEING REPAIRED
TRUCK NUMB 227 OF UNIT A12FA      WHICH IS TYPE   1CARRYING AMMO MIX   4 IS BEING REPAIRED
TRUCK NUMB 269 OF UNIT B13FA      WHICH IS TYPE   1CARRYING AMMO MIX   5 IS BEING REPAIRED
TRUCK NUMB 272 OF UNIT D13FA      WHICH IS TYPE   1CARRYING AMMO MIX   5 IS BEING REPAIRED
TRUCK NUMB 323 OF UNIT D14FA      WHICH IS TYPE   4CARRYING AMMO MIX  10 IS BEING REPAIRED
TRUCK NUMB 332 OF UNIT A22FA      WHICH IS TYPE   1CARRYING AMMO MIX   8 IS BEING REPAIRED
TRUCK NUMB 350 OF UNIT B22FA      WHICH IS TYPE   1CARRYING AMMO MIX   9 IS BEING REPAIRED
TRUCK NUMB 357 OF UNIT C22FA      WHICH IS TYPE   1CARRYING AMMO MIX   9 IS BEING REPAIRED
TRUCK NUMB 363 OF UNIT A21FA      WHICH IS TYPE   1CARRYING AMMO MIX   8 IS BEING REPAIRED
TRUCK NUMB 377 OF UNIT B21FA      WHICH IS TYPE   1CARRYING AMMO MIX   9 IS BEING REPAIRED
TRUCK NUMB 388 OF UNIT D22FA      WHICH IS TYPE   4CARRYING AMMO MIX  10 IS BEING REPAIRED
TRUCK NUMB 475 OF UNIT NON-UNIT   WHICH IS TYPE   5CARRYING AMMO MIX  18 IS BEING REPAIRED
TRUCK NUMB 477 OF UNIT NON-UNIT   WHICH IS TYPE   5CARRYING AMMO MIX  19 IS BEING REPAIRED
TRUCK NUMB 481 OF UNIT NON-UNIT   WHICH IS TYPE   5CARRYING AMMO MIX  18 IS BEING REPAIRED
TRUCK NUMB 482 OF UNIT NON-UNIT   WHICH IS TYPE   5CARRYING AMMO MIX  18 IS BEING REPAIRED
TRUCK NUMB 483 OF UNIT NON-UNIT   WHICH IS TYPE   5CARRYING AMMO MIX  18 IS BEING REPAIRED
```

Figure 11. Truck Killed or Under Repair

7. WAR GAME INTERFACE.

a. ARM was designed to adapt to nearly any attrition based war game (of any size up to division). It merely requires the ammunition demand in terms of rounds fired by type and the number of weapon systems that fired. Both of these values must be distributed among the units being gamed. For some studies, it may be better to develop these demands based upon some particular scenario, then to use them as constant inputs to ARM, performing variations in, for example, perhaps, the TOE truck structure or method of resupply. It is important, however, to illustrate the method by which a typical war game could be modified to produce the desired information. This paragraph shows the specific calculations that were entered into the coding of the JIFFY war game (versions I and II) to accommodate the requirements of ARM.

b. JIFFY is a computer assisted interactive war game, which takes manual map-play input, compares the forces that opposing force gamers commit against each other, and determines attrition of weapons by source, FEBA movements, and expenditures of ammunition.

(1) The overall committed Blue force, whether it be a company or a corps (or anything in between), is divided into sectors such that each sector bounds a particular confrontation. Sectors are artificial boundaries used to isolate particular portions of the battle that possess a certain set of environmental conditions. Factors used to define a sector are unit tactical boundaries, terrain, defensive or offensive tactic, etc.

(2) Finally, the gamers define the time segments they wish to play. Each segment is called a Critical Incident (CI) and is used to define a particular battle or confrontation of forces. Usually, gamers select CIs to be 4 hours (of battle-time) long. A night CI may last 6 or 8 hours.

(3) For each Critical Incident, each sector of the battle is played. Hence, when the game begins, CI1, sector 1 is played; then CI1, sector 2, etc., until all sectors have been completed. The play of each sector begins as the opposing gamers input a series of "command decisions," which further define the tactics and environmental conditions of the sector. Once these values have been entered into the computer, the battle within each sector is determined by a series of computer subroutines, which focus on a particular function. For example, armor-antiarmor, indirect fire, air defense-helicopter, and dismounted infantry are some of the functions played. Essentially, each computer subroutine takes the environmental and tactical input, counts the number of Red and Blue weapon systems involved in the specific battle function, degrades the firepower capability of those systems accordingly, and computes the number of weapon systems on each force killed and the number of rounds fired to accomplish those kills. All killer-victim score tables, unit equipment strengths, and other similar statistics are adjusted. The gamers evaluate these statistics, make

adjustments to their forces in position and perhaps tactics, then commence with the next CI.

(4) Properly assessing the ammunition resupply problem requires a knowledge of the number of weapons that actually fire, the number of those that survive (obviously a "killed" weapon does not need to be resupplied), and the number of rounds fired by those survivors. The remainder of this paragraph deals with the specific calculations needed within JIFFY to feed ARM its demand information.

c. Because of the internal logic and analysis used to develop the JIFFY game, it was necessary to categorize two different type of weapon systems for the demand calculations.

(1) Category A type weapons are defined as those that act somewhat independently, such as tanks, TOWs, DIVAD guns, etc. These weapons, although part of a precisely defined unit (for firepower and maneuver control), will engage targets individually. Most of their engagements are determined within the armor-antiarmor routine of JIFFY, enabling the demand calculation to be performed for each in a similar fashion.

(2) Category B type weapons are defined as those that fire as part of a group of weapons, such as 155mm or 8 inch artillery, mortars, and GSRS. These systems are handled in JIFFY within the indirect fire routine and are treated uniformly; that is, if the gamer has committed several 155mm artillery batteries into one sector, and during a particular critical incident he decides to fire his 155mm artillery in that sector at a certain level (rounds per hour per tube), then the JIFFY indirect fire subroutine will force each tube to fire at the same rate (with the same kind of ammunition). This subsequently makes the demand calculation for ARM simple because each weapon system of Category B is treated identically (within a sector).

(3) Demand calculations for Category A type weapon systems are handled as follows.

(a) Since nearly all calculations for Category A weapons are performed within the JIFFY Armor-Antiarmor (AA) subroutine, it is important to understand one major portion of the AA logic. Once the gamers have elected to play the AA routine for a certain sector, they must begin by choosing a range that will separate the opposing forces when, theoretically, the first hostile rounds are fired. Options include 500-meter gradations from 500 to 3500 meters. After this selection is made, the gamers each input the percent of the maneuver forces located in the sector that are to be involved (percent committed) in the first engagement. The computer then assesses the firers, rounds fired, and losses to both sides. After this assessment, the gamers appraise the results and select the next engagement range, and the next cycle is performed similarly. For example, the gamers may select four successive ranges: 2500m, 2000m, 1500m, 1500m, which suggests that the attacking force began to exchange fire at 2500 meters, closed to 1500 meters, and after a lengthy "firefight" at the closer range, broke contact to regroup.

(b) Throughout the following calculations, each range separation selected is referred to as a 'situation.' In the above example, there are four situations in which firers, targets, kills, and rounds fired are assessed. The following terms are defined:

S = number of battle situations played in JIFFY.

NW_{ij} = the total number of Blue weapons of type i in the force array (by sector) at the beginning of situation j.

$LOSS_{ij}$ = the total number of Blue weapons of type i killed by the Red force during situation j.

$NSURV_{ij}$ = the total number of Blue weapons of type i that survived situation j.

$$NSURV_{ij} = NW_{ij} - LOSS_{ij}$$

RND_i = total number of rounds fired by Blue weapons i during the entire CI (at all targets).

Other factors particular to weapons and taken directly from the JIFFY methodology are:

OA_i = operational availability of Blue weapon i

SF_i = suppression of Blue weapon i

ECF_i = expected number of completed firings for each Blue weapon i

Some general scenario factors (not weapon dependent) are:

$PCOM_j$ = percent of Blue force committed during situation j

SMK_j = degradation of Blue force due to smoke during situation j

(c) The initial thrust must be to determine two important probabilities: the probability that a weapon fires (prob($FIRE_{ij}$)) during a particular situation, and the probability that a weapon survives a particular CI (prob ($SURV_i$)). The number of firers (by type weapon i) in a particular situation j can be calculated as:

$$NWF_{ij} = NW_{ij} * OA_i * SF_i * PCOM_j * SMK_j \qquad (2)$$

Assuming a Bernoulli situation, then, a specific weapon can have two conditions: either it is a firer or it is not. Hence, the probability that a weapon fires, taken from the sample data, is the relationship of the number of firers to the number of systems (of type i) that could fire (or the number of weapon systems committed (in situation j)).
First, the number of systems committed is portrayed by equation (3).

$$NCOM_{ij} = NW_{ij} * PCOM_j \tag{3}$$

The probability during situation j that any weapon would fire (of those committed) is:

$$\text{Prob }(FIRE_{ij}) = \frac{NWF_{ij}}{NCOM_{ij}} \tag{4}$$

This, of course, must be multiplied by the probability that this circumstance would happen, which is simply the probability that the weapon would be in the committed force, or present committed. Since $PCOM_{ij} = NCOM_{ij}/NW_{ij}$ and prob (Commitment) = PCOM, this factor is multiplied by prob $(FIRE_{ij})$ to obtain the overall (total) probability of fire:

$$\text{Prob }(TFIRE_{ij}) = \frac{NWF_{ij}}{NW_{ij}} \tag{5}$$

Equations (2) and (3) are then substituted into equation (5):

$$\text{Prob }(TFIRE_{ij}) = \frac{NCOM_{ij} * OA_i * SF_i * SMK_j}{NW_{ij}} \tag{6}$$

Finally, the probability that a particular weapon fires during the <u>entire</u> CI is determined by combining the probabilities for each individual situation. For each situation, the probability that a weapon is not a firer is determined by:

$$\text{Prob } (\text{NTFIRE}_{ij}) = 1\text{-Prob } (\text{TFIRE}_{ij})$$

Then, assuming independence of situations, the probability a weapon did not fire throughout the CI is the product of the above individual probabilities:

$$\text{Prob } (\text{NTFIRE}_i) = \prod_{j=1}^{S} 1\text{-prob}(\text{TFIRE}_{ij}) \qquad (8)$$

Then, the probability that a weapon fired during the entire CI can be expressed as shown in equation (9) below:

$$\text{Prob } (\text{TFIRE}_i) = 1\text{-prob } (\text{NTFIRE}_i) \qquad (9)$$

(d) The probability of survival, however, need not be determined by situation. It may be calculated with data that pertains to the entire CI. Prob (SURV_i) is defined as the probability that weapon i will survive a given CI. Then:

$$\text{Prob } (\text{SURV}_i) = \frac{\text{NW}_{iB} - \sum_{\text{TOT}} \text{KILL}_i}{\text{NW}_{iB}} \qquad (10)$$

where B indicates the beginning of the CI.

This procedure will permit the probability of survival to account for kills by all sources as well as the armor-antiarmor assessment.

(e) To calculate the probability that a particular weapon is a "firing survivor" (i.e., the weapon remain alive at the end of the CI <u>and</u> had expended rounds), one further assumption is made:

that the events of firing and surviving are independent. This is not a fair assumption, since the act of firing creates a firing signature, which decreases the firer's chance of survival. However, considering the resolution played in JIFFY and the fact that determining the necessary correlation coefficients is nearly impossible with the data available, this assumption of independence is made to obtain a figure as realistic as possible within the data constraints. Therefore, the probability that a weapon is a firing survivor is calculated with equation (11) below:

$$\text{Prob}\ (FS_i) = \text{Prob}\ (TFIRE_i) * \text{Prob}\ (SURV_i). \tag{11}$$

Similarly, after determining the probability that a weapon is killed during the CI (Prob $(KILL_i)$ as $1\text{-}P(SURV_i)$, the probability that a weapon i fired <u>and</u> was killed is given by:

$$\text{Prob}\ (FK_i) = \text{Prob}\ (TFIRE_i) * \text{Prob}\ (KILL_i) \tag{12}$$

This equation, along with equation (11), will be used later to determine the number of rounds expended by the survivors and those killed.

(f) The final step, then, is to calculate the <u>expected number</u> of firing survivors. Since this circumstance can be described by the binomial family, the expected value can be expressed by the number of elements subject to the probability times the probability itself, or

$$E\ (FS_i) = \text{Prob}\ (FS_i) * NW_i \tag{13}$$

(g) By sector, JIFFY produces the total number of rounds fired by type weapon during the entire CI. However, it does not break this number down into those fired by the surviving weapons and those fired by the weapons that were killed. Taking the total number of rounds fired by type calculated within JIFFY, and summing for all situations, RND_i is defined as the total number of rounds fired during the CI by weapon i. Since $E(FS_i)$ and $E\ (FK_i)$, the data calculated similarly to equation (13), combine to fire the entire number, and assuming that kills occur uniformly throughout the CI, the number fired by the survivors can be obtained as a simple ratio. Let RFS_i be the number of rounds fired by the firing survivors, and let RFW_i be the number of rounds fired by each weapon of type i. Then:

$$RFW_i = \frac{RND_i}{E\ (FS_i) + E\ \frac{(FK_i)}{2}} \tag{14}$$

Hence: $$RFS_i = E\ (FS_i) * RFW_i \qquad (15)$$

(b) For Category A units, the following calculations must be completed:

1. Equation (13) must be calculated for each type weapon in each unit (utilizing equations appearing in the development).

2. The results of equation (13) must be apportioned to each Category A unit in the sector in direct proportion to the density of weapons of that type in each unit and its combat intensity level.

3. Once the number of firing survivors has been apportioned, the number of rounds expended must be calculated using equations (13) and (14).

(4) Demand calculations for Category B type weapon systems are developed as follows:

(a) Category B units, as defined in paragraph 7C, possess a much simpler solution concerning the demand data preparation. Category A units (particularly because of the Jiffy methodology for determining firers, etc.) required calculation on the actual number of weapon systems that expended rounds. In the case of, say, an artillery battery, resupply is done at the battery level to all firing weapons simultaneously (or continuously as would be the real-world case). Jiffy methodology, however, fires all weapons within an artillery battalion equally. Although operational availability, smoke or EW, and commander's percent employment may restrict the guns from firing continuously, when one fires, they all fire.

1. The following terms are defined:

N_i = number of weapons of type i in the force (by sector) at the beginning of the CI.

$TBAT_i$ = number of tubes of weapon type i in each battery

$KILL_i$ = number of tubes of weapon type i killed in each battery

RND_{ij} = number of rounds of type j fired by weapon type i during the CI.

2. With Category B weapons, the assumption is made that if any firing was done during the CI by weapon type i, the probability that any single weapon of that type fired is 1. Hence, the only concern is with the probability of survival which, as before, is the relationship of those of weapon i that survive a given CI to the total of those that started the CI. Since ARM bases its resupply cycle on demand (as opposed to resupply to a basic load), the only purpose in carrying the weapon survival status

for artillery is simply to adjust the number of rounds expended. Similarly, as with Category A units, the number of rounds fired by weapons that survive a particular CI must be determined since "dead" weapons will not be resupplied. Hence, let RFS_{ij} = the number of rounds of type j fired by <u>surviving</u> weapons of i , and let RFW_{ij} be the number of rounds fired by each weapon of round type j, then:

$$RFW_{ij} = \frac{RND_{ij}}{(N_i - KILL_i) + \frac{KILL_i}{2}} \tag{16}$$

and, therefore: $RFS_{ij} = RFW_{ij} * (N_i - KILL_i)$ (17)

Since JIFFY plays artillery by battery, and ARM likewise plays resupply at battery level, the data requirements are consistent.

(b) For Category B units, the following calculations must be completed.

1. The number of surviving weapons of type i for each battery of artillery (per sector). This result was already produced by the existing JIFFY methodology, so no alteration was required.

2. The rounds expended by weapon type i of ammunition type j (per sector). For simplicity purposes, since the data in (1) above and provided by sector, the ammunition consumption can be presented as a sector. expenditure.

d. This subparagraph describes the format in which the demand data must be presented to be compatible with ARM. Figure 2 below is a sample data printout, which illustrates the form and substance of the demand information. In this case, the data originate within the JIFFY FORTRAN coding in the manner previously described and and stored in an array. When printed, the data appear in the format shown in figure 12, and ARM reads the data sequentially using the same format shown.

a.	b.	c.	d.	e.	f.
TF1A	0.000	1.000	29.100	13.489	29.7128
	0.000	2.000	2.900	1.382	.593
	0.000	25.000	10.400	5.563	2.089
	0.000	11.000	3.200	3.200	94.614
	0.000	15.000	4.200	0.000	0.000
312FA	0.000	4.000	6.000	6.000	303.121
	0.000	6.000	6.000	6.000	13.988
	0.000	5.000	6.000	6.000	615.428
	0.000	0.000	0.000	0.000	0.000
	0.000	0.000	0.000	0.000	0.000
1ACAS	4.000	13.000	3.537	3.537	16.478
	4.000	13.000	3.814	3.814	14.748
	4.000	13.000	3.737	3.787	19.255
	4.000	13.000	3.414	3.414	13.179
	0.000	0.000	0.000	0.000	0.000

Figure 12. Jiffy Demand Data

A description of each column of figure 12 follows:

(1) Column a, the unit name. This name corresponds directly in spelling to the unit name known to JIFFY (loaded by the gamers). It must also correspond to the unit name internal to ARM.

(2) Column b, number of surviving helicopter sorties (number of surviving helicopters times number of sorties each). JIFFY calculates the number of TOW (or similar missile) firings based upon the number of helicopter sorties flown, consequently, ARM has been programmed similarly. Essentially, JIFFY assumes that a helicopter will fire all of its rounds before returning from a sortie. If the helicopter survives a sortie, it must be reloaded. This logic does not fit directly to Category A or B weapons, so it is treated separately.

(3) Column c, ammunition code number. This code corresponds to the ARM ammo codes listed in Chapter IV, paragraph 3.

(4) Column d, number of weapons in this unit alive (which fire the ammunition indicated) at the end of the CI.

(5) Column e, number of firing survivors of the weapon indicated in column d. This number indicates, of those weapons that survive the CI (column d), the number that expended ammunition.

(6) Column f, rounds fired. This column displays the total number of rounds fired by the firing survivors (this number is most likely less than the total rounds fired in JIFFY because some rounds were probably expended by killed weapons).

e. The assumptions and equations described in this paragraph are presented to illustrate a technique that can be used to adjust any attrition-based war game to produce demands for ARM. Although JIFFY logic allowed a somewhat straightforward development, some models may require a more detailed series of calculations, while some may require none at all. And, as explained earlier, ARM may be used in a stand-alone fashion without an adjoining war game. In that case, the demands could be manufactured in a format similar to that of figure 12, and these values could be read directly into ARM, thereby permitting the analyst to manipulate the demand levels as he desires.

REFERENCES

1. USACACDA, TR3-77, CACDA Jiffy War Game Programers Manual, 1 March 1977.

2. USACACDA, TR4-77, CACDA Jiffy War Game Users Manual, 1 March 1977.

3. US Army Engineer Waterways Experiment Station, HIMC West Germany Study Area (TACV Study).

4. BDM Corporation, (S) Ammunition Initiative Task Force (AITF) (U), Main Report, McLean, Virginia, October 1977.

5. BDM Corporation, (C) Ammunition Initiative Task Force (AITF) (Supporting Documentation (U), McLean, Virginia, October 1977.

6. USAMMCS, Munitions System Support Structure (MS3) Volume 1, Redstone Arsenal, Alabama, April 1978.

7. US Army, FM 101-10-1 w/change 1, February 1978.

APPENDIX A

ARM Operators Guide

A-1. PURPOSE. The following is a set of procedures to enable use of the Ammunition Resupply Model (ARM). The procedures assume that the operator is logged in on a CRT type terminal with a 150K octal password. Attach the required files for ARM to execute.

ATTACH,T1,HDARMDATA,ID=TRACOMD (beginning data file)

ATTACH,T2,HDARMEVENTS,ID=TRACOMD (empty event file)

ATTACH,T3, (local file name), ID = (permanent file name) (JIFFY created input file)

The next statement accesses a call file, which has the job control statements required to process the ARM.

CALL,HJARMANOTHER,ID=TRACOMD

A-2. METHODOLOGY. During the processing of the ARM, a number of messages will be written to the CRT that provide information and/or require operator response. These messages will be referred to herein as CUEs. The operator response, if required, will be referred to herein as ANS, with RES being the result of the operator reply.

a. CUE: RETAIN EVENTS CURRENTLY SCHEDULED (YES/NO)

ANS: N or NO

RES: Zeros out event list

b. CUE: ARE YOU CREATING AN ANSWER FILE (Y or N)

ANS: Y or YES

RES: Will bypass METRO produced input file to enable data base building and the creation of future events, goes to e below.

ANS: N or NO

RES: Regular logic will be followed.

c. CUE: ENTER TIME TO STOP SIMULATION

ANS: 240.1

RES: The ARM will process the critical incident (CI) until time equals 240.1 minutes.

d. CUE: INITIALIZE TRUCKS' TIME SINCE LAST FAILURE? (YES/NO)

ANS: YES or Y

RES: Trucks are loaded with time since last failure occurred by use of a uniform random number between 0.0 and 1.0 multiplied by the mean time between failure (MTBF) for each different type of truck.

e. CUE: (Known as the control menu.)

(1) - EDIT DATA

(2) - WRITE REPORT

(3) - SCHEDULE CONTROL

(4) - RETURN

(5) - STOP SIMULATION NOW

(6) - EDIT TRUCK QUEUES

(7) - CREATE EVENTS

f. ALTERNATIVE 1

ANS: 1

RES: Gives control to the edit subroutine, which enables modification and/or listing of common based data. See annex I for details. For the first CI enter variable name LUOUT.

CUE: VARIABLE NAME.

ANS: LUOUT

CUE: . .

ANS: C 1 6. This transfers the output to tape 6, the output tape, required only at the beginning of the game.

CUE: . .

ANS: Look

ANS: C 1 1 -- Required each time a new start is made as it reverts back to 0 when the simulation terminates.

ANS: INTER

ANS: C 3 ( ) Number of trucks to be interdicted.

ANS: C 7 ( ) The integer to be used as the cycle number; e.g., 22 means that every 22d truck to move in the simulation will be interdicted.

CUE: . .

ANS: END

RES: Returns to control menu, e above.

Since an answer file or history file is being developed the next action is to create the events that will force feed the ATPs with additional ammunition. This involves scheduling the arrival of S&P trucks at the ASP and CSA to pick up new loads of ammunition for delivery to the ATPs.

ANS: 7

CUE: TO CREATE AN EVENT, INPUT AS A GROUP SEPARATED BY COMMAS OR SPACES THE FOLLOWING 7 Values: EVENT TYPE (INTEGER VALUES BETWEEN 1 AND 17) 5 PARAMETERS FOR EACH EVENT (INTEGER, DEPEND ON EVENT TYPE) AND TIME (DECIMAL MINUTES, REAL) EXAMPLE: 9, 1, 508, 1, 0, 65. CSA-TO-ATP TRUCK 508 WILL ARRIVE AT CSA AT TIME = 65 MIN.

EXPLANATION OF 9, 1, 508, 1, 0, 65.
9 = EVENT TYPE 9, ARRIVAL OF CSA TRUCK AT CSA
1 = ATP NO. TO WHICH TRUCK IS GOING
508 = TRUCK NO.
1 = ASP NO.
0 = ZERO, no value, but required.
65. = Time in minutes when event is to occur. The decimal point is required.

ANS: 9, 1, 508, 1, 0, 65.

CUE: ?

ANS: 12, 1, 512, 1, 0, 45.

CUE: ?

ANS: END - after last event has been scheduled

RES: Returns to control menu, e above. If creation of the answers file is complete, then it is necessary to save the new events and new data base. Therefore, it is necessary to stop the simulation without proceeding any further.

ANS: 5

RES: Schedules a stop simulation event at the present time.

CUE: Stop

CUE: COMMAND

RES: Processing is complete, new files available are:

LOCAL FILE NUMBER	Definition
T1	New data base
T2	New events file containing the events just created.

An option available at this time is the cataloging of T1 and T2 to form the answer files, the basis for a restart at present time without going through the time consuming process of creating the events again. Once the files have been cataloged the input data must be reconnected.

```
ATTACH, T3, (local file name), ID = (          )--
             JIFFY created input file.
CALL, HJARMANOTHER, ID = TRACOMD
```

CUE: RETAIN EVENTS CURRENTLY SCHEDULED (YES/NO)

ANS: YES of Y

RES: Retains events currently scheduled. Return to paragraph A-2, b for continuation of the ARM logic.

g. ALTERNATIVE 2

ANS: 2

RES: Gives control to report subroutine. Details at Annex II.

h. ALTERNATIVE 3

ANS: 3

RES: Prints the following CUE

CUE: ENTER TIME OF NEXT CONTROL EVENT

ANS: 240.05

RES: At time 240.05 minutes ARM will return to the control routine.

i. <u>ALTERNATIVE 4</u>

ANS: 4

RES: Returns to main driver routines to continue processing.

j. <u>ALTERNATIVE 5</u>

ANS: 5

RES: Schedules a stop simulation event with the present time.

k. <u>ALTERNATIVE 6</u>

ANS: 6

RES: Gives control to subroutine TRKPUT, which enables modification of truck queue assignments. Examples are at Annex III.

l. <u>ALTERNATIVE 7</u>

ANS: 7

RES: Gives control to subroutine CREEVT, which enables the creation of any type event for later processing. See Annex IV.

Normally after exercising one or more of the alternatives, alternative 4 is chosen to start the processing of the ARM.

CUE: HAVE FINISHED RDJIFF

ANS: N/A (information only that the input tape from the attrition model has been processed).

*NOTE 1

CUE: SCHEDULED STOP TIME = 240.1.

*NOTE 1. If a control event had been scheduled at TIME = 240.05 the menu would be printed here enabling the printing of reports at the end of the CI.

ANS: N/A (information only. Processing of CI is complete). Files available at this time are as follows:

Local File Number	Definition
T1	Data base resulting from run just completed
T2	Events that are to be processed at a time subsequent to 240.1 minutes
Output	Audit trail of all events and listing of all reports

An option available at this time is the cataloging of T1 and T2 to form the basis of a checkpoint restart at battle time 240.1 minutes. However, first the output should be batched to the printer to obtain a status of the units and the ammunition resupply network. Before cataloging T1 it is advisable to call HJEDIT in order to update the location of all the units that participated in the last battle. Since the Blue forces are giving ground and moving back, their distance to the ATP and ASP will become shorter. Therefore, it is necessary to change the distance to the ATP and ASP at the end of each CI. This is accomplished by subtracting a standard distance from the values contained in attributes 4 and 5 of each unit. The distance moved by artillery batteries will probably be greater than the distance moved by maneuver battalions, therefore, each unit must be called separately. To accomplish these changes the following procedures should be followed:

CALL, HJEDIT, ID = TRACOMD

CUE: EDIT DATA FILE? (YES/NO)

ANS: N

CUE: UPDATE ARRAYS?

ANS: Y

CUE: VARIABLE NAME (OR END)

ANS: IUNIT

CUE: CHANGE OR REPLACE?

ANS: C

CUE: Enter Word #, Attribute #, New Value (or change)
(0, 0, 0 When Done)

ANS: 1, 4, -4 (subtracts 4 km from ATP distance for maneuver unit 1)

CUE: NEXT-

ANS: 1, 5, -4 (subtracts 4 km from ASP distance for maneuver unit 1)

CUE: NEXT-

. . . .

CUE: NEXT-

ANS: 19, 4,-6 (subtracts 6 km from ATP distance for unit 19, a 155mm battery)

CUE: NEXT-

ANS: 19, 5, -6 (subtracts 6 km from ASP distance for unit 19, a 155mm battery)

CUE: NEXT-

ANS: 30, 5, -5 (subtracts 5 km from ASP distance for unit 30, an 8 in battery)

CUE: NEXT-

ANS: 0, 0, 0 (To get out of routine)

CUE: VARIABLE NAME (OR END)

ANS: END

CUE: EDIT DATA FILE? (YES/NO)

Since the file contains the results of the first CI, it may be desirable to simulate previous stockpiling of artillery ammunition at the first and second supplemental positions. To accomplish this simulation it is necessary to have the output from the CI available to find out which units are short ammunition. If the output is not available, the information can be obtained through the CRT. The method of providing this additional ammunition is to update the current supply of each type of ammunition to 100 percent of the basic load of the surviving weapons. To accomplish this the following procedures should be used, assuming the output is available to look at:

ANS: YES

CUE: VARIABLE NAME =

ANS: IUNIT 19

CUE: . .

ANS: C 10 0 (Zeros out number of weapons short first ammunition type)

CUE: . .

ANS: C 11 0 (Zeros out number of rounds short)

CUE: . .

ANS: C 12 273 (Brings weapons up to 100 percent of basic load of first ammunition type)

CUE: . .

ANS: C 22 0 (Zeros out number of weapons short second ammunition type)

CUE: . .

ANS: C 23 0 (Zeros out number of rounds short)

CUE: . .

ANS: C 24 609 (Brings weapons up to 100 percent of basic load of second ammunition type)

This procedure is continued for all units that are short ammunition. Once completed the word "END" is entered.

CUE: . .

ANS: END

CUE: EDIT DATA FILE? (YES/NO)

ANS: N

CUE: UPDATE ARRAYS

ANS: N

CUE: COMMAND

There is now an updated data file on TAPE 1. Now is the time to catalog TAPE 1 and discard or return T1. It is then necessary to copy TAPE 1 to T1 before processing an additional CI.

Now if the output from the first CI is not available a more time consuming process is followed to bring the units up to 100 percent of their basic load.

CUE: EDIT DATA FILE? (YES/NO)

ANS: Y

CUE: VARIABLE NAME

ANS: IUNIT 20

CUE: . .

ANS: L 10 12 (It is necessary to list the three attributes for each ammunition type for each unit in order to determine if the unit is short that type of ammunition)

CUE: IUNIT 20

ATTRIBUTE 10 = 7

ATTRIBUTE 11 = 140

ATTRIBUTE 12 = 133

ANS: C 10 0

ANS: C 11 0

ANS: C 12 273

CUE: . .

ANS: L 22 24

CUE: IUNIT 20

ATTRIBUTE 22 = 7

ATTRIBUTE 23 = 280

ATTRIBUTE 24 = 329

ANS: 0 22 0

ANS: 0 23 0

ANS: 0 24 609

When all units have been checked, enter the word "END"

CUE: . .

ANS: END

Now by answering N or NO to the next two cues the model will produce the new Tape 1, which should then be cataloged and copied to a new T1. To process an additional or the next CI requires only the following two statements to be keyed in.

```
ATTACH,T3, (next JIFFY produced file)

CALL,HJARMANOTHER,ID=TRACOMD
```

CUE: RETAIN EVENTS CURRENTLY SCHEDULED (YES/NO)

ANS: Y or YES

RES: Events created last CI which are part of local file T2 will be retained.

CUE: Are you creating an answer file (Y or N)

ANS: N or NO

CUE: ENTER TIME TO STOP SIMULATION

ANS: 480.1

RES: A stop simulation event is scheduled for 480.1 minutes.

CUE: INITIALIZE TRUCKS, ETC

ANS: N or NO

RES: NO CHANGE

QUE: (1) - EDIT DATA

(2) - WRITE REPORT

(3) - SCHEDULE CONTROL

(4) - RETURN

(5) - STOP SIMULATION

(6) - EDIT TRUCK QUEUES

(7) - CREATE EVENTS

ANS: 1

CUE: Variable Name =

ANS: LOOK -- must be changed each time the program is started. Only way of obtaining a complete audit trail.

CUE: . .

ANS: C 1 1

ANS: TCIST

CUE: ..

ANS: C 1 240.

RES: Have loaded CI START TIME of 240 minutes

CUE: ..

ANS: INTER

CUE: ..

ANS: C 1 0 -- Zeros out counter for zone 1 trucks

ANS: C 2 0 -- Zeros out counter for zone 2 trucks

ANS: C 3 Number of trucks to be interdicted

ANS: C7 The integer to be used as the cycle number; e.g., 22 means that every 22d truck to move in the simulation will be interdicted.

RES: Have loaded interdiction values for this CI by Initializing zone 1 and zone 2 counters

CUE: ..

ANS: END

RES: Returns to full control menu of alternatives

CUE: (1) - EDIT DATA

(2) - WRITE REPORT

(3) - SCHEDULE CONTROL

(4) - RETURN

(5) - STOP SIMULATION NOW

(6) - EDIT TRUCK QUEUES

(7) - CREATE EVENTS

ANS: 3

CUE: Enter time to stop simulation

ANS: 480.05

RES: At time 480.05 minutes ARM will return to the control routine

RES: Returns to full menu

ANS: 4

RES: Return to process the events for this CI

CUE: HAVE FINISHED RDJIFF

ANS: N/A (information only)

RES: N/A

CUE: Scheduled stop at 480.05. Returns to full menu of alternatives.

ANS: 2

RES: Asks for report type, see Annex II.

RES: After printing last report, program returns to full menu of alternatives.

ANS: 4

CUE: SCHEDULED STOP. TIME = 480.1

ANS: N/A (INFORMATION ONLY)
RES: CI processing has been completed. T1 contains resulting data base and T2 contains unprocessed events, both of which are input for the next CI. Subsequent CIs are run as CI2 but with different times for TCIST.

ANNEX A-I

EDIT DATA FUNCTION

A-I-1 PURPOSE. This appendix describes the EDIT DATA function of the ARM software.

A-I-2 GENERAL.

a. Edit data can be called from any control event when the cue is:

(1) - EDIT DATA

(2) - WRITE REPORT

(3) - SCHEDULE CONTROL

(4) - RETURN

(5) - STOP SIMULATION NOW

(6) - EDIT TRUCK QUEUES

(7) - CREATE EVENTS

ANS: 1

RES: Control is passed to the EDIT subroutine.

b. Edit routine interactions are listed below:

CUE: VARIABLE NAME =

The following commons are then available for access:

(1) IATP

(2) IASP

(3) IUNIT

(4) ITRUCK

(5) ITYPE

(6) IMIX

(7) INTER

(8) IRSTME

(9) IATPSD

(10) IDAY

(11) TIME

(12) ICSA

(13) LPPAR

(14) IASPAM

(15) LUOUT

(16) TCIST

(17) TCILNG

(18) LOOK

c. Definition of each common is at Annex V. Each common has two ordered subscripts, I and J. The I is the ENTITY number; e.g., the unit number, the truck number, or the ATP number. The J is the ATTRIBUTE, which could be the truck type or truck speed, describing the entity.

d. The functions that can be performed by the edit routine are as follows:

(1) List present values.

(2) Change present values for the run only.

e. The next cue is . .

f. The functions to be entered are as follows:

(1) L or LIST enables the listing of present values of an attribute or attributes. For example:

CUE: VARIABLE NAME =

ANS: ITRUCK 17

CUE: . .

ANS: L 2

```
RES:  ITRUCK 17

      ATTRIBUTE 2 = (VALUE)

CUE:  . .

ANS:  IUNIT 22 24

CUE:  . .

ANS:  L 5 6

RES:  IUNIT 22
      ATTRIBUTE 5 = (VALUE)
      ATTRIBUTE 6 = (VALUE)

      IUNIT 23
      ATTRIBUTE 5 = (VALUE)
      ATTRIBUTE 6 = (VALUE)

      IUNIT 24
      ATTRIBUTE 5 = (VALUE)
      ATTRIBUTE 6 = (VALUE)
```

(2) C enables the changing of present values. For example:

```
(a) Cue:  VARIABLE NAME =
     ANS:  ITRUCK 17
     Cue:  . .
     ANS:  C 4 5
     RES:  ITRUCK (17, 4) = 5

(b) Cue:  VARIABLE NAME
     ANS:  ITRUCK
     . .   C 6 0
     RES  (ITRUCK (I, J), I=1, 560, J=6) = 0
```

ANNEX A-II

ARM REPORTS

A-II-1. PURPOSE. This annex describes the reports available from the ARM software.

A-II-2. GENERAL.

a. Reports can be called from any control event when the cue is:

(1) EDIT DATA

(2) WRITE REPORT

(3) SCHEDULE CONTROL

(4) RETURN

(5) STOP SIMULATION NOW

(6) EDIT TRUCK QUEUES

(7) CREATE EVENTS

ANS: 2

CUE: ENTER REPORT TYPE * (1, 2, 3, 4, 5, 6, 7, 8, 9)

ALTERNATIVE 1 Truck Report
*ANS: 1
CUE: 1 - PRINT ALL
2 - SINGLE UNIT
3 - RETURN

SUB-ALTERNATIVE 1
ANS: 1
RES: Produces a report of all unit trucks assigned to units having a non-zero unit type attribute (IUNIT (I,1) a non-zero JIFFY name attribute (IUNIT (I,7)). Sample report is at tab 1 to Annex II.

SUB-ALTERNATIVE 2
ANS: 2
CUE: ENTER JIFFY UNIT ID
ANS: (JIFFY UNIT ID)
RES: Report is printed for one unit only

SUB-ALTERNATIVE 2
ANS: 3
RES: Returns control to control event and menu in A-II-2a above.

Alternative 2 Unit Report
*ANS: 2
CUE: 1 - PRINT ALL
2 - SINGLE UNIT
3 - RETURN

SUB-ALTERNATIVE 1
ANS: 1
RES: Prints all active units and returns to menu in A-II-2a. above.
SUB-ALTERNATIVE 2
ANS: 2
CUE: ENTER JIFFY UNIT ID
ANS: (JIFFY UNIT ID)
RES: Report is printed for one unit only. (Tab 2 to Annex II)
CUE: 1 - PRINT ALL
2 - SINGLE UNIT
3 - RETURN
SUB-ALTERNATIVE 3
ANS: 3
RES: Returns control to control event and menu in A-II-2a. above.

ALTERNATIVE 3 ATP Report
*ANS: 3
CUE: ENTER NUMB OF ACTIVE ATPS (1,2,3, or 4)
ANS: 1
RES: Will print summary data on 1 ATP (tab 3 to Annex II) and returns to the menu in A-II-2a above.

ALTERNATIVE 4
*ANS: 4
CUE: ENTER NUMBER OF ACTIVE ASPS, 1,2,3, or 4
ANS: 1
RES: Prints report for 1 ASP (tab 4 to Annex II) and returns to the menu in A-II-2a above.

ALTERNATIVE 5 CSA Report
*ANS: 5
RES: Prints report (tab 5 to Annex II) on how many rounds by round type have been drawn from the CSA and returns to the menu at A-II-2a.

ALTERNATIVE 6 Multiple ATP Report
*ANS: 6
RES: Produces a summary report (tab 6 to Annex II) containing information summarized across all ATPs and returns to the menu at A-II-2a.

ALTERNATIVE 7   Ammo Removed from ASP Report
*ANS: 7
RES: Prints a report (tab 7 to Annex II) containing summary information across all ASPs and returns to the menu at A-II-2a.

ALTERNATIVE 8   Trucks Killed and/or Broken Down
*ANS: 8
RES: Prints a report (tab 8 to Annex II) containing a list of all trucks whose status equals seven (killed by interdiction) or six (down for remedial maintenance) and returns to the menu in A-II-2a.

ALTERNATIVE 9   All Reports
*ANS: 9
CUE: 1 - PRINT ALL
     2 - SINGLE UNIT
     3 - RETURN
ANS: (See ALTERNATIVE 2 above.)
CUE: ENTER NUMB OF ACTIVE ATPS (1,2,3, or 4)
ANS: (See Alternative 3 above.)
CUE: ENTER NAME OF ACTIVE ASPS (1, 2, 3, or 4)
ANS: (See Alternative 4 above.)
CUE: ENTER NUMB OF ACTIVE ASPS (1, 2, 3, or 4)
ANS: (See Alternative 7 above.)
RES: Prints a copy of all reports alternatives 1 through alternative 8 and returns to the menu at A-II-2a.

b. If the value of LUOUT is set equal to 2 in EDIT the reports will be printed at the CRT. If LUOUT is set to 6 the reports will be written to the output file to be printed at the DPFO if desired.

ANNEX A-III

SUBROUTINE TRKPUT

A-III-1. PURPOSE. This annex describes the subroutine TRKPUT and its functic

A-III-2. GENERAL. TRKPUT enables interactive assignment, unassignment, or reassignment of trucks to truck queues.

A-III-3. Process Menu as appears from the control event follows:

a. CUE: TIME = XXX.XX
(1) - EDIT DATA
(2) - WRITE REPORT
(3) - SCHEDULE CONTROL
(4) - RETURN
(5) - STOP SIMULTION NOW
(6) - EDIT TRUCK QUEUES
(7) - CREATE EVENTS
ANS: 6

b. CUE: COMMAND EXAMPLES
GET 3 FROM 35
PUT 3, 10 IN 105
LIST 105
TAKE ALL OUT
END

ALTERNATIVE 1 - GET
Purpose: GET removes one truck from a queue and returns to menu A-III-3b.
Examples: GET 3 FROM 35 removes truck number 3 from queue number 35.
G 3 FROM 35 (same as above).
G 3 35 (same as above).
G, 3, 35 (same as above).

ALTERNATIVE 2 - PUT
Purpose: PUT places a single truck or a continuously numbered set of trucks a queue and returns to menu A-III-3b.
Examples: PUT 3, 10 IN 105 (places trucks 3 through 10 into queue 105)
P 3, 10 IN 105 (same as above)
P 3 10 105 (same as above)
P 3, 10, 105 (same as above)

ALTERNATIVE 3 - LIST
Purpose: LIST displays the numbers of the trucks in the queue and returns to menu A-III-3b.
Examples: LIST 105 (Lists the trucks in Queue 105)
L 105 (same as above)
L, 105 (same as above)

ALTERNATIVE 4 - TAKE ALL OUT
Purpose: TAKE ALL OUT initializes all the truck queues and returns to menu A-III-36.

Examples: TAKE ALL OUT (Initializes all truck Queues)
T (same as above)

ALTERNATIVE 5 - END
Purpose: Returns to control event and the first menu in A-III above.
Example: END (returns to the first menu in A-III-3 above).

ANNEX A-IV

SUBROUTINE CREEVT

A-IV-1. PURPOSE. This annex describes the subroutine create event (CREEVT) and its function.

A-IV-2. GENERAL. CREEVT enables the building of events to occur later within the simulation. The 17 event types, their parameters, and a brief description of each are listed below:

Event Type	Parameters (Up to 5 possible)	Description
1 - Demand	Unit Number	Checks the ammo demand of units.
2 - Reload	Unit Number	Replaces ammo rounds at unit weapons
3 - UNTDEP	Unit Number, Truck Number	Departure of empty ammo truck from unit.
4 - ATPARV	Unit Number, Truck Number, ATP Number	Arrival of unit truck at ATP
5 - ASPARV	Unit Number, Truck Number, ASP Number	Arrival of unit truck at ASP
6 - ATP	Type Queue, ATP Number	Process unit truck at ATP
7 - ASP	Type Queue, ASP Number	Process unit truck at ASP
8 - UNTARV	Unit Number, Truck Number	Arrival of unit truck at UNIT
9 - CSAARV	ATP or ASP Number, Truck Number, ATP-ASP Flag	Arrival of Truck at CSA
10 - ATPAR1	ATP Number, Truck Number	Arrival of CSA Truck at ATP
11 - ATPAR2	ATP Number, Truck Number	Arrival of ASP Truck at ATP
12 - ASPAR1	ATP Number, Truck Number, ASP Number	Arrival of ASP Truck at ASP
13 - HELARV	Unit Number, Truck Number	Arrival of Helicopter at Unit
14 - HASPAR	Empty, Truck Number	Arrival of Helicopter at ASP
15 - REPORT	Report Number	Prints Report
16 - CONTRL	TIME	Prints Menu in A-IV-3
17 - ENDSIM	NONE	Writes events to Mass Storage

A-IV-3 PROCESS. the menu as it appears in the control event follows:

```
CUE:  TIME = XXX.XX
      (1)  - EDIT DATA
      (2)  - WRITE REPORT
      (3)  - SCHEDULE CONTROL
      (4)  - RETURN
      (5)  - STOP SIMULATION NOW
      (6)  - EDIT TRUCK QUEUES
      (7)  - CREATE EVENTS
       ?
ANS:  7
CUE:   TO CREATE AN EVENT, INPUT AS A GROUP SEPARATED BY COMMAS OR SPACES
       THE FOLLOWING 7 VALUES: EVENT TYPE (INTEGER VALUES BETWEEN 1 AND
       17) 5 PARAMETERS FOR EACH EVENT (INTEGER, DEPEND ON EVENT TYPE) AND
       TIME (DECIMAL MINUTES, REAL)  EXAMPLE:  10, 1, 512, 0, 90, 305.
       CSA-TO-ATP TRUCK 512 WILL ARRIVE AT ATP AT TIME = 305 MIN.
```

Sub-alternative 1
ANS: END
RES: Returns to the control menu above.
Sub-alternative 2
ANS: HELP
RES: Returns to line 5 of CUE in A-IV-3 above.
Sub-alternative 3
Enter the events to be processed.

ANNEX A-V

DATA COMMONS

A-V-1. PURPOSE. This annex describes the contents of the data commons of the ARM software.

A-V-2. GENERAL. The software requires space to store descriptive data to simulate the ammunition resupply. Four labeled common areas are used: LOG, EVENTS, QUENUM, and QUEPNT. Within the labeled common areas are a number of groups of specifically related data. The following is a field by field definition of the grouped data sets:

A-V-3. IATP (4,30)
4 ATP's
30 words each as follows:

1. Distance to CSA
2. Distance to ASP
3. UTM Coordinate
4. Number of helicopter missions scheduled in total simulation
5. Empty
6. Associated ASP number
7. Empty
8. A flag that = 0 if arty queue has not served a truck this war, 1 otherwise
9. Number artillery servers active
10. Number maneuver unit servers active
11. Artillery queue number
12. Maneuver unit queue number
13. A flag that = 0 if mnvr queue has not served a truck this war, 1 otherwise
14. Number trucks in artillery queue
15. Number trucks in maneuver unit queue
16. Current ammo supply, ammo 1
17. Queue ammo demand, ammo 1
18. Basic ammo level, ammo 1

19-21 , ammo 2
22-24 , ammo 3
25-27 , ammo 4
28-30 , ammo 5

A-V-4. IATPSD (5) -- ATP Service Data

1. Max servers
2. Threshold 1 for queue 1
3. Threshold 2 for queue 1
4. Threshold 1 for queue 2
5. Threshold 2 for queue 2

A-V-5. IASP (4,41)

4 ASP

41 words each as follows:

1. Distance to CSA
2. Distance to ATP
3. UTM coordinate
4. Empty
5. Number trucks to CSA
6. A flag that = 0 if a routine queue has not served a truck this war, 1 otherwise.
7. Number routine servers active
8. Number GSRS servers active
9. Routine queue number
10. GSRS queue number
11. A flag that = 0 if a GSRS queue has not served a truck this war, 1 otherwise
12. Number trucks in routine queue
13. Number trucks in GSRS queue
14. Current ammo supply, ammo 1

15-33. Ammo 2-20

34-41. Empty

A-V-6. IASPAM (4,20) AMMO REMOVED FROM ASP

4 ASP, 20 AMMO TYPES

A-V-7. IUNIT (75,69)

75 Units

69 Words each as follows:

1. Type
2. ATP number
3. ASP number
4. Distance to ATP
5. Distance to ASP
6. UTM coordinate
7. Unit name
8. First ammo type
9. Number weapons alive, First ammo type
10. Number weapons short ammo, First ammo type
11. Number rounds short, (wpns) First ammo type
12. Current ammo supply, (wpns) First ammo type
13. Routine resupply level, (per wpn) First ammo type
14. Critical resupply level, (per wpn) First ammo type
15. Basic Ammo level, (per wpn) First ammo type
16. Ammo on trucks First ammo type
17. Number of weapons killed during CI, 1st ammo type
18. Number of weapons short ammo First ammo type
19. Total rounds short through whole CI First ammo type

20-31 , Second ammo type

32-43 , Third ammo type
44-55 , Fourth ammo type
56-67 , Fifth ammo type
68. Number of helicopters assigned
69. = 0 if single pulse demand per CI
= 1 if multiple pulses per CI

IRSTME (20,3) -- resupply time data
20 types of ammo
3 words each as follows
1. Weapon set-up time
2. Load time per round
3. Travel time to weapon

ITRUCK (560,7)
560 trucks
7 words each as follows:
1. Truck type
2. Mission type
3. Status type
4. Owner number
5. Ammo mix number
6. Percent loaded
7. Time since last failure

A-V-8. ITYPE (6,6)
6 types of trucks
6 words for each type truck as follows:
1. Secondary road night speed (unit to ASP, ATP)
2. Secondary road day speed (unit to ASP, ATP)
3. Highway night speed
4. Highway day speed
5. MTBF
6. MTTR

A-V-9. Truck Queues: QUEUE TYPE:

1 - 75	1	At each unit
101 - 104	2	At ATPS for CSA-ATP trucks
105 - 108	3	At ATPS for ASP-ATP trucks
109 - 112	4	At ATPS for unit artillery server
113 - 116	5	At ATPS for unit maneuver server
117 - 120	6	EMPTY
121 - 124	7	At ASPS for CSA-ASP trucks
125 - 128	8	At ASPS for routine server
129 - 132	9	At ASPS for GSRS server
133 - 136	10	EMPTY

A-V-10. IMIX (40,23)
40 ammo mixes
23 words for each mix as follows:

1-20. Number rounds of each ammo type
21. Load time at CSA
22. Load time at ATP
23. Load time at ASP

A-V-11. Event scheduling
Common/Events/JSTAT(6), JEVDS (1024,4), IEVS (5,1024)

A-V-12. QUEUE DATA
COMMON/QUENUM/IHEAD (136)
COMMON/QUEPNT/ITEMS (560)

A-V-13. Interdiction Data - Common Inter (9)
1. Counter for zone 1 trucks killed in intrdk
2. Counter for zone 2 trucks killed in intrdk
3. Number of trucks to be killed in zone 1
4. Number of trucks to be killed in zone 2
5. Time to replace truck in zone 1
6. Time to replace truck in zone 2
7. Modulo of trucks to be killed in zone 1 and zone 2
8. Number of zone one trucks entering intrdk
9. Number of zone two trucks entering intrdk

73/73 OPT=1 FTN 4.6+460

A-V-14. IDAY: 1 = Day , 0 = Night

A-V-15. ICSA (20) -- Number of rounds by ammo type from CSA

A-V-16. LPPAR(5)
LPPAR(1) -- Total number of ammo codes (20)
LPPAR(2) -- Number of ammo codes at ATP (5)
LPPAR(3) -- Number of maneuver unit ammo codes at ATP (2)
LPPAR(4) -- Number of transports (trucks)(Lt 560)
LPPAR(5) -- Number of helicopters available

A-V-17. TCIST -- Time of start of CI in decimal minutes

A-V-18. TCILNG -- Time of length of CI in decimal minutes

A-V-19. Time -- Simulation time in minutes (decimal)

A-V-20. Unit Type Codes:

1	Tank Task Force
2	Mech Task Force
3	Armrd Cav Sqdn
4	155 Arty Btry
5	8 Inch Arty Btry
6	GSRS Btry
7	Divad Gun Plt
8	Cbt Avn Plt

A-V-21. Truck type codes:

1	10 Ton
2	5 Ton
3	5 Ton with 1 1/2 ton trailer
4	10 ton with 15 ton trailer
5	22 1/2 ton stake & platform
6	helicopter

A-V-22. Ammo type codes:

1	105mm (M60-A3/XM1)
2	TOW
3	Powder Canisters
4	155 HE
5	155 ICMDP
6	155 Smoke
7	155 CLGP
8	8 Inch HE
9	8 Inch ICMDP
10	GSRS
11	Mortar
12	DIVAD
13	HELLFIRE
14	XR-TOW
15	STINGER
16	DRAGON
17	BUSHMASTER
18	EMPTY
19	EMPTY
20	EMPTY

A-V-23. Truck Mission Type Codes:

1	Unit truck
2	CSA - ATP Link
3	CSA - ASP Link
4	ASP - ATP Link
5	ASP - Unit (Helicopter)

A-V-24. Truck Status Type Codes:

1 IN Unit Queue
2 IN ATP Queue
3 IN ASP Queue
4 IN TRANSIT
5 Unit truck going from ATP to ASP
6 Truck awaiting repair
7 Truck dead (interdicted)

DISTRIBUTION LIST

ORGANIZATION	NO. OF COPIES
Commander US Army Training and Doctrine Command Fort Monroe, VA 23651	
ATCD-SI (Mr. Christman)	1
ATCD-C	1
Commander Defense Documentation Center Cameron Station Alexandria, VA 22314	10
Director USATRASANA ATTN: ATAA-PFB White Sands Missile Range, NM 88002	2
Commander USA Logistics Center Ft Lee, VA 23801	
ATCL-C	1
ATCL-CF	1
ATCL-LE	1
ATCL-OS	1
Commander US Army Air Defense Center & Fort Bliss ATTN: -CD Fort Bliss, TX 79916	1
Commander US Army Aviation Center & Fort Rucker ATTN: ATZQ -CD Fort Rucker, AL 36362	1
Commander US Army Armor Center & Fort Knox ATTN: ATSB-CD-S Fort Knox, KY 40121	2
Commander US Army Engineer Center & Fort Belvoir ATTN: ATSEN-CTDk-CS Fort Belvoir, VA 22060	1

ORGANIZATION	NO. OF COPIES
Commander US Army Field Artillery School ATTN: ATSF-CTD-S Fort Sill, OK 73503	2
Commander US Army Infantry School ATTN: ATSH-CD-CS Fort Benning, GA 31905	2
Commander US Army Intelligence Center and School ATTN: ATSI-CTD-CS Fort Huachuca, AZ 86611	1
Commander US Army Missile and Munitions Center and School ATTN: ATSK-CD-CS Redstone Arsenal, AL 35809	2
Commander USA Ordnance Center and School ATTN: ATSL-CTD-CS Aberdeen Proving Grounds, MD 21005	1
Commander USA Institute of Military Assistance DCOMDT Cmt Tng Div Fort Bragg, NC 28307	1
Commander USA Transportation School ATTN: ATSP-CTD-CS Fort eustis, VA 23604	2
Commander USA Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, MD 20014	1
Commander USA Combined Arms Combat Developments Activity Fort Leavenworth, KS 66027	
ATZLCA-CA	5
ATZLCA-SW	2

ORGANIZATION	NO. OF COPIES
Commander Command and General Staff College ATTN: ATZLSW-TA Fort Leavenworth, KS 66027	
Deputy Commander USAMSAA ATTN: AMXSY-T Aberdeen Proving Ground, MD 21005	1
US Air Force Tactical Fighter Weapons Center/SATC ATTN: TFWC-SA Nellis AFB, NV 89191	2
Professor S. H. Parry, Code 55Py Department of Operations Research Naval Postgraduate School Monterey, CA 73940	1

M008200