FEB 80 version

AD A 124333

THE BDM CORPORATION

INTEGRATED NUCLEAR AND CONVENTIONAL THEATER WARFARE SIMULATION (INWARS) DOCUMENTATION

PART I

SYNOPSIS

(1)

THE BDM CORPORATION

7915 Jones Branch Drive
McLean, Virginia 22102
Phone (703) 821-5000

February 8, 1980

BDM/W-80-047-TR

INTEGRATED NUCLEAR AND CONVENTIONAL THEATER WARFARE SIMULATION (INWARS) DOCUMENTATION

PART I

SYNOPSIS

80W/00102

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A124333	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Final-Integrated Nuclear and Conventional Theater Warfare Simulation (INWARS). Final Report-Part I, Synopsis		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER BDM/W-80-047-TR
7. AUTHOR(s) Dr. J.R. Aldrich, J.B. Gilmer		8. CONTRACT OR GRANT NUMBER(s) DAAG39-77-C-0174
9. PERFORMING ORGANIZATION NAME AND ADDRESS The BDM Corporation, 7915 Jones Branch Drive, McLean, VA 22101		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS HQ DA (DAMO-ZD), The Pentagon, Room 3A546, Washington, D.C. 20310		12. REPORT DATE February 8, 1980
		13. NUMBER OF PAGES 49
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)

Unlimited

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

Unlimited

18. SUPPLEMENTARY NOTES

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

Theater Simulation
Command-Control-Intelligence
Combat Interactions

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

This report describes the modeling and software elements of the INWARS theater-level automated simulation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This volume constitutes the Synopsis Component of the Integrated Nuclear and Conventional Theater Warfare Simulation (INWARS) documentation. It provides an overview of the simulation in terms of unique features, inputs and outputs, and modes of application. The INWARS representation of theater warfare and its software implementation are then synopsized, and a guide to the remaining three components of the INWARS documentation is provided.

PART I - INWARS SYNOPSIS

PART II - MODELING DESCRIPTION

- VOL. I INTRODUCTION
- VOL. II GROUND COMBAT OPERATIONS
- VOL. III AIR OPERATIONS
- VOL. IV COMBAT SUPPORT OPERATIONS
- VOL. V ECHELON ABOVE DIVISION COMMAND, CONTROL, AND INTELLIGENCE (EAD C^2I) ACTIVITIES

PART III - SOFTWARE DESCRIPTION

- VOL. I SOFTWARE FRAMEWORK
- VOL. II EAD C^2I DATA STRUCTURES
- VOL. III EAD C^2I PROCEDURES
- VOL. IV INFORMATION COLLECTION AND COMMUNICATION DATA STRUCTURES AND PROCEDURES
- VOL. V COMBAT INTERACTIONS DATA STRUCTURES
- VOL. VI COMBAT INTERACTIONS PROCEDURES

PART IV - USER'S MANUAL

- VOL. I INTRODUCTION
- VOL. II COMBAT INTERACTIONS INPUTS
- VOL. III EAD C^2I INPUTS
- VOL. IV INWARS OUTPUTS

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I
AN OVERVIEW OF INWARS

A. INTRODUCTION

This volume presents a broad synopsis of the Integrated Nuclear and Conventional Theater Warfare Simulation (INWARS) developed by the BDM Corporation for the U.S. Army under Contract DAAG39-77-C-0174. INWARS has been developed to provide a tool for investigating interactions among conventional, nuclear, and chemical operations in the context of a theater-level conflict situation. In this respect, it may be regarded as an extension of previous studies concerning methods for simulating integrated warfare (most recently, N. Farrell, P. H. Lowry and J. E. Shepherd, Method for Integrated Simulation (MINTSIM), Report OAD-CR-142, January 1976). Unlike these earlier studies, (and, for that matter, other simulations of theater level conflict), INWARS is explicitly designed around the force elements to be simulated rather than, e.g., geographic sectors. Thus, individual force elements from theater down through brigade levels "exist" within the simulations; indeed, it is through the dynamic actions and interactions of these force elements that each INWARS run evolves.

INWARS is also distinguished by its focus on upper-echelon command, control, and intelligence (C^2_uI) processes. In particular, INWARS contains explicit, fully-automated representations of the C^2_uI activities involved in: (1) developing, implementing, and executing operations to achieve assigned objectives; (2) considering the employment of conventional, nuclear, or chemical weapons in support of those operations; and, (3) adapting ongoing activities to the perceived threat of enemy nuclear or chemical attacks. Since these activities are driven by generalized doctrines and policies supplied as user-inputs, INWARS can support investigations of alternative doctrinal "approaches".

These and other aspects of INWARS are surveyed over the remainder of this Chapter in terms of its distinguishing features of INWARS, its structure and operation, and the broad areas for its application. Chapters II and III provide overviews of the INWARS representation of theater-level conflict and the software which realizes that representation. Chapter IV previews the remainder of the INWARS Documentation which, synopsis, includes three major components: the INWARS Modeling Description component (5 volumes); the INWARS Software Description component (6 volumes); and, the INWARS User's Manual component (4 volumes).

B. INWARS FEATURES

The development of INWARS has been guided by the general objectives and specifications highlighted in Figure I-1. The translation of these objectives and specifications into a simulation design has evolved from interaction with the Army Study Advisory Group (SAG) and informal Working Groups. During this process, various other design considerations emerged. These are also presented in Figure I-1.

467 Subroutines
31 Elements
7 Files

56K of Main Memory (Data base + Arrays)
+ 40K of Main Memory (Library + I banks)

OBJECTIVES

- Evaluating military and political constraints on conventional, chemical, and nuclear operations.
- Identifying interactions among conventional, chemical, and nuclear operations.
- Measuring impact on conventional war of the threatened use of nuclear weapons.
- Evaluating significant decision options at theater, army group, and corps levels.
- Developing typical combat situations within which corps and division level models can be applied for more detailed analyses.

HIGHLIGHTS OF SPECIFICATIONS

- Utilization of existing components & design concepts wherever possible.
- Emphasis on fast response and simplicity of application.
- Provision of decision processes to determine allocation of resources, employment of forces, and missions for each headquarters above division level.
- Consideration of transition between levels of integrated warfare and implementation of national strategies and doctrines.
- Reflection of differences in weapon and unit employment doctrines, particularly in regard to nuclear and chemical weapons.
- Representation of ground combat processes and interactions, air support, and combat service support.
- Fully automated operation.

OTHER CONSIDERATIONS

- Deterministic simulation.
- Installation on UNIVAC 1108

Figure I-1 Principal Influences on INWARS Development

The particular "character" of INWARS has resulted from the elaboration of--and tradoffs among--these objectives, specifications, and other considerations. This "character" is manifested in four broad characteristics: (1) emphasis on breadth and interactions of the combat operations processes represented, (2) focus on command, control, and intelligence (C^2I) processes, (3) preference for data-driven representations, and (4) provision for modification and growth of the software.

1. Emphasis on Breadth and Interactions

Perhaps the dominant characteristic of INWARS is that it is a model of interactions. Of principal concern in INWARS is assessing the joint influence of a broad spectrum of processes on the overall evolution of a theater-level conflict situation. Of less concern is the detailed investigation of any particular process within the context of an ongoing conflict situation. This facet is manifested both in the range of processes treated in INWARS and in the highly aggregated representations by which they are treated. It is primarily a consequence of the range of processes specified for treatment and the desire for fast response.

2. Focus on Command, Control, and Intelligence (C^2I)

The focus on C^2I processes in INWARS is directly discernable in the range of C^2I capabilities needed. These include significant decision options and constraints regarding resource allocation, force employment, and weapons employment in integrated operations at all echelons above division, all in a fully automated, deterministic simulation. The problem this poses concerns the comprehensive representation of specific decision-making processes, the context within which the specific decisionmaking processes occur, and the processes by which a C^2I element becomes aware of the need for a decision. Of these three aspects, the latter two (context and awareness) are the more difficult. They are essentially "architectural" in character, concerning how to integrate information and decision processes into a coherent simulation of C^2I behavior. The INWARS solution to this problem has relied on a "knowledge-based" approach as discussed below and in Chapter II, Section D of this Synopsis.

3. Preference for Data-Driven Representations

To achieve the development objectives and specifications, the user must have considerable influence on most aspects of the simulated force elements' operations--constraints, decision options, resource allocations, and so forth. Since this influence must be exercised by user-inputs, the INWARS design has given preference to "data-driven" representations, especially in the treatment of C^2I processes. Thus, much of the C^2I inference and decision-making logic is directly specified by input data rather than being "hard-wired" in code: for example, C^2I elements develop and execute operations based on user-input concepts of operation and constraints. This preference gives the user the control necessary to explore a variety of doctrines and policies.

4. Provision for Evaluation and Growth

Given the exploratory nature of INWARS, both as regards internal representations (especially in the C^2I area) and objectives for intended use, it is highly likely that areas for enhancement and refinement will be identified in its early applications. To facilitate this, provisions for evolution and growth have been designed into INWARS. One important provision is the data-driven representations employed--in many respects, growth can be accomplished by changing data inputs. More fundamentally, the design emphasizes modularity, thus facilitating software changes when deemed necessary. This is especially true in the C^2I representations, where many of the basic decision procedures can be modified with few "side-effects" on other parts of the software.

C. THE STRUCTURE OF INWARS

To further characterize INWARS, this section surveys its structure in terms of overall architecture, force structure representation, space-time representation, and processes treated.

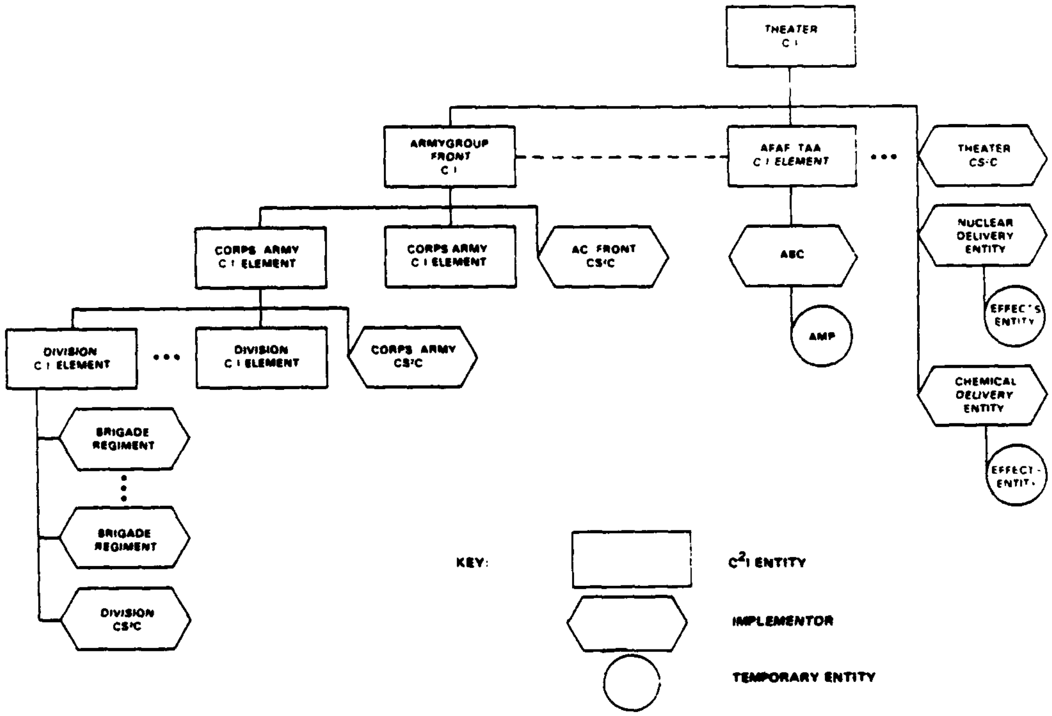
1. Architecture

INWARS has an entity-based, event-driven architecture. "Entity-based" means that all state information and simulation processes are structured around the entities of interest, namely, force elements of various types (e.g., brigades and regiments, higher level command elements, and so on). In a sense, force elements exist as distinct, independent individuals within INWARS: each has its own location, assets, objectives, expectations and other state information. Moreover, each force element operates independently in the simulation: each is guided in its operations by operation directives received from its commander, but the actual movements undertaken and engagements entered into also depend on the situation it faces.

"Event-driven" means that INWARS processes and interactions are structured around the occurrence of discrete events such as message receipt or C^2I activity (rather than, e.g., the continual passage of time). Thus, event occurences are the "points" at which the status of some or all of the entities--force elements--may change.

2. Force Structure Representation

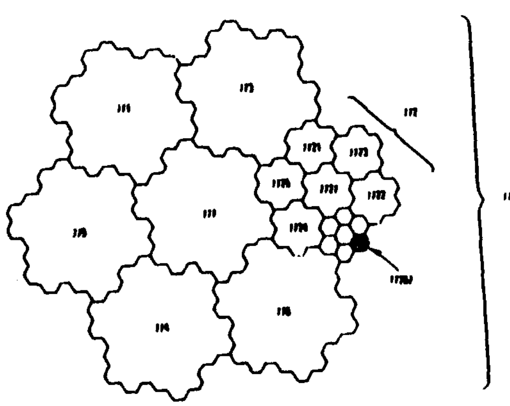
Given INWARS' entity-based architecture, its representation of force elements and structures is a central determinant of its overall scope and resolution. The organizational scope of INWARS is the theater--a theater level command is the highest echelon represented in INWARS. The organizational resolution of INWARS is the brigade/regiment--brigades and regiments are the lowest level entities which may be given orders, possess resources, move, inflict and sustain attrition, and be perceived and targeted by opposing force elements. Within this chain of command--theater down to brigade/regiment--INWARS treats five basic types of force elements: C^2I elements (theater through division headquarters/command posts), maneuver brigade/regiments, air base clusters (ABC), combat service support complexes (CS^2C) and nuclear/chemical delivery entities. In addition, two types of dynamically created entities are used in the model: Air Mission Packages (AMP) and Nuclear/Chemical Effects Entities. These entities are portrayed in chain-of-command form in Figure I-2.



80W/00102 Figure I-2. INWARS Entity Chain-of-Command

3. Space-Time Representation

Within the simulation, spatial positions are represented in terms of a hexagonal "coordinate system". This resolves the overall geography by means of a series of nested hexagonal decompositions as presented in Figure I-3. The largest hex in INWARS has a diameter of 8575 kilometers; the smallest hexes have a diameter of 9.45 kilometers. Thus, within the overall geography represented, spatial positions are essentially resolved to the nearest 10 kilometers. This spatial resolution has been chosen to be compatible with the brigade/regimental organizational resolution.



80W/00102

Figure I-3. Hex Structure

Given the event-driven architecture of INWARS, the passage of time is implicitly represented by the occurrence of events. However, each event has a specific occurrence time which is used to determine which of several events should occur "next". Upon the occurrence of an event, the simulated time in the conflict is simply advanced to that event's occurrence time. Occurrence times (and thus simulated or "game" times) are represented in terms of minutes since the start of the conflict.

4. Processes Treated

The particular structure of processes by which INWARS entities operate and interact within the simulation are illustrated in Figure I-4. These fall into three main classes: "physical" combat interactions processes, "interface" communications processes, and "mental" C^2I processes. This partitioning reflects the basic separation of physical and mental processes within INWARS--the two interact only via the explicit transmission of information as represented in the "interface" communications processes. Thus, INWARS exhibits a relatively wide scope of processes. Its process resolution, by contrast, is relatively low--each individual process is typically represented in a highly aggregated form.

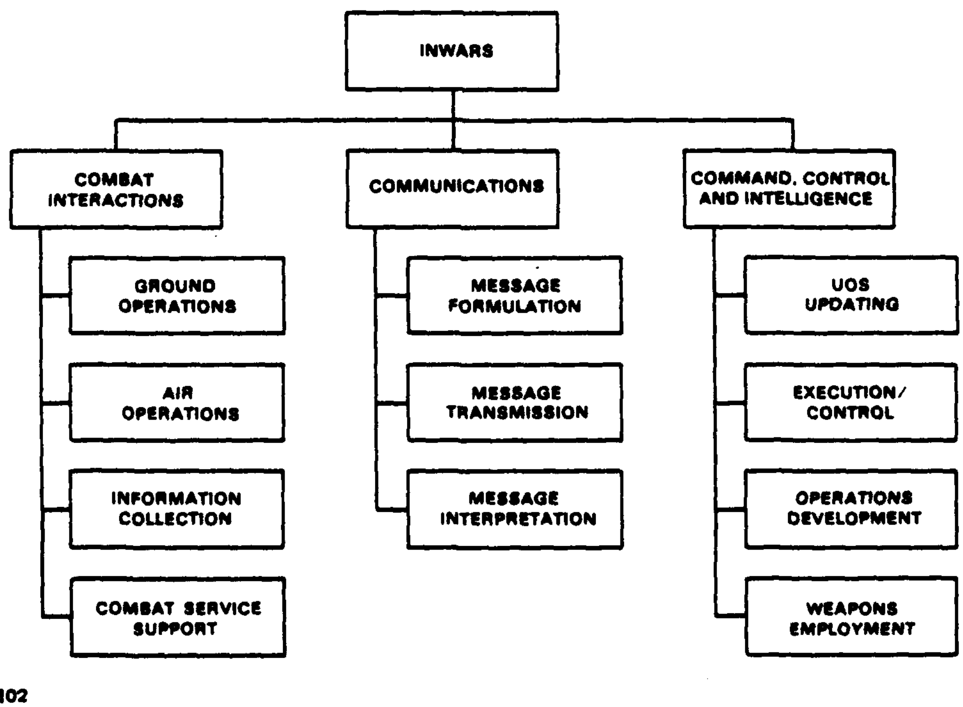


Figure I-4. Processes Represented in INWARS

D. THE OPERATION OF INWARS

The operation of INWARS--how the structure described above simulates a conflict--can be discussed in terms of user-inputs, the simulation process itself, and the outputs obtained. Figure I-5 illustrates the relationships among these components.

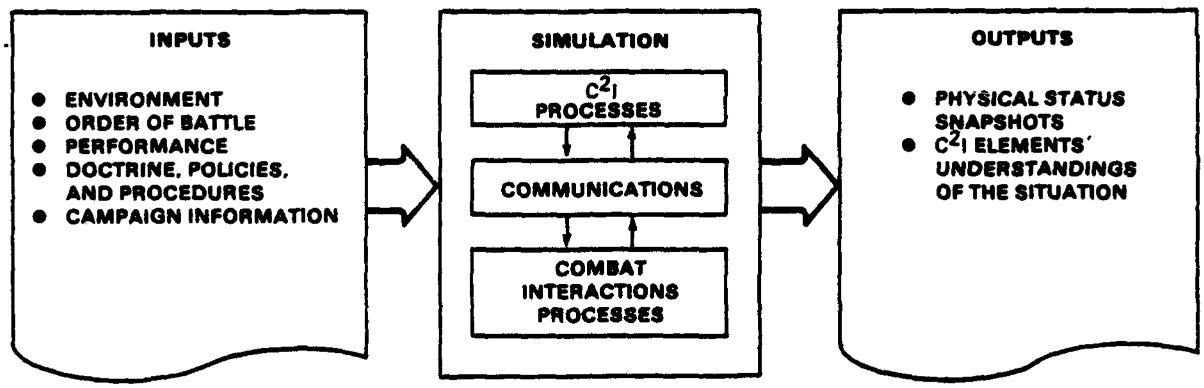


Figure I-5. INWARS Operations

1. User-Inputs

Inputs to INWARS are of five broad types: Environmental Information, which characterizes the physical terrain in which the force elements will interact; Order of Battle Information, which characterizes the conflicting force elements and structures in terms of composition, disposition, strength, tactics and other order of battle features; Performance Information, which characterizes the capabilities of force elements to undertake and carry out various types of activities and interactions; Doctrine, Policy, and Procedure Information, which characterizes the inference and decisionmaking processes of C^2I elements at echelons above division; and, Campaign Information, which characterizes the overall objectives, constraints, and initial configuration of the conflicting theater-level forces. Detailed discussion of inputs will be found in the User's Manual Component of the INWARS documentation.

Aside from the basic inputs to the simulation; the user may influence the course of a particular simulation run by "sending" appropriate types of directives to the theater-level C^2I elements. Thus, for example, if the user wishes to authorize the utilization of nuclear weapons after the third day in the conflict, he may include among the inputs a weapons employment directive authorizing a particular quantity of nuclear weapons as of the third day.

2. Simulation Flow

Once the inputs have been made, the simulation is initialized and begins to execute. The theater C^2I elements receive the user-input campaign directives and proceed to develop operations which will accomplish the assigned objective as using available forces and allocated resources. They implement these operations by sending operations directives to their subordinate army group/front C^2I elements. These C^2I elements then proceed to develop and implement operations by sending operations directives to their subordinates, the corps/army C^2I elements. This process continues on down through the chain-of-command until the brigade/regimental level force elements have been assigned objectives.

At this point, the brigades and regiments begin moving towards their assigned objectives. Eventually, the brigades and regiments of opposing forces may encounter each other and become engaged. As engagements occur, participating forces may sustain some attrition, and may react to the situation by changing objectives, changing operations, or requesting support (e.g., CAS). Force elements also make reports regarding their own status, the perceived status of enemy forces, and features of the situation. These reports are sent to the parent C^2I element.

Reports from subordinates provide the basis upon which higher echelon C^2I elements take actions to adapt their operations to the evolving situation. This may involve, for example, adjusting resource allocations among subordinates, committing reserves, modifying the ongoing operation, or even developing an entirely new operation to achieve assigned objectives. Also included in the higher echelon actions are the employment of conventional, nuclear, and/or chemical weapons. As such actions are undertaken and implemented by lower echelon force elements, the course of individual engagements and battles may be altered. To the extent that such alterations are reflected in reports from the engaged force elements, higher-echelon C^2I elements continue to take actions in pursuit of their objectives and the process continues. The simulation continues to run in this fashion until an end-of-game time is reached.

3. Outputs

The outputs produced in an INWARS simulation run may be collectively characterized as "state snapshots" presenting the status of some or all of the simulated force elements at a particular point in time. True "physical status" snapshots of all force elements are taken periodically and may be used to follow the actual course of the simulated conflict. Equally important are "perceived status" snapshots taken from the point of view of paritcular C^2I elements at echelons above division; these are taken periodically and also at key decision points of individual C^2I elements. Details will be found in the User's Manual component of the INWARS documentation.

E. THE APPLICATION OF INWARS

Like any simulation of complex interactions among a variety of entities, INWARS in no way purports to be predictive of what would "really happen" in a theater-level conflict. It must be regarded as a tool which analysts can use to investigate--and gain insight into--various problems and issues, perhaps in conjunction with other tools, but always in conjunction with their own judgement. Moreover, to the extent that INWARS meets its development objectives, it is not, in itself, a general purpose model, but is rather better suited to some types of problems and issues than others. Given the emphasis of scope over resolution, INWARS itself is not, for example, well-suited for evaluation of alternative major weapon systems (e.g., Tank A versus Tank B). Based on the development objectives, two principal analytical roles for INWARS application may be identified:

INWARS as a doctrine-policy simulator, and INWARS as a scenario generator. The first of these applications uses INWARS by itself, while the second uses INWARS in conjunction with other, more detailed, simulations.

1. INWARS as a Doctrine-Policy Simulation

The first four INWARS development objectives, (see Figure I-1, above) relate fundamentally to the impact of doctrines and policies on the overall course of a theater-level conflict. Consistent with these objectives, INWARS has been designed to allow the user to specify--via input data--a large portion of the doctrines, policies, constraints, and standard operating procedures by which the simulated force elements are to operate. These include, in particular:

(1) concepts of operation to guide the planning and execution of conventional operations;

(2) weapons-employment concepts and constraints to guide the utilization of conventional, nuclear, and chemical weapons; and,

(3) threat response policies, to guide the adaption of ongoing operations to changes in perceived nuclear and chemical threat.

INWARS C^2I elements at echelons above division apply these doctrines and policies during the simulation by dynamically "fitting" them to the (perceived) situations they face as they plan operations, consider weapons employment, and respond to nuclear and chemical threat. Thus, the actions of INWARS force elements reflect the generalized, user-input doctrines and policies as applied by the model itself in the specific situations which arise. Consequently, the course of the simulated conflict represents the dynamic interactions among the doctrines and policies of the opposing forces in the particular situation simulated.

To utilize INWARS in this doctrine-policy simulation role, an initial configuration of force elements with specific distribution of assets (major systems, etc.) would be postulated. Likewise, broad goals for the opposing theaters would be postulated. Finally, a range of alternative systems of doctrines and policies would be postulated, for one or both sides. A set of simulation runs would then be conducted, one for each of the alternative doctrine-policy systems under consideration. By comparing the evolution and outcomes of the resulting simulated conflicts, insights could be gained into impacts of the differing doctrines and policies. Of course, even here, the analysis must be tempered by the fact that the simulated C^2I elements are quite "doctrinaire": they cannot creatively conceive "new" doctrines and policies as a real commander might.

2. <u>INWARS as a Scenario-Generator</u>

The last INWARS development objective (see Figure I-1) relates to the use of INWARS in conjunction with more detailed corps and division level models. As the simulation is run and lower level entities-brigades and regiments--move towards their objectives and become engaged with one another, situations may arise which are of interest for more detailed analysis. The entity-based architecture of INWARS enables these situations to be taken from INWARS, mapped out, and used as a basis for inputs to correspondingly more detailed models. In this sense, INWARS can be used as a "scenario-generator" for more detailed models. Thus, for example, while INWARS itself is not well suited for detailed comparative evaluation of alternative major systems, it may be used to generate specific combat situations in which to evaluate the alternative systems with more detailed models. Moreover, since the situations generated are dependent on the user-specified doctrines and policies, the role of the alternative systems under different broad forms of operation may, to an extent, be explored.

CHAPTER II
INWARS REPRESENTATION OF INTEGRATED THEATER-LEVEL WARFARE

A. MODELING ARCHITECTURE

Consistent with INWARS' entity-based architecture, its modeling is concerned with the types of force elements to be included, their representation, the activities they can undertake within the model, and the processes by which the activities of several entities may interact to produce changes in their states. Some general features of the INWARS approach to these areas are presented in this section. Details will be found in the Modeling Description component of the INWARS documentation.

1. Entity Representation in INWARS

Eight different types of entities are treated in INWARS. As summarized in Figure II-1, these entity types are the basic "building blocks" which the user may use to construct--via input--force structures for a simulated conflict.

ENTITY TYPE	FUNCTIONAL DESCRIPTION
ECHELON ABOVE DIVISION (EAD) C^2I ELEMENT	PRINCIPAL DECISIONMAKERS WITH "FULL" C^2I RESPONSIBILITIES INCLUDING SITUATION UNDERSTANDING, OPERATIONS DEVELOMENT, WEAPONS EMPLOYMENT, AND EXECUTION/CONTROL
DIVISION C^2I ELEMENT	LIMITED DECISIONMAKERS WITH RESPONSIBILITY FOR SIMPLIFIED OPERATIONS DEVELOPMENT
GROUND MANEUVER ENTITY	PRINCIPAL IMPLEMENTOR OF GROUND OPERATIONS (MANEUVER BRIGADES AND REGIMENTS)
COMBAT SERVICE SUPPORT COMPLEX	AGGREGATE ENTITY RESPONSIBLE FOR SUPPLY, REPLACEMENT, MAINTENANCE/REPAIR, AND HOSPITALIZATION SUPPORT TO A C^2 ELEMENT
NUCLEAR/CHEMICAL WEAPONS DELIVERY ENTITY	AGGREGATE ENTITY RESPONSIBLE FOR IMPLEMENTATION OF NUCLEAR OR CHEMICAL WEAPONS EMPLOYMENT DECISIONS BY C^2I ELEMENTS
AIR BASE CRUISER	AGGREGATE ENTITY RESPONSIBLE FOR IMPLEMENTING AIR OPERATIONS
AIR MISSION PACKAGE	TEMPORARY ENTITY COMPOSED AND LAUNCHED BY AIR BASE CLUSTERS TO IMPLEMENT A SPECIFIC AIR MISSION REQUEST (CAS, INTERDICTION)
NUCLEAR/CHEMICAL WEAPONS EFFECTS ENTITY	TEMPORARY "ENTITY" CREATED BY NUCLEAR/CHEMICAL WEAPONS DELIVERY ENTITIES TO REPRESENT EFFECTS IN A PARTICULAR EMPLOYMENT OF SUCH WEAPONS

80W/00102

Figure II-1. Entity Types Included in INWARS

Figure II-2 illustrates some of the principal information elements involved in representing entities within INWARS. As the figure suggests, these fall naturally into familiar Order of Battle type information categories; this is a consequence of the entity-based architecture and promotes a more direct, operationally-oriented representation.

COMPOSITION
- TYPE
- ECHELON
- NATIONALITY
- SUPERIOR & SUBORDINATES

DISPOSITION
- POSITION
- ORIENTATION
- SPEED
- READINESS (NUCLEAR & CHEMICAL)

STATUS
- ASSETS
- STRENGTH
- SUPPRESSION

OPERATIONS
- MISSION/OBJECTIVE
- CONTROL MEASURES
- PERCEPTIONS

Figure II-2. Typical Information Elements Involved in Entity Representation

Certain information elements are applicable to entities of all types: all entities have a location, possess assets, and so forth. Within the simulation, these common information elements are organized in the form of a "unit scoreboard" which serves as the basic entity representation. In addition to these basic information elements, certain types of entities require additional information to perform their functions. For example, to perform C^2I functions, EAD C^2I elements require an overall "Understanding of the Situation" (UOS); likewise, to perform supply and replacement functions, Combat Service Support Complexes require "issue guidance". Within the simulation, such additional information is typically "attached" to an entity's scoreboard.

2. Activities of INWARS Entities

Just as the evolution of a real conflict is driven by the activities of its participants, so too the evolution of an INWARS simulation of a conflict is driven by the activities of its entities. Again, this is a consequence of the entity-based architecture and leads to a more direct, operationally-oriented representation. Certain basic activities--moving, perceiving, inflicting attrition, etc.--can be undertaken by all entities. Other more specialized activities--issuing resources, composing a mission packages, etc.--are performed only by certain types of entities (Combat Service Support Complexes, Air Base Clusters). Finally, some activities are performed differently by different types of entities; for example, EAD C^2I elements develop operations in much more depth than do Division C^2I elements.

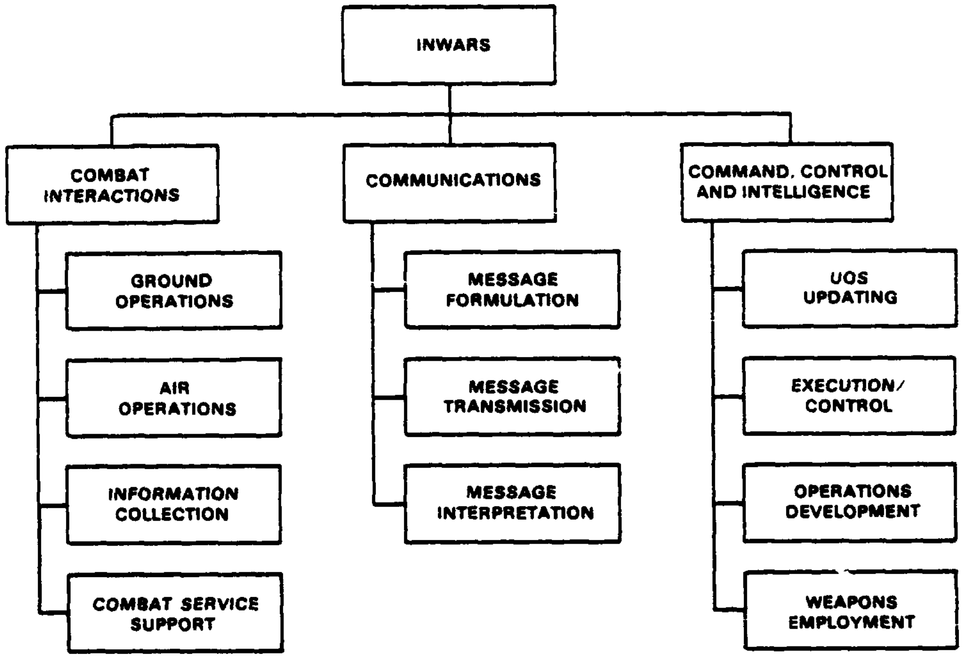
The determination of what activities to undertake at any given point in the simulated conflict is made dynamically by the entities themselves. Activity determination processes of INWARS entities range from explicit, flexible decisionmaking by EAD C^2I elements to more automatic, mechanical reactions by brigades and regiments. In general, though, all entities determine their activities based on: (1) assigned objectives and constraints, (2) perceptions of the enemy forces they face, (3) perceptions of their own capabilities, and (4) their own plans, expectations, and ongoing operations. In effect, these four components establish the situational context within which INWARS entities undertake activities. Differences among entities concern the form in which these context components are represented, and the extent to which they are considered in determining activities to be undertaken.

3. Interactions Among INWARS Entities

Although INWARS entities determine what activities to undertake, they cannot determine the outcomes of these activities. Within INWARS, outcomes--changes in the states of entities--generally depend on the joint activities of several interacting entities. The combat process illustrates this. For example, if opposing brigades were given the same objective they would independently take action to move to it. As they both approach the objective, they would eventually perceive each other; typically, they would then undertake activities to fire on each other, thus becoming engaged. As a consequence, each brigade would sustain some attrition and suppression. Eventually, one (or both) of the brigades could become so impaired as to "back off" from the objective, at least temporarily. Thus, although both undertook activities to achieve the objective, at least one did not achieve the desired outcome.

Under this approach, it is necessary to determine when entities can interact, i.e., when the activities of a given entity can affect--or be affected by-the activities of another entity. Within INWARS, this determination is made by the model and varies with the types of activities involved. For physical activities, spatial proximity is the principal criterion. For example, an entity cannot inflict attrition on other entities unless they occupy the same hex (direct <u>or</u> indirect fire) or adjacent hexes (indirect fire only). Likewise, an entity cannot collect information about other entities unless they are within its (user-specified) search pattern. For mental activities, no direct interaction is possible--a Corps C^2I element cannot "will" a subordinate Division to carry out some operation. Rather, mental interaction is accomplished through communications--explicit transfer of information--among entities. The Corps C^2I element must formulate and transmit an appropriate operation directive to its subordinate. The possibility of communication among force elements is itself constrained to user-specified channels including chain-of-command and support relationships.

The processes by which the activities of INWARS entities interact and are resolved are presented in Figure II-3. They will be surveyed in following three sections corresponding to the three main groups suggested in the figure.



80W/00102

Figure II-3. INWARS Processes

B. COMBAT INTERACTIONS PROCESSES

It is in the physical processes that combat interactions are resolved, changing the states and capabilities of the simulated force elements and thus driving the evolution of the simulated conflict. Reflecting this structure, physical processes are executed periodically for all entities. Within the event-driven architecture, there is a cyclic event representing the physical evolution of the conflict. Upon its occurrence, the effects of entity actions and interactions over the preceding period are resolved: entities move, perceive, inflict and sustain attrition, launch air mission packages, issue and receive supplies, and so on.

1. Conflict Environment

Physical interactions among INWARS entities take place within a conflict environment represented by a hexagonal coordinate system. As noted earlier, this structure of nested hexes provides the means of representing the spatial position of entities within the model (down to the nearest 9.45 kilometer hex in the system). Hexes may also be given attributes corresponding to features of the actual geography they represent. Thus, INWARS hexes have attributes for terrain type, nationality, population density, and nuclear/chemical contamination. In addition, hex boundaries can be identified as rivers or barriers to effect movement between the bounded hexes.

2. Ground Operations

Ground operations are carried out by brigades and regiments operating under the direction of their parent division C^2I elements. Upon receiving an operations directive from their parent corps/army C^2I element, the division C^2I elements conduct a rudimentary operations development process. In this, objectives and control measures are assigned to the subordinate brigades and regiments consistent with the overall division objective. The brigades and regiments then undertake activities to achieve the assigned objectives. This involves moving toward the objective, entering into engagements with opposing forces, and perceiving and reacting to the situations they face.

At this lowest level within the model, the representation of such activities is largely implicit. As mentioned earlier, the physical states of all entities are updated periodically. During updating, each entity has an opportunity to:

(1) Perceive its Situation: search nearby hexes (in accordance with user-specified search capabilities) for enemy entities.

(2) Inflict Attrition: allocate available firepower among perceived enemy entities and inflict corresponding attrition and suppression on those entities.

(3) Consider its Operation: examine the perceived situation, consider changing the ongoing operation (temporarily or permanently), and consider taking specific actions (such as requesting air support or sending reports to the parent C^2I element, etc.).

(4) Consider Movement: based on the perceived situation, move an appropriate distance within the hex and, in some cases, move to a new hex based on objectives, perceived enemy unit locations, terrain features, and so forth.

(5) Reconstitute: implicitly reorganize existing forces to increase effectiveness with which available resources can be employed.

3. Air Combat Support Operations

Air support operations in INWARS are carried out by Air Mission Packages operating under the control of Air Base Clusters. Each ATAF/TAA C^2I element controls a single Air Base Cluster representing all individual air bases within that ATAF/TAA. Air Base Clusters possess various user-specified types of aircraft assets. Although they are capable of limited "operations" (e.g., air defense), Air Base Clusters' main activity is composing available aircraft assets into discrete groups to carry out specific missions. Such groupings--Air Mission Packages-are composed and launched in response to requests from supported force elements. Of course, lack of aircraft precludes Air Mission Package composition; similarly, the finite launch capability "possessed" by an Air Base Cluster limits the number of Air Mission Packages which can be launched in a given combat cycle.

After composition and launch, Air Mission Packages begin to move towards their assigned targets (objectives). This movement is conducted in higher level hexes (66 km in diameter) which provide for ongoing "air battles" consisting of engagements among separate Air Mission Packages (and ground-based air defense capabilities). Upon arriving at their assigned objectives, Air Mission Packages enter the ongoing ground battle and inflict effects on opposing force elements. After their missions have been carried out, Air Mission Packages return to their parent Air Base Cluster (passing through Air Battles once again) where they are decomposed into constituent aircraft assets. The Air Base Cluster may then utilize these assets to compose other Air Mission Packages.

Both Air Base Clusters and Air Mission Packages are susceptable to attrition of assets and degradation of performance as a result of attack by opposing forces. In particular, the attack of an Air Base Cluster may degrade its launch capabilities, thus limiting its ability to service air support requests.

4. Fire Support Operations

Fire support operations are modeled in two forms in INWARS. First, all entities may possess fire support assets (e.g., artillery tubes, rocket launchers, or air defense systems). These assets are utilized as indirect fire weapons which are allocated and targeted implicitly as a part of the normal ground combat cycle. Second, and more explicit, INWARS may contain specific Weapons Delivery Agencies. At present, these are organized to represent aggregate nuclear and chemical Weapons Delivery Agencies available to each Theater C^2I element.

In terms of modeling structure, the activities of Weapons Delivery Agencies are much like Air Base Clusters. Although capable of limited operations, Weapons Delivery Agencies are principally concerned with composing available weapons assets into discrete groups to be delivered against specific targets. Such groupings are composed and delivered in response to specific target engagement requests received from supported force elements. Unlike Air Mission Packages, however, these temporary weapons groups are simply created, moved to the target hex, and allowed to inflict their effects.

Weapons Delivery Agencies can be perceived and acquired by opposing force elements. If targeted, their assets (i.e., nuclear or chemical weapons) may be destroyed and their overall capabilities may be degraded via suppression.

5. Combat Service Support Operations

Combat service support operations including supply, replacement, and repair are carried out by INWARS Combat Service Support (CS^2) complexes. A single CS^2 complex is associated with each INWARS C^2I element (EAD and Division levels) to represent the aggregate service support organizations, systems and capabilities controlled by that C^2I element.

Within the simulation, CS^2 complexes handle all operational resources. Undamaged resources--supplies, major systems, personnel, etc.--are received from higher CS^2 complexes, stocked as "undamaged resources", and gradually issued to lower CS^2 complexes or consumers (e.g., brigades and regiments (in accordance with "issue guidance" received from the controlling C^2I element). Simultaneously, resources damaged in combat are received, stocked as "damaged resources" and gradually transfered to "undamaged resource" stocks by (implicit) repair and maintenance processes. The two basic CS^2 activities--issuing and repairing resources--are separately constrained by corresponding implicit capability levels: issue capability constrains the amount of resources which may be issued in a given period of time by a CS^2 complex; repair capability constrains the amount of resources which can be transformed from "damaged" to "undamaged" status in a given period of time.

CS^2 complexes can be perceived, acquired, and attacked by opposing air and/or ground forces. The effects of such attacks may include destruction of stocked resources, degradation of issue capability, and/or degradation of repair capability.

6. Information Collection Operations

As a part of their basic perception processes, INWARS force elements at division level and above may send status, unit intelligence, or situation feature reports to their parent C^2I elements. This involves translating scoreboard information into a form acceptable to EAD C^2I elements. However, in the case of unit intelligence reports, information about enemy force elements may be degraded to provide imperfect information.

Since INWARS is a deterministic simulation, no stochastic errors in information collection are represented. Rather, information imperfection is represented in terms of incompleteness and imprecision in the information collected. The extent of degradation is controlled by user-inputs; differing modes and levels of degradation may be specified for different 20 kilometer "range bands".

C. COMMUNICATIONS INTERFACE PROCESSES

Throughout the foregoing discussion of INWARS Combat Interactions processes, it was noted that the different types of operations conducted by different entities depend heavily on guidance--operation orders, air requests, weapons requests, issue guidance, etc.--received from other EAD C^2I elements. Similarily it was noted that as the various operations were conducted, reports on own status, enemy forces or situation features would be prepared and sent to other entities. Communications processes are thus a very important form of interaction among INWARS entities. In fact, all interfaces between EAD C^2I processes and physical implementation processes are made by means of communications.

Within INWARS, the communications process is represented in terms of the formulation, transmission, and interpretation of explicit messages (reports, requests, and directives). Each type of entity has a system of procedures by which it decides when and how to construct particular messages. It may send these messages to its parent, one of its subordinates, or to user-specified support elements. The actual transmission process is represented by scheduling the message to be received by the intended recipient following a transmission time delay. A highly aggregated time delay is computed based on the side and echelon of the sending entity; additional delays may result if the sender is a recent victim of a nuclear attack. Upon receiving the message, the recipient interprets it. This may involve changing some aspects of the entity's operational status or perception of the situation.

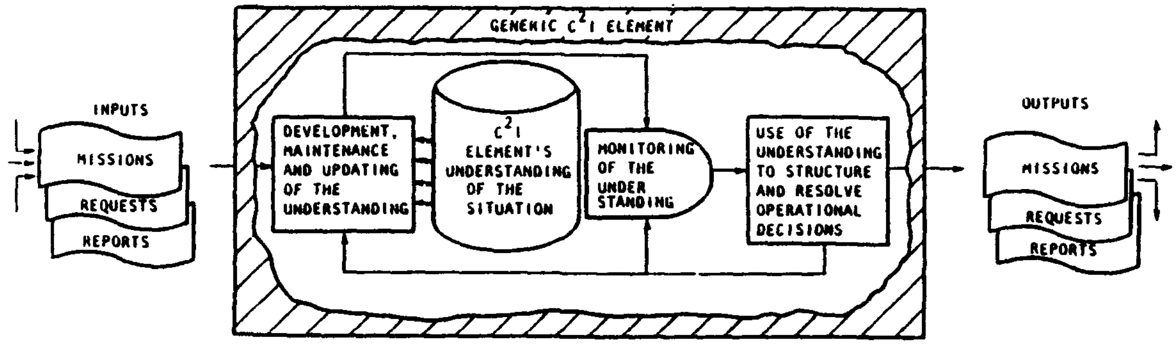
D. COMMAND, CONTROL, AND INTELLIGENCE (C^2I) PROCESSES

Given the INWARS focus on upper-echelon command, control and intelligence, C^2I processes are represented most fully at echelons above division (EAD). These processes are vested in INWARS entities representing Theater, Army Group/Front, and Corps/Army Headquarters--"EAD C^2I elements". Like all entities, EAD C^2I elements possess assets, are located in the hex coordinate system, may undertake various activities such as moving, shooting, etc., and may be acquired and attacked by opposing force elements. In addition, EAD C^2I elements are responsible for the overall operations of major force components: corps and armies, army groups and fronts, and theaters. This responsibility includes developing and maintaining an "understanding" of the overall situation faced, developing operations to achieve assigned objectives, executing and controlling those operations, adapting ongoing operations to the evolving situation, and employing conventional, nuclear, and/or chemical weapons in concert with ongoing operations.

1. C^2I Modeling Approach

The approach to modeling EAD C^2I processes in INWARS regards EAD C^2I elements as nodes in an information processing network. The links --information channels--in this network reflect chain-of-command relationships as well as operational support arrangements. The information which may flow in these links consist of various types of messages including directives, requests and reports. In INWARS, modeling C^2I processes takes the form of modeling the substantive information processing activity which occurs within the nodes--C^2I elements--in this network.

C^2I information processing must include a variety of decision processes. However, it must be based on an "architecture" within which the "proper" decision process(es) can be invoked at the "proper" time. Within INWARS, this C^2I process architecture is based on the fundamental observation that in reality C^2I elements operate on the basis of their overall Understanding of the Situation (UOS). Information received via messages from the network provides each C^2I element a basis for "updating" its UOS. As this UOS changes over time, operationally significant problems or opportunities may become apparent. Such problems and opportunities then stimulate the C^2I element to formulate and resolve a specific type of operational decision problem. In very abbreviated terms, this is the approach to modeling EAD C^2I processes in INWARS--it is presented graphically in Figure II-4.



00858A 79W

Figure II-4. Structure of Generic C^2I Element

Founding the C^2I process modeling on the UOS concept leads to a "knowledge-based" representation having two distinctive features. First, the principal focus is the structure and contents of the UOS, i.e., the nature of the knowledge used by C^2I elements to perform their functions. Second, this focus orients the C^2I process modeling in terms of: (1) processes by which the UOS is updated in response to new information, (2) processes by which the evolving UOS is monitored for operationally significant problems and opportunities, and (3) processes to deal with such problems and opportunities.

2. UOS Structure and Contents

Within INWARS, an EAD C^2I element's UOS is represented by a dynamic structure of information. Although each EAD C^2I element possesses its own individual UOS, all are instances of a general INWARS UOS structure having the form presented in Figure II-5.

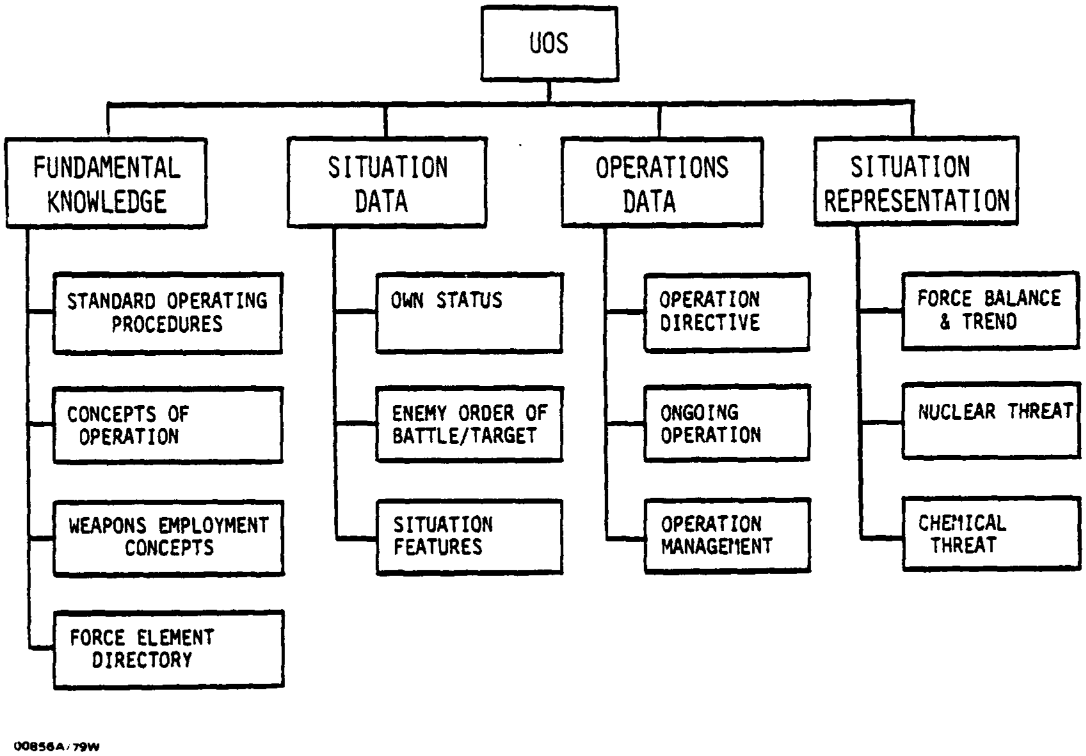


Figure II-5. INWARS UOS Structure

Fundamental Knowledge includes representations of standard operating procedures, concepts of operation, weapons employment concepts, and other general doctrines, policies, and guidance for EAD C^2I elements. These information elements are specified by the user as inputs and are not changed by the simulation during a run. Consequently, Fundamental Knowledge information is one of the user's principal means of influencing the EAD C^2I elements' behavior (and, hence, the course of the simulated conflict).

Situation Data information represents a C^2I element's "data base" of detailed information about friendly forces, enemy forces, and features of the situation (e.g., nuclear/chemical threat indicators). Each EAD C^2I element develops and updates this information within the simulation. As new information about the situation is received (via perceptions or reports), C^2I elements update their Situation Data appropriately. Of course, since the new information may be incomplete or aged, so too may be a C^2I element's Situation Data.

Operations Data represents a C^2I element's understanding of its assigned objectives and constraints, its plans for accomplishing these objectives, its expectations regarding implementation of these plans, and various other information involved in managing large-scale operations, readiness, and weapons employment (especially nuclear and chemical weapons). EAD C^2I elements develop and maintain this information over the course of the simulation run.

Situation Representation information provides a synthesis and aggregation of detailed Situation Data from the perspective of Operations Data. For example, the principal Situation Representation feature--force balance--synthesizes friendly and operationally significant enemy force elements and strengths over time. C^2I elements develop Situation Representation information over the course of the simulation based on their Situation Data and Operations Data.

3. UOS Updating

As EAD C^2I elements obtain new information about the situation, they must update their UOS's accordingly. Fundamentally, it is Situation Data components which must be updated: (1) if a subordinate status report is received, UOS Own Status information must be updated; (2) if a unit intelligence report is received, Enemy Order of Battle/Target information must be updated; and (3) if a situation feature report is received, Situation Features information must be updated. These basic UOS updating processes are directly triggered by the receipt of the corresponding reports; as such, they provide a natural opportunity to check for "operationally significant" changes in the situation. Thus, beyond simply replacing old information with new, comparisons between, e.g., new strengths and old strengths are made; if the difference exceeds user-specified levels of significance, user-specified activities may be undertaken. This may involve derivative updating of other UOS information elements (e.g., force balance or threat indices) to maintain consistency. It may also involve analyzing the possibility of certain operational problems (e.g., poor progress) or opportunities (e.g., target engagement) and responding accordingly. Thus, a C^2I element's UOS is an "active" data base which may trigger decisionmaking activities as it is updated.

4. Operations Development

One of the principal responsibilities of INWARS C^2I elements is developing operations to achieve objectives assigned by higher echelon C^2I elements within the simulation. This means providing subordinate commands with operations directives containing specific assignments of objectives, missions, control measures (sector widths, timing), and resources (air support, weapons, supplies and replacements). Of course, subordinates' objectives, control measures, and resources must be assigned so that a coherent and coordinated overall operation results. Moreover, the overall operation must generally be decomposed into shorter phases appropriate to subordinate-level operations. Finally, to guide the execution of the overall operation, the developing C^2I element must make estimates of expected progress against which to appraise the actual conduct of the operation.

In essence, the operations development process utilized by INWARS EAD C^2I elements involves applying generalized "concepts of operation" in specific situations. The user supplies each EAD C^2I element with concepts of operation which specify the general "form" of operations (e.g., envelopments, penetrations, active defenses, delays, and so forth). These concepts of operation are then dynamically "developed" by the EAD C^2I elements to fit the specific situations they face. As user-inputs, the concepts are "abstract" in that no specific force elements, deployments, objectives, or timing are identified. Rather, the user formulates concept of operation in terms of general force roles (main attacker, reserve, etc.), general operational relationships among roles, and corresponding configurations of objectives in a generalized "planning grid". In effect, a concept of operation may be regarded as a kind of "procedure" for developing and conducting a general form of operation in a specific situation.

To interpret these "operation procedures", INWARS C^2I elements have processes to assign subordinate forces to specific roles, establish specific operational linkages among roles, detail specific objectives consistent with a general configuration, and so forth. Figure II-6 illustrates the particular processes involved in INWARS operations development activities.

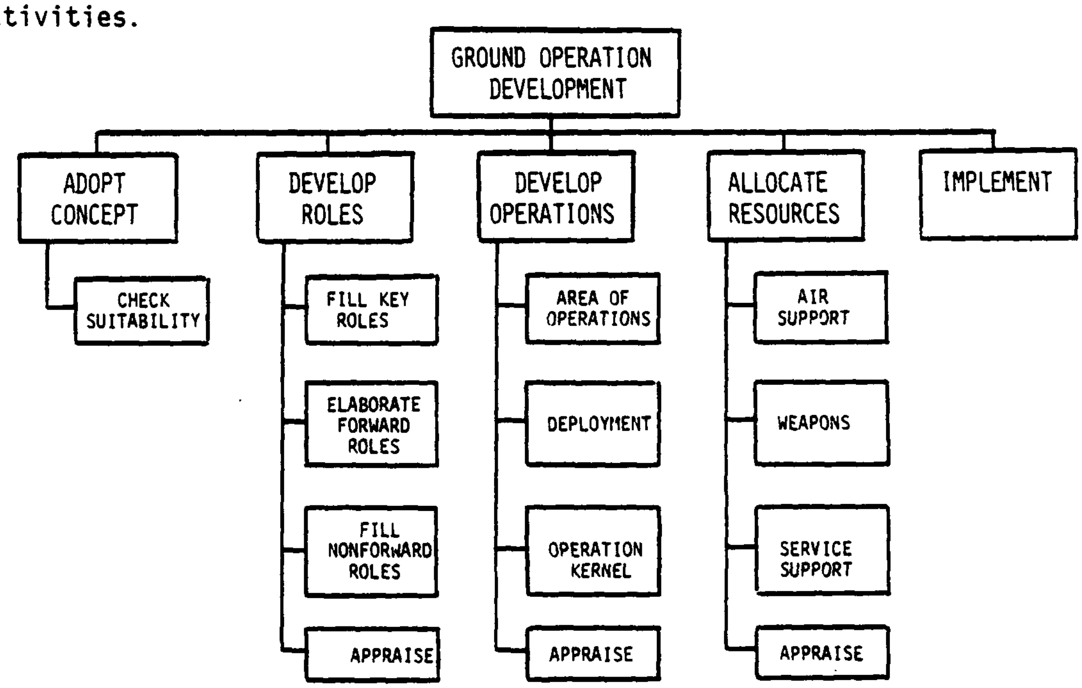


Figure II-6. INWARS Operation Development Processes

Various appraisals are distributed throughout the overall process. These concern such operational features as distribution of forces, expected completion time for the operation, and resource feasibility. Consequently, EAD C^2I elements can develop and compare different user-supplied concepts of operation with respect to these common measures in the specific situation they face. This capability is the basis for the overall INWARS operations development process: EAD C^2I elements utilize the user-specified concepts of operation to develop an appropriate set of operations, make comparisons within the resulting set of specified operations, and implement the "best" of the specified operations (with respect to the various appraisals).

As just surveyed, the operations development processes reflect a kind of automated policy application. EAD C^2I elements are constrained to utilize the policies--concepts of operation--supplied by the user; these provide the broad form of the operational alternatives available to the C^2I elements. However, since concepts of operation are "abstract", (in the sense discussed above), EAD C^2I elements must make many decisions regarding how a concept is to be specifically implemented (e.g., which subordinate forces are to perform which of the concept's roles). Thus, although the form of the alternatives is user-specified, EAD C^2I elements must still define the specific alternatives to be considered in specific situations.

5. Weapons Employment

Another principal responsibility of INWARS EAD C^2I elements is the employment of weapons, especially nuclear and chemical weapons. In developing a particular employment of weapons, an EAD C^2I element may explicitly assign weapons against certain targets it has acquired; it may also apportion weapons to subordinate EAD C^2I elements for employment at their own discretion.

Like operations development, weapons employment is a concept-guided process in INWARS. The user supplies each EAD C^2I element with a set of "weapons employment concepts" for each type of weapons to be employed (conventional interdiction, nuclear weapons, and chemical weapons). These weapons employment concepts provide generalized guidance concerning the applicability of the concept in different types of situations, the general priority and desirability of various levels of effect against different types of targets, and the execution of an employment developed under that concept. A weapons employment concept may be regarded as a "procedure" for employing weapons in a certain manner (e.g., counter-ground forces, counter-support, or counter-air).

INWARS EAD C^2I elements have the capability to interpret these general weapons employment "procedures" to develop specific weapons employment plans. Figure II-7 presents the structure of this process. Employment concepts are utilized principally in the "Develop Plan" activity where they guide the development of a discretionary fire plan (weapons-target assignments) and the apportionment of weapons among subordinates. The surrounding activities--preparing to develop a plan and acting on a developed plan--manage the variety of situations and conditions under which weapons employment may be conducted (i.e., on own initiative or in response to a request from a subordinate, with or without authorized weapons, in target-rich or target-poor environments, and so forth).

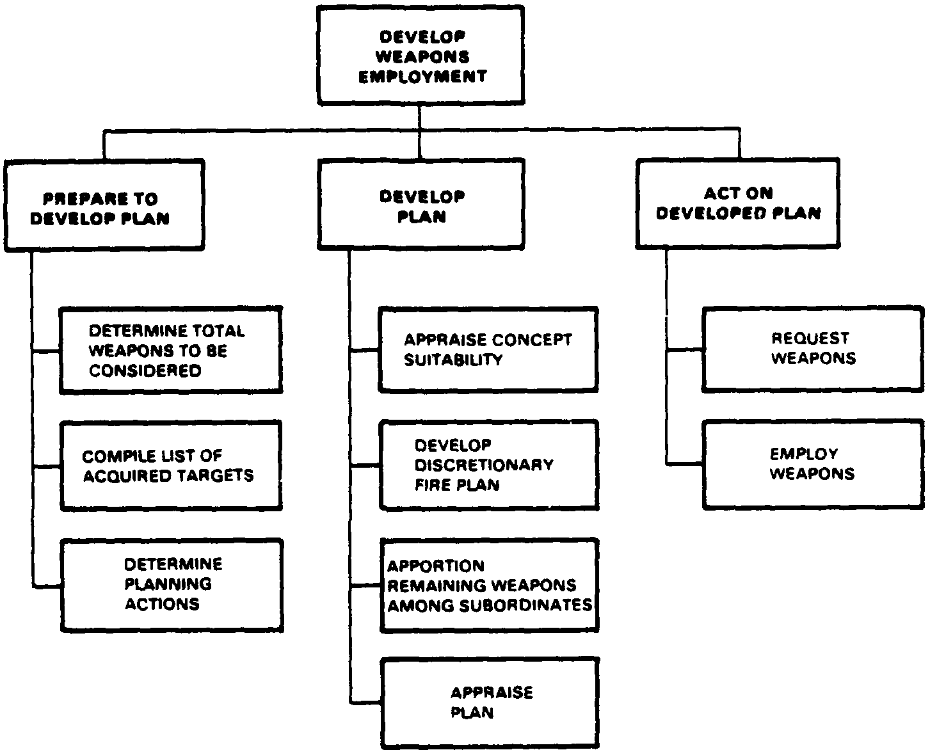


Figure II-7. INWARS Weapons Employment Activities

6. <u>Execution and Control</u>

Once an EAD C^2I element has developed and implemented an operation, it must control the execution of that operation, keeping it adapted to the evolving situation. INWARS EAD C^2I element execution and control procedures are structured around the notion of "contingencies", i.e., operationally significant problems or opportunities. Within the simulation, a contingency is essentially defined by: (1) a recognition procedure, to identify the existence (or potential existence) of the contingency, and (2) a response procedure, to determine how the recognized problem (or opportunity) whould be rectified (or exploited). These procedures may be more or less complex, drawing on more or less extensive portions of the UOS. Recognition procedures may be invoked on the basis of specific perceived changes in the situation identified during UOS updating, or during a periodic "internal review" by a C^2I element.

The particular contingencies treated in INWARS are presented in Figure II-8. These range from very specific contingencies ("new directive", "new request") to very general contingencies ("operational progress").

- INTERNAL REVIEW
- NEW DIRECTIVE
  - OPERATIONS
  - READINESS
  - WEAPONS
- NEW REQUEST
  - INFORMATION
  - CAS
  - TARGET ENGAGEMENT
  - WEAPONS
- THREAT LEVEL
  - NUCLEAR THREAT
  - CHEMICAL THREAT
- OPERATIONAL PROGRESS
  - KERNEL OPERATION
  - OVERALL FORWARD OPERATION
- TARGET ENGAGEMENT

Figure II-8. INWARS Contingencies

General contingencies insure that operational problems and opportunties will eventually be recognized and acted on. At present, the "operational progress" contingency serves this function for INWARS EAD C^2I elements--almost by definition, operational problems and opportunities must ultimately impact on operational progress. By contrast, specific contingencies promote responsive recognition and response to particular problems of interest. The "threat level" contingencies exemplify this in that they enable EAD C^2I elements to monitor changes in the perceived threat of nuclear or chemical attack and respond by implementing (or de-implementing) "adaptive measures" such as changing the nuclear or chemical readiness of subordinate forces. These specific actions are determined on the basis of user-input threat response doctrines.

CHAPTER III
INWARS SIMULATION SOFTWARE

A. INTRODUCTION

This chapter surveys the software which realizes the INWARS representation of theater level conflict as a simulation. Section B presents the overall structure of this software. The development of this software is then discussed in Section C. Detailed discussion of the INWARS software will be found in the Software Description component of the documentation.

B. INWARS SOFTWARE STRUCTURE

The top-level structure of INWARS software involves the three procedural components presented in Figure III-1; Input, Simulation, and Output. The arrows suggest the general flow of the execution although the processes are not executed in a completely sequential manner--output, for example, is extracted over the course of the simulation run. These three components will now be discussed.

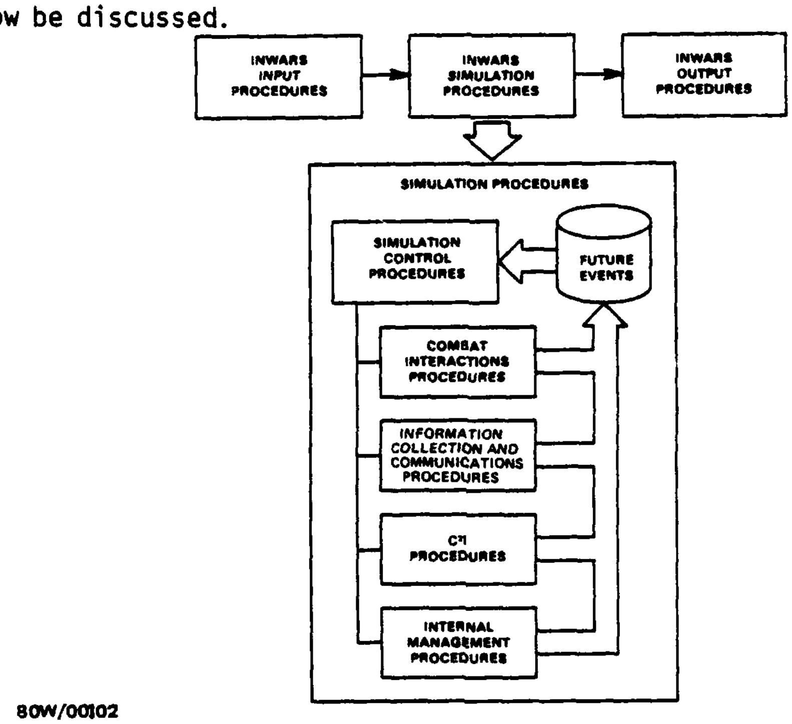


Figure III-1. INWARS Software Structure

1. INWARS Input Procedures

Input procedures provide the means by which user-specified environmental features, force configurations, performance parameters, doctrines, policies and procedures, and theater campaign directives are interpreted and transformed into acceptable forms for the simulation. For present purposes, initialization procedures may be regarded as a part of the INWARS input procedure component--initialization procedures are, in fact, interlaced with the procedures which read input and build appropriate data structures. In summary, then, input procedures read user inputs and construct the internal representation of the "initial state" of the conflict to be simulated.

Given the INWARS focus on upper-echelon C^2I processes, and the complexity of the associated doctrine, policy, and procedure inputs, emphasis has been placed on facilitating these inputs. This has taken the form of a User-Oriented Input Language (UOIL) which permits the user to construct the various SOPs, concepts of operation, weapons employment concepts, and other C^2I structures in a natural, self-documenting input text.

2. INWARS Simulation Procedures

It is in the INWARS simulation procedures that the modeling approach surveyed in Chapter II, above, is realized. The structure of the simulation procedures is founded on INWARS event-driven architecture. As portrayed in Figure III-1, there are four basic systems of procedures: Combat Interactions procedures, Communications procedures, C^2I procedures, and Internal Management procedures. The first three systems of procedures realize the INWARS representation of Combat Interactions processes, Communications Interface processes, and EAD C^2I processes as described in Sections B, C, and D (respectively) of Chapter II, above. The "Internal Management" procedures provide for certain initialization functions and outputs.

As suggested in Figure III-1, these systems of procedures execute independently under the overall management of a system of Simulation Control procedures. To determine which system of procedures should be executed, Simulation Control utilizes an internally generated set of "future events". Each of these events involves a particular system of procedures and is scheduled to occur at some specific time. The events thus consitute a kind of "agenda" of what is to happen in the conflict. It is emphasized that this agenda is dynamically generated by the simulation itself -- it is not supplied as "script" by the user. To use this agenda, Simulation Control scans the events and selects a unique "next event" (i.e., that scheduled event with the smallest--"soonest"--occurrence time). It then causes this event to "occur" by updating the simulated conflict time to the event occurrence time, and passing the event to the appropriate system of procedures for processing. As a part of processing the event, other events may be generated--e.g., a C^2I procedure may lead to a communications event ("message receipt") at some later time. In this way, the simulation continues to evolve through time until a specified "end-of-game" time is reached. To insure that the supply of future events is never exhausted, the Combat Interactions procedures are executed cyclically by simply "rescheduling" themselves as a part of each occurrence.

3. INWARS Output Procedures

As was noted earlier, INWARS outputs have the form of "state snapshots" of the true state of the conflict as well as the state "perceived" by the upper echelon C^2I elements. Snapshots of both types are taken periodically. This is accomplished by a special "internal management" type event whose occurrence invokes the appropriate write-out procedures. "Perceived state" snapshots are also taken for individual EAD C^2I elements whenever they make key decisions (such as developing new operations or employing weapons). This is accomplished by invoking appropriate output procedures upon completion of the corresponding decision process. These procedures simply output selected information elements in appropriate data structure formats.

C. INWARS SOFTWARE DEVELOPMENT

The development of INWARS software has emphasized top-down, modular design, and the use of existing software approaches where possible. For example, much of the combat interactions software has been developed by extending, adapting, and generalizing approaches used in other models, notably the earlier T-COR/Corps Level Electronic Warfare (CLEW) model. Even where there were no existing software approaches (as in the C^2I area), INWARS has drawn on various software development tools. Principal among these are the use of the Program Design Language (PDL), and the Modular Information Data Access System (MIDAS). PDL and MIDAS are especially important in that they alleviate many of the difficulties of structuring algorithms and data in FORTRAN.

1. Program Design Language (PDL)

Software design has traditionally been expressed through a combination of flow charts, decision tables, and narrative description. These techniques have a serious disadvantage in that they are usually physically separated from the final software whose design they describe. Recently, program design languages (PDL) have been developed independently by several investigators. Although the dialects of PDL vary, the principles are the same. PDL provides: (1) A vehicle to translate functional specifications into program design, (2) A replacement for logic flowcharts, and (3) A means of communications between designers and implementors.

PDL utilizes the concepts of structured programming to achieve a structured modular design. A significant advantage of PDL is that it permits software design to be expressed in a manner which is independent of implementation programming language, implementation details, and the computer system for which the program is being developed. Only the logical aspects of the design are expressed in PDL, not the implementation or physical aspects. On the other hand, PDL has a closer relationship to programming languages than traditional methods of expressing design, thereby permiting a more direct mapping of the design into code.

PDL is English-like in expression and follows certain semantic and syntactic conventions. The concepts of structured programming are applied in the form of basic control structures of logic flow and indentation. Top-down programming is implemented by specifying in PDL the levels of detail of the modules, and enriching the detail in an evolutionary process.

The use of PDL eliminates the need for all flow charts and provides a self-documenting capability for the program itself. Realization of an implementation consists simply of adding the necessary coding to the logical PDL design statements. Thus, the design language and the implementation language co-exist in the final source code.

2. Modular Information Data Access System (MIDAS)

The limitations of data structures supported by FORTRAN are widely recognized. As used in INWARS, MIDAS extends the data structuring capabilities to include: (1) records (composite structures whose elements are accessible by name) and (2) linked structures (composite structures whose components are accessible by pointers or chains of pointers). Thus, MIDAS makes it practical to utilize rich data structures in FORTRAN--this has proven to be especially useful in the treatment of EAD C^2I processes.

MIDAS consists of two parts: the MIDAS language and the MIDAS Translator. The MIDAS language allows one to write programs using only the logical aspects of data structures. The MIDAS Translator reads data structure definitions and programs written in the MIDAS language, realizes the physical implementation of the logical data structures, and generates a FORTRAN program or subprogram as output, ready for compilation.

The advantages of using MIDAS are two-fold. First, the designer is no longer concerned with the details of data structure implementation, and is free to use natural names for elements of the data structures. Second, a program written in MIDAS is easier to convert to another type of computer since the logical definition of the data structures in the code does not change, only the physical implementation. This is accomplished by means of the MIDAS Translator which is controlled by a set of tables defining the logical data structures to be translated and their specific physical implementation.

The MIDAS Translator constructs these tables automatically using information supplied by the user. This definition information is expressed using a Data Structure Definition Language (DSDL) which defines the logical data structures and their physical implementation details. This language can also be used to uniquely specify and document the logical design of data structures.

CHAPTER IV
INWARS DOCUMENTATION

To conclude this synopsis of INWARS, this chapter provides a "roadmap" to the INWARS documentation. It is intended to assist the reader in determining where to look for more information about particular aspects of the simulation. As noted in Chapter I, the overall INWARS documentation has four major components: (1) this synopsis, (2) a Modeling Description component, (3) a Software Description component, and (4) a User's Manual component. These components and their subordinate volumes are presented in Figure IV-1.

The Modeling Description component discusses the form in which integrated theater level warfare is represented in INWARS. In essence, it expands on the synopsis given in Chapter II, herein, by discussing the form in which entities are represented in the simulation, the activities they may undertake, and the processes by which these activities may impact on other entities. The discussions are organized by broad functional areas: Ground Combat (Volume II), Air Support (Volume III), other Combat Support (Volume IV), and EAD C^2I activities (Volume V). An introductory volume reviews the general modeling architecture of INWARS and discusses the representation of the environment.

The Software Description component presents the software by which the INWARS representation of integrated theater-level warfare is realized. This component expands on the synopsis given in Chapter III above. An introductory volume (Volume I) presents the overall software framework within which INWARS has been implemented. The remaining volumes discuss the data structures and procedures which constitute the INWARS software itself. Here, the organization shifts from a functional orientation to emphasize the principal software components: EAD C^2I data structures and procedures (Volumes II and III), Information Collection and Communications data structures and procedures (Volume IV), and Combat Interactions data structures and procedures (Volumes V and VI).

PART I - INWARS SYNOPSIS

PART II - MODELING DESCRIPTION

VOL. I INTRODUCTION
VOL. II GROUND COMBAT OPERATIONS
VOL. III AIR OPERATIONS
VOL. IV COMBAT SUPPORT OPERATIONS
VOL. V ECHELON ABOVE DIVISION COMMAND, CONTROL, AND INTELLIGENCE (EAD C^2I) ACTIVITIES

PART III - SOFTWARE DESCRIPTION

VOL. I SOFTWARE FRAMEWORK
VOL. II EAD C^2I DATA STRUCTURES
VOL. III EAD C^2I PROCEDURES
VOL. IV INFORMATION COLLECTION AND COMMUNICATION DATA STRUCTURES AND PROCEDURES
VOL. V COMBAT INTERACTIONS DATA STRUCTURES
VOL. VI COMBAT INTERACTIONS PROCEDURES

PART IV - USER'S MANUAL

VOL. I INTRODUCTION
VOL. II COMBAT INTERACTIONS INPUTS
VOL. III EAD C^2I INPUTS
VOL. IV INWARS OUTPUTS

Figure IV-1. Overview of INWARS Documentation

The User's Manual component discusses the inputs and outputs of INWARS. Following an overview in Volume I, inputs for the Combat Interactions and C^2I portions of the model are discussed in Volume II and III, respectively. Finally, the outputs produced over a run of the simulation are described in Volume IV.