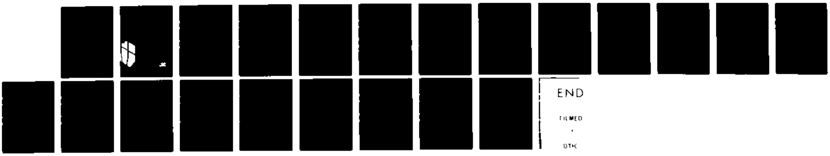
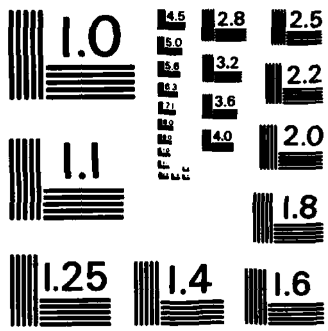
AD-A121 168 ENGINEERING MODELING STUDY VOLUME III CORDIVEM/ENGINEER MODULE INTERFACE MANUAL(U) CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL C MILLS SEP 82 1/1

UNCLASSIFIED CERL-TR-P-131-VOL-3 F/G 15/7 NL



END
FILMED
DTIC



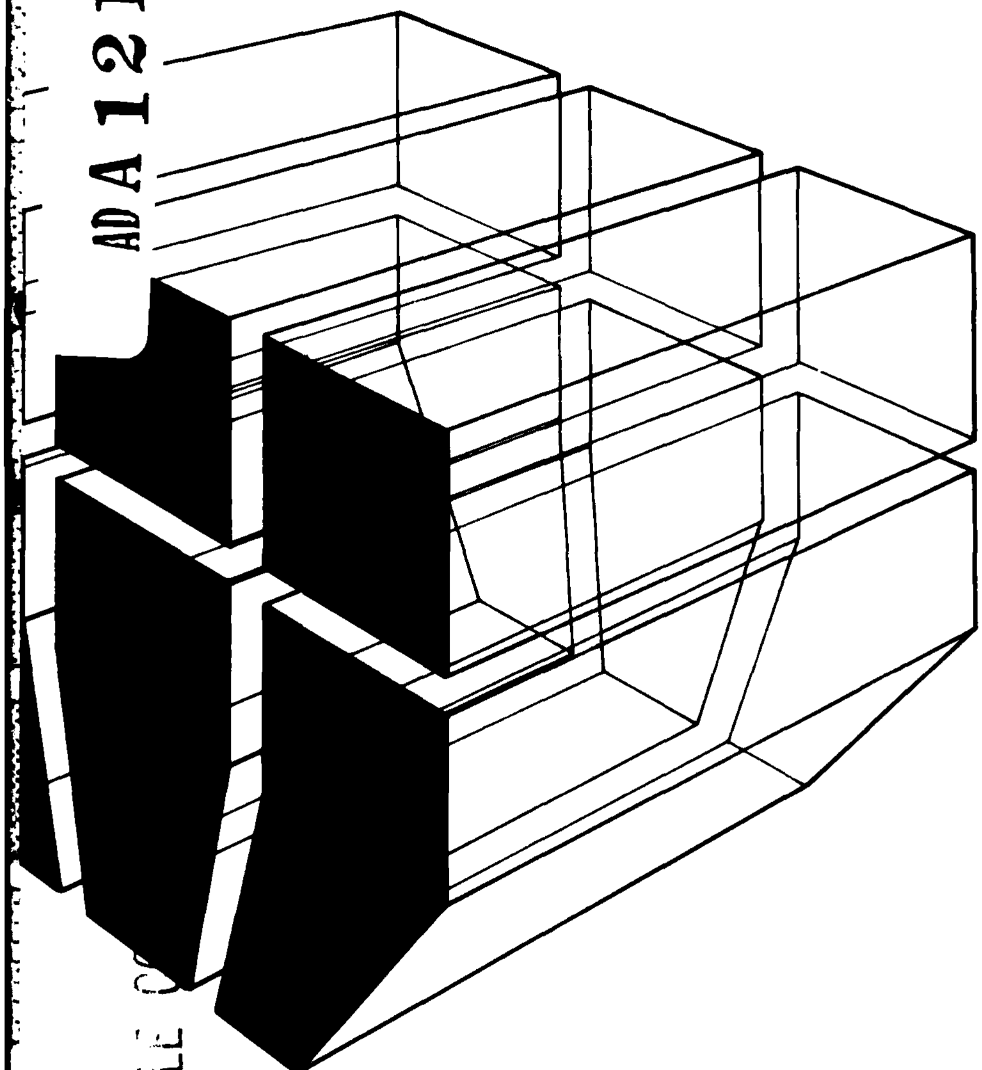
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

construction
engineering
research
laboratory

United States Army
Corps of Engineers
...Serving the Army
...Serving the Nation

TECHNICAL REPORT P-131
September 1982
Engineer Modeling Study

ENGINEER MODELING STUDY VOLUME III: CORDIVEM/ENGINEER MODULE INTERFACE MANUAL



by
Carlton Mills

CERL

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CERL-TR-P-131	2. GOVT ACCESSION NO. AD-A121168	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) ENGINEERING MODELING STUDY VOLUME III: CORDIVEM/ENGINEER MODULE INTERFACE MANUAL		5. TYPE OF REPORT & PERIOD COVERED FINAL
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Carlton Mills		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. ARMY CONSTRUCTION ENGINEERING RESEARCH LABORATORY P.O. BOX 4005, CHAMPAIGN, IL 61820		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 4A762731AT41-D-048
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE September 1982
		13. NUMBER OF PAGES 16
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) Unclassified
		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

Copies are obtainable from the National Technical Information Service
Springfield, VA 22151

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

Models
War games
Military engineering
Combat support
Computerized simulation

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This is the third volume of a three-volume report that documents the capabilities and use of the Engineer Module developed by the U.S. Army Construction Engineering Research Laboratory (CERL). This module is designed to operate as part of CORDIVEM, the corps- and division-level component of the Army Model Improvement Program (AMIP). The Engineer Module represents the contribution of the combat engineer to the combined arms team during computer-simulated battles by altering terrain characteristics within the simulation's data base.

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

BLOCK 20. CONTINUED.

This volume describes the interface of the Engineer Module with CORDIVEM, including automatic generation of engineer orders, modification of terrain features due to engineer activities, and an interactive interface for an engineer gamer.

FOREWORD

This study was performed for the Directorate for Combat Developments (DCD), U.S. Army Engineer School (USAES), under Project 4A762731AT41, "Military Facilities Engineering Technology"; Task D, "Combat Engineering Strategy"; Work Unit 048, "Engineer Modeling Study." The USAES/DCD Technical Monitor was CPT Clifford Clausen.

The study was conducted by the Facility Systems (FS) Division of the U.S. Army Construction Engineering Research Laboratory (CERL). CERL personnel directly involved in the study were Dr. Gordon Bagby, Mr. Gerald Brown, Mr. Carlton Mills, and Mr. Theodore Tourlentes. Appreciation also is expressed to Professor John Boyd of Knox College, Galesburg, IL. Mr. Edward Lotz is Chief of CERL-FS.

COL Louis J. Circeo is Commander and Director of CERL, and Dr. L. R. Shaffer is Technical Director.

CONTENTS

TABLES

ENGINEER MODELING STUDY VOLUME III: CORDIVEM/ENGINEER MODULE INTERFACE MANUAL

1 INTRODUCTION

Background

In 1980, the Army initiated the Army Model Improvement Program (AMIP). One of the goals of AMIP is to ensure that the next generation of Army combat models is "systemic." That is, one model will simulate several days or even weeks of combat with no manual intervention. As a consequence of this requirement, the U.S. Army Construction Engineering Research Laboratory's (CERL's) Engineer Modeling Study (EMS) group was asked to find ways to represent engineers in one of the Army's combat simulation models. When CERL's EMS group began its study, existing Army combat models either ignored engineers completely or confined their representation to obstacle effects (e.g., minefields, tank ditches) whose sizes and location were input manually. Because manual input was tedious, often the only engineer contribution played was an initial barrier plan. In one model, the engineer representation was so slow that it was usually turned off. Another model only estimated (by drawing a random number) the engineers' arrival time to clear a minefield and then decided whether to wait or bull-through. Thus, CERL's EMS group directed its efforts at designing an Engineer Module which could automatically generate engineer orders without time-consuming input requirements. This module was designed for use as part of the Corps/Division Evaluation Model (CORDIVEM) component of AMIP.

Objective

The overall objectives of this study were:

1. To model the contribution of friendly and enemy engineers to the effectiveness of the combined arms team.

2. To represent, with accuracy and consistency, the effectiveness of the combat engineer effort throughout the AMIP model hierarchy.

3. To develop a system that the U.S. Army Engineer School (USAES) and other agencies can use to (a) review modeling data so new equipment or doctrine can be easily incorporated, and (b) support the evaluation of hypothetical equipment or doctrine.

The objective of this volume is to describe how the Engineer Module is implemented as part of the CORDIVEM component of AMIP.

Approach

In 1980, the Combined Arms Study Activity (CASAA) decided to base its implementation of CORDIVEM on the Integrated Corps Model (ICOR) developed by the BDM Corporation. CERL was asked to integrate its SIMSCRIPT Engineer Module with the ICOR FORTRAN program. The CORDIVEM task force at Fort Leavenworth, KS, implemented the CORDIVEM Game Executive Module (GEM) in SIMSCRIPT. The GEM calls ICOR using the SIMSCRIPT-to-FORTRAN call facility.* CERL's Engineer Module could then be compiled as part of the GEM, thereby eliminating FORTRAN/SIMSCRIPT compatibility problems.

Implementation efforts initially concentrated on automatically generating engineer orders for tactical units in the defense posture. ICOR had no method for the gamers to enter a defensive concept of operations. This deficiency presented the opportunity to implement a facility which would allow the gamers merely to specify a series of defensive positions and have the program automatically generate engineer orders to support the unit. The default mode of operation then would include minimal engineer support, even if no engineer gamer existed. The default also would select defensive positions when they were not explicitly provided, thus complying with the "systemic" requirements of AMIP.

Final implementation efforts were directed toward developing an interactive interface, generating mobility orders, and smoothly integrating a resource allocation module. Implementation requirements were:

1. Automatic ("systemic") generation of engineer orders.

2. Interactive entry of engineer orders and supervision of the Engineer Module.

3. Representation of engineer effects.

4. Representation of engineer chain of command.

5. Integration of the engineer resource allocation module.

*It is not possible for FORTRAN routines to call SIMSCRIPT routines.

Organization of Report

Chapter 2 describes, in military terms, what CERL's Engineer Module does. The description follows the structure of engineer doctrine and summarizes what the implementation purports to have done. Chapters 3 through 12 describe what the software does and are intended for use by CORDIVEM maintenance personnel.

2 SUMMARY OF FEATURES

Defensive Path

Every tactical unit in contact with a hostile force has a defensive path. The path is a preplanned route to an alternate fighting position. The decision to move to the alternate position due to threat pressure is made by CORDIVEM—the defensive path implementation does not affect the decision logic. The defensive path anticipates the move, issues countermobility engineer orders along the path, and issues survivability orders to prepare the new position. The series of positions can be entered by the gamer (man-in-the-loop [MITL]) or generated automatically by the software. Since the path is preplanned, lead time is provided to perform engineer work before the supported unit moves.

If a unit has no preplanned defensive path, the following steps are used to create one:

1. Select a hex 12 to 20 km to the unit's rear.

2. Examine that hex and the six neighboring hexes for defensibility (see below).

3. Select the hex with the highest defensibility.

4. Plan a path from the unit's current hex to the selected hex.

5. Issue countermobility orders for targets of opportunity along the path.

The algorithm for scoring defensibility:

1. Scores the target hex for cover using defensive terrain weights (rivers and roads do not degrade cover values).

2. Scores the two adjacent hexes facing the threat for cover using offensive terrain weights (rivers and roads decrease cover value for attacking units).

3. Calculates a composite score for the two adjacent hexes using speed as a weighting factor (a hex with an autobahn is more likely to be used for an attack).

4. Computes the defensibility score by dividing the Step 1 value by the Step 3 value.

The hexes with the highest defensibility scores will be those hexes where the terrain gives the defender an advantage relative to the attacker in an adjacent hex. The chosen defensive position might be a wooded area next to an open area, or behind a river, which forces the threat to attack across a river. Hexes with no inherent advantage will not be selected (e.g., adjacent heavily forested areas or adjacent open areas).

The gamer may override the default defensive path mechanism at any time by explicitly entering a unit's defensive path. For instance, if a critical bridge is to be held, the gamer can specify fighting positions for two or more units in front of the bridge. Threat pressure which causes units to move then would concentrate friendly protection for the bridge. The default would select positions across the river and issue demolition orders to destroy the bridge. Threat pressure would cause the defending units to move across the river; the engineers then would destroy the bridge. Advancing threat units then would confront a river barrier covered by fire. Default action will occur for all units, Blue and Red, unless explicitly overriden or disabled by the gamers.

Offensive Path

The CORDIVEM Engineer Module includes a facility to support gamer planning during the simulation. This facility, called the offensive path, permits the gamer to ask, in effect, "I have a unit at point A and would like to send it to point B. What is the best possible path and what engineer effort is required to create this ideal route?" Often, the best path is the fastest. However, the calculation can include an evaluation of cover. By making the cover weight high (relative to speed), the path will tend to avoid open areas and request bridges in areas with good cover.

Mobility Operations

The module will automatically generate engineer orders to remove any obstacles encountered by tactical units. The module will not alter movement rates and the movement path for the encountering unit. However, after the obstacle is cleared, any following units will move faster and possibly on a different route. For minefields, the CORDIVEM obstacle routines calculate attrition due to mines, modify movement rates, and make the decision whether to bull-through, by-pass, or wait for support.

The MITL has several ways to issue mobility orders, including designating an area where all obstacles shall be cleared, indicating sites for bridging, and ordering the clearing of specific obstacles.

For the first version of CORDIVEM (CORDIVEM MOD 1), no attempt is made to automatically generate major river crossing orders. There is no simple method of deducing where and when to commit these scarce resources. A major river bridging order will be available for use by the scenario developers (and online gamers) as they implement their concept of operation. Later versions of CORDIVEM will use more sophisticated decision logic, which could include major river crossing operations.

Countermobility Operations

For those units with defensive operation orders, a systemic facility exists to automatically generate countermobility and survivability orders. Whenever a defensive tactical unit comes into threat contact, the program generates a defensive path, issues orders to prepare fighting positions at the end point, and issues countermobility operations along the path. If the scenario staff has entered a defensive concept of operations, then the generated path will observe the scenario constraints.

The type of countermobility orders issued depends on the kinds of terrain traversed by the defensive path. The algorithm selects locations where roads cross areas that have low cross-country mobility (e.g., an autobahn on a forested mountain), and issues a crater order.

Survivability

The only automatically generated survivability orders are the prepared fighting positions on the defensive path. When the defensive path is tied to a scenario's concept of defensive operation, the gamer may request "strong points" instead. Of course, a survivability order may be entered at any time by the MITL.

Summary

This concludes the general information section of this report. The remaining chapters provide reference material for the CORDIVEM/Engineer Module integration staff. Table 1 summarizes the systematic (automatic) and MITL features of the CORDIVEM implementation.

Table 1
Summary of Systemic and MITL Actions

	Countermobility
SYSTEMIC:	Generation of defensive path
	Prepared position at movement objective
	Road craters and/or linear obstacles along path
MITL:	Entry of defensive concept of operation with default defensive path actions, explicit strong point, or explicit barrier/fighting position-fre area
	Explicit designation of obstacle trace
	Explicit designation of obstacle-free areas
	Survivability
SYSTEMIC:	Prepared positions on defensive path
MITL:	Strong points on defensive path
	Explicit strong points
	Mobility
SYSTEMIC:	Automatic clearing of obstacles encountered by tactical units
MITL:	Explicit river crossing representation
	Explicit designation of battlefield area where obstacles shall be cleared
	Offensive path mechanism to create engineer work packages

3 PROGRAM STRUCTURE

Implementation can be viewed as several ongoing processes: planning, target execution, and game support.

Planning

The planning process monitors the status and objectives of supported maneuver units in order to anticipate combat engineer requirements. This planning horizon provides the lead time between issuing an order (e.g., crater a road) and execution (e.g., actually cratering the road) needed to simulate the movement of engineer assets to the work site and the performance of the task. Given the ability to explicitly model engineer assets, it then becomes possible to change the engineer force structure and determine the battlefield consequences.

Implementation required changes to subroutine CONSDR.

Target Execution

The target execution process modifies the terrain, given completion information received from the resource allocation module (e.g., 300 m of tank ditch completed). However, some targets, like blowing a bridge, cannot be executed immediately; the module must wait until the supported unit has crossed the bridge and then change the terrain data to show an unbridged river to a pursuing threat force.

Implementation required changes to subroutines MOVSCHD and MOVRCT.

Game Support

The game support process includes such things as explicit entry by the gamers of engineering orders, graphic display of obstacles, force structure representation, etc.

Implementation required changes to the Player Control Station (PCS) program.

4 OFFENSIVE PATH IMPLEMENTATION

The offensive path algorithm consists of the following steps:

1. *Given* a start hex and a target hex, *build* a *static table* of all hexes (nodes) within a five-hex (17.5-km) wide swath between the two and a *link table* connecting these nodes.

2. Calculate the cost of traversing each link.

3. Build a node cost table where each node value is the sum of the link values for the least costly path to the target node.

4. Calculate the cost of traversing each link (assuming engineer activity).

5. Build another node cost table where the node values reflect the engineer activities.

6. Using the table constructed in Step 5, find the least costly path and engineer orders required to create this path.

7. Report the following items:

a. The cost for moving to the target using unimproved terrain from Step 3.

b. The cost of using engineer-enhanced terrain from Step 5.

c. The engineer work required (from Step 6) on the optimal path.

The report is presented to the gamer, who can interactively accept and reject each of the possible engineer orders. Then, the gamer will release the entire work package to the module. The gamer can optionally issue orders for the tactical unit to follow the path.

The initial implementation will calculate cost in terms of time; however, the method is completely general. Other functions could be used to provide the link cost values of Step 2 and Step 5. (For instance, one could include a cover factor and calculate a reasonably fast path that minimizes the use of open terrain.)

Data Structures

The data structures are given in Table 2.

Building the Static Tables

The path table is built by alternately moving a hex pointer (initially set to the start hex) and another pointer (initially set at the end hex) towards each other. The pointers are incremented one hex at a time. After each increment a five-hex (17.5-km) wide swath is added to the table. When the pointers meet at the halfway point, the process terminates with the table built.

Table 2
Data Structures

Node Table: scores static information about node

(0, ⟨ node # ⟩)	Number of links for this hex
(1, ⟨ node # ⟩)	Link data
.	= 0 no link
.	⟩ 0 link number (this node is source)
(6, ⟨ node # ⟩)	⟨ 0 ABS (link number) (this node is sink)
(7, ⟨ node # ⟩)	Number of incoming links
(8, ⟨ node # ⟩)	Hex number
(9, ⟨ node # ⟩)	Pointer to hex data structure

Link Table: static

(1, ⟨ link # ⟩)	Source node
(2, ⟨ link # ⟩)	Sink node
(3, ⟨ link # ⟩)	Direction

Note: Node 1 is always the source node
Node 2 is always the destination node

COST ORIG	(⟨ node # ⟩)	Cost without engineers
COST ENGR	(⟨ node # ⟩)	Cost with engineers
CLINK ORIG	(⟨ link # ⟩)	Link cost without engineers
CLINK ENGR	(⟨ link # ⟩)	Link cost with engineers
LINK ENGR ORD	(⟨ link # ⟩)	Engineer order required to change terrain
OFPATH	(*)	List of hex numbers along path
NUMNODES		
NUMLINKS		

Because of the vagaries of hex arithmetic, there is no guarantee that the points will meet in the same hex. Therefore, the implementation will terminate whenever the distance between the two points is less than two hexes or when no new hexes have been added to the table.

The links between nodes (hex locations) are established as hexes are added to the table. Whenever one of the six possible neighbors of a new hex is found in the node table, a link is added to the link table.

Finding the Best Path

The best path can be found easily if every node had a least-cost value for the best path from it to the destination node. The best path can be enumerated by beginning with the start node and choosing the link to the lowest cost neighbor node and repeatedly choosing the least-cost neighbor until the target node is reached. If there are multiple least-cost paths, only one will be chosen.

Building the Least-Cost Table

The problem is now how to find the least-cost value for every node. If a value for a source node is known, then each link can be taken from that node, and the link cost can be added to the source cost, which calculates a cost for the sink. If this cost is lower than the current cost, then store the new lower cost as the sink node value. When this algorithm can be performed on every node without calculating new node values, the least-cost node table has been built (assuming that the nodes are initialized with very large values).

Clearly, though, just repeating the calculation for every link is time-consuming. A much faster method uses a list of pending nodes which have new values, but whose values have not been propagated.

The algorithm that builds the node cost table removes a node from the pending list, calculates new values for those nodes connecting with it, and when a new lower node cost value is found, adds that node to the pending list. Eventually, the pending list will be emptied and the process terminated. The process starts by putting the offensive path target node into the pending list and setting the cost of the target node to zero.

5 RIVER CROSSING REPRESENTATION

The CORDIVEM terrain attributes have been extended to support representation of engineer-built bridges and rafting operations. The bridge sites may be selected by the offensive path algorithm.

New Value Range of Hexside Attributes

Road Values

0 = no road
1 = secondary/tertiary road
2 = primary road
3 = autobahn
5 = secondary road (point blockage)
6 = primary road (point blockage)
7 = autobahn (point blockage)

The above applies where point blockage is a blown bridge when river is present, or road crater/abati in the absence of rivers.

River Values

0 = no river
1 = river width under 10 m
2 = river width 10 to 120 m
3 = river width greater than 120 m
4 = river with rafting site in place
(The crossing rate will depend on the river width and number of rafts available.)
5 = Level 1 river with AVLB bridge in place
6 = Level 2 river with float bridge in place

Discussion

The following examples show how these attributes are used:

River	*Road*	
2	3	100-m river with autobahn bridge
2	7	Autobahn leading up to 100-m river (no bridge)
6	3	Autobahn with float bridge crossing 100-m river
4	3	Autobahn leading up to raft site
1	0	10-m river (no bridge, no road)
5	0	10-m river with AVLB (no road)

6 ENGINEER DATA STRUCTURES

A typical engineering order, like prepare fighting positions, will involve many individual items of work, e.g., defilades, tank ditches, abatis, and minefields. The number and mix of items varies with the terrain, while the order of completion depends on resource constraints. All items may not be finished by the time the supported unit arrives, but the partially completed order should affect the battlefield. Some method to translate notifications of individual item completion into an integrated partial completion value for making terrain modifications had to be invented.

The method chosen requires that each item in a particular engineer order be given a point value. The nominal total point score for all items is 1000; the current sum of points for all completed items divided by 1000 yields a percentage complete value, which is used to make terrain modifications. The resource allocation model signals item completion by invoking a subroutine with the item's point score as a parameter. The score is added to the total score for the order. If the resulting terrain modifications do not have to be deferred (cratering a road, for instance, has to occur after the supported unit uses the road), the terrain modification routines are invoked immediately.

To execute deferred orders, the module calls a target execution routine whenever a supported unit initiates or completes a hex-to-hex move. This routine searches the supported unit's list of outstanding orders and invokes the terrain modification routines if the unit has moved behind the deferred target.

7 MODEL FRONT END

The model front end is a special, CORDIVEM-specific module. It translates the generic order number and situation parameters in the COMRD vector (generated inside CORDIVEM) into the specific order format required by the SIMSCRIPT Engineer Module. The basic design strategy was:

1. CORDIVEM will not set priorities or create situation numbers for the SIMSCRIPT module.

2. CORDIVEM will send to the Engineer Module a command vector, including parameters specifying the requesting unit's situation and the engineer job requested.

3. CORDIVEM will issue only simple generic orders to the Engineer Module (e.g., blow bridges, crater road, fortify, emplace minefield).

4. The SIMSCRIPT Engineer Module front end will choose parameters from the command vector in order to:

a. Select single job order/multiple job order.

b. Set a default priority.

c. Generate a situation number for task/technique grading/selection.

These three values are passed to the interpret order section of the Engineer Module, along with such things as supported unit number, battlefield locations, etc. Thus, the front end supplies all information required for an Engineer Module order.

When improvements or refinements are needed, the subroutines embedded in the ICOR implementation will not need changing; only the front end section of the SIMSCRIPT program or the front end data will have to be changed.

The front end does the following:

1. Inputs a CORDIVEM command vector, COMORD (see below), which includes a generic order number (e.g., 902 for crater road), terrain data, work site location, supported unit, etc.

2. Finds the generic order in the front end data tables, yielding the method number for Steps 3, 4, and 5, and the list of orders for Step 3.

3. Selects a terrain-specific multiple job order (e.g., crater two-lane road).

4. Calculates a priority.

5. Calculates a situation number.

6. Invokes the Engineer Module interpret order code.

The data for the front end consist of the following items for each generic order:

1. The generic order number issued by CORDIVEM (e.g., 902 for crater road or 901 for prepare battle position).

2. A list of multiple job order names. Most orders will have several possibilities depending on terrain (e.g., crater road depends on the width of the road).

3. An order selection method number. This specifies the algorithm for selecting from the list of orders given the terrain factors.

4. A priority calculation method number.

5. A situation generation method number. Different orders may need to use different factors in order to generate a situation number.

8 MIDAS CHANGES

MIDAS is a data definition language and FORTRAN preprocessor used to describe and access data stored in FORTRAN common areas. This section lists the new data structures added and modifications to existing structures.

The following MIDAS structures have been altered:

Structure	*Description*
SCORBRD	Unit killer/victim scoreboard
ADDRESS	Hex terrain attributes

And the following new structures have been added:

Structure	*Description*
EMKNTRL	Basic control structure
DLYPATH	Scores MITL defensive path
EMORDER	Engineer order data structure

SCORBRD—Modifications

The SCORBRD data structure defines each combat unit in CORDIVEM. The additional attributes support treating an engineer work site as a unit (enabling attrition at the work site), as well as providing links to the engineer data structures. See Table 3.

ADDRESS—Modifications

The ADDRESS data structures store the terrain data for each box. The new data attributes listed below represent engineer effects. See Table 4.

EMKNTRL—New MIDAS Structure

Every unit receiving engineer support owns an instance of EMKNTRL, which links the various structures required to support the unit. See Table 5.

DLYPATH—New MIDAS Structure

DLYPATH stores the MITL-supplied delay path. Each DLYPATH instance stores one point in the path. Maneuver units follow this path when forced to move by threat pressure. See Table 6.

Table 3
SCORBRD Structures

Field	Size	Purpose
MVIGNR	1	= 1 if unit should be ignored by the ICOR move selection algorithm; CORDIVEM may use this bit for headquarters units
EMSUTYPE	2	Type of unit engineer support = 0 if unit has never received engineer support = 1 do not issue engineering orders automatically for unit = 2 if unit has received systemic engineer support = 3 if engineer work site
PEMKNTL	16	Pointer to the EMKNTRL structure for unit
	12	Reserved for expansion

NOTE: Engineer work site units are not added to the C2 tree.

Table 4
ADDRESS Structures

Field	Size	Purpose
OBSTCL	18	Obstacle hex side value (three bits/hexside); represents countermobility effects (tank ditch, rubble)
IFRTFY	3	Fighting position level of hex interior
	11	Reserved for expansion

NOTE: Currently no Red or Blue indicators exist.

Table 5
EMKNTRL Structures

Field	Size	Purpose
KKDLY	2	Unused, reserved for enhancements
PDPTH	16	Pointer to head of the list of DLYPTH structures (may be empty, if no MITL entries)
HEXST	32	Short-term objective on delay path
HEXLT	32	Long-term objective on delay path
PEMORD	16	Pointer to head of the list of outstanding engineer orders for the unit
	30	Reserved for expansion

Table 6
DLYPATH Structures

Field	Size	Purpose
PDPTH	16	Link to next DLYPATH structure in the path
KKGMR	2	Gamer control data (see definition below)
HEXOBJ	32	Hex number of this point
	14	Reserved for expansion

Tentative definition of KKGMR:

= 0 Default (systemic generation of fighting position
= 1 No systemic action
= 2 Use default mechanism, but give position a higher priority
= 3 This is a "strong point"; use systemic mechanism to issue fighting position and linear obstacle orders at high priority

Table 7
EMORDER Structures

Field	Size	Purpose
LINKWS	16	Next EMORDER on work site chain
LINKU	16	Next EMORDER on issuing unit chain
PSBWS	16	Pointer to work site (SCORBRD)
PSBUNIT	16	Pointer to SCORBRD of unit issuing the order
IORDNN	12	Generic order number (e.g., 902 for crater)
IDIR	3	Hex side where target is to be placed and the direction used for one deferred target execution algorithm
K1	4	Target execution class
K2	4	Target type
V1	4	Total number of completion points (RANGE 0-1000) received from *SIMSCRIPT* Engineer Module
V2	4	Total number of completion points represented by terrain modifications
IDELTA	4	Amount hex attributes have been changed

NOTE: EMORDER requires modification to handle several units requesting the same engineer order at the same location.

EMORDER—New MIDAS Structure

EMORDER stores data for orders issued, but not completed, and for unexecuted targets. Each order is chained to the work site and to the requesting unit. See Table 7.

9 FORTIFICATION TASKS

The amount (and kind) of engineering effort required to create a standard prepared position varies according to the terrain. The model front end uses two parameters to choose the proper job order: (1) the sum of the hex interior attributes (built-up, forestation, and mountain), and (2) as a special case, a built-up value of 3. See Table 8.

Table 8
Fortification Tasks

Sum	*Minefield*	*Ditch*	*Tank Defilades*	*Other*
Bulit-up = 3	1000 m	--	--	29 craters 9 rubble
8	800 m	--	25	21 craters 7 abati 1000-m trail
5-7	1000 m	500 m	25	500-m trail 12 craters 6 abati
3-4	1000 m	1000 m	25	9 craters 33 abati
0-2	2000 m	1800 m	30	---

10 CORDIVEM ENGINEER MODULE COMMUNICATION DATA STRUCTURES

COMORD

The COMORD vector is passed from the ICOR code to the model front end via the event structure in the GEM. The vector is 15 words long with the structure shown in Table 9.

Generic Engineer Order Numbers

These numbers specify the class of engineering work being performed (or requested).

Value	*Description*
901	Prepare battle positions
902	Crater road
903	Blow bridge
904	Install linear obstacle

COMORD Command

The command number occurs in the COMORD data vector sent to the engineer resource allocation module.

Value	*Meaning*
1	Issue order
2	Recalculate priority
3	Recalculate, but do not lower, priority
4	Cancel order

Table 9
COMORD Structure

Word	
1	COMORD command (see below)
2	Address of EMORD
3	Engineer order class (see below)
4	Road width: RANGE 0..3
5	River width: RANGE 0..3
6	ICOR unit number
7	X location: REAL
8	Y location: REAL (Note: X, Y values computed by ICOR routine HAZXYL)
9	
10	Distance between work site and supported unit (calculated by ICOR routine MDIST)
11	Terrain word
12	Order source (= 0 game generated, = 1 MITL)
13	Order class (= 0 countermobility on defensive path, = 1 halt point on defensive path, = 2 prepared position on defensive path)
14	Combat danger level: RANGE 0..4 (= 0 no contact, = 4 close combat, flank threat)
15	Direction: RANGE 1..6 (hex side number)

11 CORDIVEM ROUTINE—MODIFICATIONS AND USE

CORDIVEM Utility Routines Invoked

The Engineer Module uses the following ICOR utility routines. These routines have not been altered, but supply utility services to the Engineer Module subroutines. See Table 10.

Modifications to Existing CORDIVEM Routine CONSIDR

1. When the Operation Reaction System (ORS) requires a new objective, CONSIDR now calls DLYPCR to create/update the delay path and LDJST to load the short-term objective as the unit's objective.

2. CONSIDR will call DLYPCR for a unit in contact with enemy forces, if the unit has an OP order and no delay path. This provides lead time so that combat engineering work can be completed before it is needed.

Note: Current implementation restricted to Blue forces only.

Modifications to CORDIVEM Routines MOVSCHD and MOVRCT

Both routines call EMUCHK to determine if deferred targets (like blowing bridges) can be executed.

1. MOVRCT calls EMUCHK when the unit arrives in a new hex.

2. MOVSCHD calls EMUCHK after calculating the movement time to the next hex. Thus, a Blue unit will get the advantage of a bridge or an uncratered road, but a Red unit that schedules a movement to pursue Blue will not.

12 CONCLUSION

This report gives a brief description, in military terms, of what the Engineer Module does. It also gives information on the module's program structure, engineer data structures, model front end, and MIDAS structures for use by the CORDIVEM/Engineer Module integration staff.

Table 10
CORDIVEM Utility Routines

Routine	Description
HXDIST (HEXA, HEXB)	Returns the number of hexes between HEXA and HEXB
HXADD (HEX, HV)	Returns the sum of the two HEX numbers
HXSUB (HXA, HXB)	Returns the difference between the two hex numbers
HXNEXT (HEXA, HEXFNL)	Returns the next hex number in the path from HEXA to HEXFNL
HXLINE (HEXA, HEXB, N)	Returns hex *N* hexes along a line between HEXA and HEXB
HXMOVE (HEXNUM, HEXFNL, HXA, HXB)	Given a hex position (HEXNUM) and an end point (HEXFNL), sets HEXA and HEXB to the two possible hexes next to HEXNUM on the path to HEXFNL (HEXA may equal HEXB if a line toward HEXFNL bisects a hex side)
MDIST (HEXA, HEXB)	Returns the distance (in meters) between the centers of the hexes
GIMME (P, LB)	ICOR space allocation routine for MIDAS data structures

Chief of Engineers
ATTN: Tech Monitor
ATTN: DAEN-ASI-L (2)
ATTN: DAEN-CCP
ATTN: DAEN-CW
ATTN: DAEN-CWE
ATTN: DAEN-CWM-R
ATTN: DAEN-CWO
ATTN: DAEN-CWP
ATTN: DAEN-MP
ATTN: DAEN-MPC
ATTN: DAEN-MPE
ATTN: DAEN-MPO
ATTN: DAEN-MPR-A
ATTN: DAEN-RD
ATTN: DAEN-RDC
ATTN: DAEN-RDM
ATTN: DAEN-RM
ATTN: DAEN-ZC
ATTN: DAEN-ZCE
ATTN: DAEN-ZCI
ATTN: DAEN-ZCM

FESA, ATTN: Library 22060

FESA, ATTN: DET III 79906

US Army Engineer Districts
ATTN: Library
Alaska 99501
Al Batin 09616
Albuquerque 87103
Baltimore 21203
Buffalo 14207
Charleston 29402
Chicago 60604
Detroit 48231
Far East 96301
Fort Worth 76102
Galveston 77550
Huntington 25721
Jacksonville 32232
Japan 96343
Kansas City 64106
Little Rock 72203
Los Angeles 90053
Louisville 40201
Memphis 38103
Mobile 36628
Nashville 37202
New Orleans 70160
New York 10007
Norfolk 23510
Omaha 68102
Philadelphia 19106
Pittsburgh 15222
Portland 97208
Riyadh 09038
Rock Island 61201
Sacramento 95814
San Francisco 94105
Savannah 31402
Seattle 98124
St. Louis 63101
St. Paul 55101
Tulsa 74102
Vicksburg 39180
Walla Walla 99362
Wilmington 28401

US Army Engineer Divisions
ATTN: Library
Europe 09757
Huntsville 35807
Lower Mississippi Valley 39180
Middle East 09038
Middle East (Rear) 22601
Missouri River 68101
New England 02154
North Atlantic 10007
North Central 60605
North Pacific 97208
Ohio River 45201
Pacific Ocean 96858
South Atlantic 30303
South Pacific 94111
Southwestern 75202

US Army Europe
HQ, 7th Army Training Command 09114
ATTN: AETTG-DEH (5)
HQ, 7th Army ODCS/Engr. 09403
ATTN: AEAEN-EH (4)
V. Corps 09079
ATTN: AETVDEH (5)
VII. Corps 09154
ATTN: AETSDEH (5)
21st Support Command 09325
ATTN: AEREH (5)
Berlin 09742
ATTN: AEBA-EN (2)
Southern European Task Force 09168
ATTN: AESE-ENG (3)
Installation Support Activity 09403
ATTN: AEUES-RP

8th USA, Korea
ATTN: EAFE (8) 96301
ATTN: EAFE-Y 96358
ATTN: EAFE-ID 96224
ATTN: EAFE-4M 96208

8th USA, Korea
ATTN: EAFE-H 96271
ATTN: EAFE-P 96259
ATTN: EAFE-T 96212

ROK/US Combined Forces Command 96301
ATTN: EUSA-HHC-CFC/Engr

USA Japan (USARJ)
Ch, FE Div, AJEN-FE 96343
Fac Engr (Honshu) 96343
Fac Engr (Okinawa) 96331

Rocky Mt. Area 80903

Area Engineer, AEDC-Area Office
Arnold Air Force Station, TN 37389

Western Area Office, CE
Vandenberg AFB, CA 93437

416th Engineer Command 60623
ATTN: Facilities Engineer

US Military Academy 10996
ATTN: Facilities Engineer
ATTN: Dept of Geography & Computer Science
ATTN: DSCPER/MAEN-A

Engr. Studies Center 20315
ATTN: Library

AMMRC, ATTN: DRXMR-WE 02172

USA ARRCOM 61299
ATTN: DRCIS-RI-I
ATTN: DRSAR-IS

DARCOM - Dir., Inst., & Svcs.
ATTN: Facilities Engineer
ARRADCOM 07801
Aberdeen Proving Ground 21005
Army Matls. and Mechanics Res. Ctr.
Corpus Christi Army Depot 78419
Harry Diamond Laboratories 20783
Dugway Proving Ground 84022
Jefferson Proving Ground 47250
Fort Monmouth 07703
Letterkenny Army Depot 17201
Natick R&D Ctr. 01760
New Cumberland Army Depot 17070
Pueblo Army Depot 81001
Red River Army Depot 75501
Redstone Arsenal 35809
Rock Island Arsenal 61299
Savanna Army Depot 61074
Sharpe Army Depot 95331
Seneca Army Depot 14541
Tobyhanna Army Depot 18466
Tooele Army Depot 84074
Watervliet Arsenal 12189
Yuma Proving Ground 85364
White Sands Missile Range 88002

DLA ATTN: DLA-WI 22314

FORSCOM
FORSCOM Engineer, ATTN: AFEN-FE
ATTN: Facilities Engineer
Fort Buchanan 00934
Fort Bragg 28307
Fort Campbell 42223
Fort Carson 80913
Fort Devens 01433
Fort Drum 13601
Fort Hood 76544
Fort Indiantown Gap 17003
Fort Irwin 92311
Fort Sam Houston 78234
Fort Lewis 98433
Fort McCoy 54656
Fort McPherson 30330
Fort George G. Meade 20755
Fort Ord 93941
Fort Polk 71459
Fort Richardson 99505
Fort Riley 66442
Presidio of San Francisco 94129
Fort Sheridan 60037
Fort Stewart 31313
Fort Wainwright 99703
Vancouver Bks. 98660

HSC
ATTN: HSLO-F 78234
ATTN: Facilities Engineer
Fitzsimons AMC 80240
Walter Reed AMC 20012

INSCOM - Ch, Instl. Div.
ATTN: Facilities Engineer
Arlington Hall Station (2) 22212
Vint Hill Farms Station 22186

MDW
ATTN: Facilities Engineer
Cameron Station 22314
Fort Lesley J. McNair 20319
Fort Myer 22211

MTMC
ATTN: MTMC-SA 20315
ATTN: Facilities Engineer
Oakland Army Base 94626
Bayonne MOT 07002
Sunny Point MOT 28461

NARADCOM, ATTN: DRDNA-F 071160

TARCOM, Fac. Div. 48090

TRADOC
HQ, TRADOC, ATTN: ATEN-FE
ATTN: Facilities Engineer
Fort Belvoir 22060
Fort Benning 31905
Fort Bliss 79916
Carlisle Barracks 17013
Fort Chaffee 72902
Fort Dix 08640
Fort Eustis 23604
Fort Gordon 30905
Fort Hamilton 11252
Fort Benjamin Harrison 46216
Fort Jackson 29207
Fort Knox 40121
Fort Leavenworth 66027
Fort Lee 23801
Fort McClellan 36205
Fort Monroe 23651
Fort Rucker 36362
Fort Sill 73503
Fort Leonard Wood 65473

TSARCOM, ATTN: STSAS-F 63120

USACC
ATTN: Facilities Engineer
Fort Huachuca 85613
Fort Ritchie 21719

WESTCOM
ATTN: Facilities Engineer
Fort Shafter 96858

SHAPE 09055
ATTN: Survivability Section, CCB-OPS
Infrastructure Branch, LANDA

HQ USEUCOM 09128
ATTN: ECJ 4/7-LOE

Fort Belvoir, VA 22060
ATTN: ATZA-DTE-EM
ATTN: ATZA-DTE-SW
ATTN: ATZA-FE
ATTN: Engr. Library
ATTN: Canadian Liaison Office (2)
ATTN: IWR Library

Cold Regions Research Engineering Lab 03755
ATTN: Library

ETL, ATTN: Library 22060

Waterways Experiment Station 39180
ATTN: Library

HQ, XVIII Airborne Corps and 28307
Ft. Bragg
ATTN: AFZA-FE-EE

Chanute AFB, IL 61868
3345 CES/DE, Stop 27

Norton AFB 92409
ATTN: AFRCE-MX/DEE

Tyndall AFB, FL 32403
AFESC/Engineering & Service Lab

NAFEC
ATTN: RDT&E Liaison Office
Atlantic Division 23511
Chesapeake Division 20374
Southern Division 29411
Pacific Division 96860
Northern Division 19112
Western Division 64066
ATTN: Sr. Tech. FAC-03T 22332
ATTN: Asst. CDR R&D, FAC-03 22332

NCEL 93041
ATTN: Library (Code L08A)

Defense Technical Info. Center 22314
ATTN: DDA (12)

Engineering Societies Library 10017
New York, NY

National Guard Bureau 20310
Installation Division

US Government Printing Office 22304
Receiving Section/Depository Copies (2)

Resource Allocation Team Distribution

Chief of Engineers
ATTN: DAEN-MPO-B
ATTN: DAEN-MPR
ATTN: DAEN-MPZ-A
ATTN: DAEN-ZCI
ATTN: DAEN-ZCP

US Army Engineer District
New York 10007
ATTN: Chief, Design Br
Pittsburgh 15222
ATTN: Chief, Engr Div
Philadelphia 19106
ATTN: Chief, NAPEN-D
Baltimore 21203
ATTN: Chief, Engr Div
Norfolk 23510
ATTN: Chief, NAOEN-D
Wilmington 28401
ATTN: Chief, SAWEN-PP
ATTN: Chief, SAWEN-PM
Charleston 29402
ATTN: Chief, Engr Div
Savannah 31402
ATTN: Chief, SASAS-L
Jacksonville 32232
ATTN: Env Res Br
Mobile 36128
ATTN: Chief, SAMEN-D
Nashville 37202
ATTN: Chief, ORNED-P
Memphis 38103
ATTN: Chief, LMMED-P
Vicksburg 39180
ATTN: Chief, Engr Div
Louisville 40201
ATTN: Chief, Engr Div
St Paul 55101
ATTN: Chief, ED-D
Chicago 60604
ATTN: Chief, NCCPD-ER
St Louis 63101
ATTN: Chief, ED-B
ATTN: Chief, ED-D
Kansas City 64106
ATTN: Chief, Engr Div
Omaha 68102
ATTN: Chief, Engr Div
Tulsa 74102
ATTN: Chief, Engr Div
Albuquerque 87103
ATTN: Chief, Engr Div
San Francisco 94105
ATTN: Chief, Engr Div
New Orleans 70160
ATTN: Chief, LMNED-DG
Sacramento 95814
ATTN: Chief, SPKED-D
Far East 96301
ATTN: Chief, Engr Div
Seattle 98124
ATTN: Chief, NPSEN-PL-ER
Walla Walla 99362
ATTN: Chief, Engr Div
Alaska 99501
ATTN: Chief, NPASA-R

US Army Engineer Division
New England 02154
ATTN: Chief, NEDED-T
North Atlantic 10007
ATTN: Chief, NADEN-T
Middle East (Rear) 22601
ATTN: Chief, MEDED-T
South Atlantic 30303
ATTN: Chief, SADEN-TS
Huntsville 35807
ATTN: Chief, HNDED-CS
Lower Mississippi Valley 39180
ATTN: Chief, PD-R
Ohio River 45201
ATTN: Chief, Engr Div
North Central 60605
ATTN: Chief, Engr Div

US Army Engineer Division
Southwestern 75202
ATTN: Chief, SWDED-TA
South Pacific 94111
ATTN: Chief, SPDED-TG
Pacific Ocean 96858
ATTN: Chief, Engr Div
ATTN: Chief, PODED-M
North Pacific 97208
ATTN: Chief, Engr Div

6th US Army 94129
ATTN: AFKC-EN

7th US Army 09407
ATTN: AETTM-HRD-EHD

West Point, NY 10996
ATTN: Dept of Mechanics
ATTN: Library

Ft. Belvoir, VA 22060
ATTN: Learning Resources Center
ATTN: ATSE-TD-TL (2)
ATTN: British Liaison Officer (5)

Ft. Clayton Canal Zone 34004
ATTN: DFAE

Ft. Leavenworth, KS 66027
ATTN: ATZLCA-SA

Ft. Lee, VA 23801
ATTN: DRXMC-D (2)

Ft. McPherson, GA 30330
ATTN: AFEN-CD

Ft. Monroe, VA 23651
ATTN: ATEN-AD (3)
ATTN: ATEN-C

Ft. Richardson, AK 99505
ATTN: AFZT-FE-E

Naval Facilities Engineer Command
ATTN: Code 04
Alexandria, VA 22332

NCEL 93043
ATTN: Morell Library

Patrick AFB, FL 32925
ATTN: XRQ

Tinker AFB, OK 73145
2854 ABG/DEEE

Tyndall AFB, FL 32403
AFESC/PRT

Building Research Advisory Board 20418

Transportation Research Board 20418

Airports and Construction Services Dir
Ottawa, Ontario, Canada K1A 0N8

Division of Building Research
Ottawa, Ontario, Canada K1A 0R6

National Defense Headquarters
Ottawa, Ontario, Canada K1A 0K2

Ms. Dee Scholes
Computer Science Corporation
Defense Systems Division
304 W. Route 38
Morristown, NJ 08057

Engineer Study Center
ATTN: Doug Lehman
6500 Brooke Lane
WASH DC 20315

Commander, TRADOC
ATTN: ATCD-A/Mark Murray
Fort Monroe, VA 23651

Director, AMSAA
ATTN: DRXSY-GB/Lee Meredith
Aberdeen Proving Grounds, MD 21005

CDR USACE (DAEN-DRD/John Henry)
WASH DC 20314

USA TRASANA
ATTN: ATAA-TGM/Keith Thorpe
White Sands Missile Range, NM 88002

Commander, U.S. Army Engineer School
ATTN: ATEA-CDE/CPT Cliff Clausen
Fort Belvoir, VA 22060

CAC & Fort Leavenworth
ATTN: ATZL-CAC-A
Fort Leavenworth, KS 66027

Director, U.S. Army Concepts Analysis Agency
CSCA-FAS (Major Alvord)
8120 Woodmont
Bethesda, MD 20014

Pentagon
SAUS-OR/Herb Fallin
Room 2E 621
WASH DC 20310

Gerald W. Turnage
Mobility Systems Division
Geotechnical Laboratory
USA Waterways Experiment Station
P.O. Box 631
Vicksburg, MS 39180

U.S. Army Engineer District, Memphis
ATTN: CPT Van Horn
668 Clifford Davis Federal Building
Memphis, TN 38103

CPT Edward Wright
Executive Officer
HQ 2nd BN, 2nd Brigade
Fort Leonard Wood, MO 65475

Department of the Army
HQ, USA, Europe, and 7th Army
Office of the Deputy Chief of Staff
ATTN: Mr. A. Fleming
APO New York 09403

Engineer modeling study. -- Champaign, Ill : Construction Engineering
Research Laboratory ; available from NTIS.
3v. (Technical report / Construction Engineering Research Laboratory ;
p-131)

Contents: v.1. Executive summary / by John Evans. -- v.2. Users
manual / by Gerald Brown, Hugh Henry. -- v.3. CORDIVEM/Engineer Module
Interface manual / by Carlton Mills.

1. War games. 2. Military engineering -- models. I. Evans, John.
II. Brown, Gerald J. III. Henry, Hugh IV. Mills, Carlton. V. Series:
Technical report (Construction Engineering Research Laboratory (U.S.));
P-131.