131
JPL D-~~1488~~

U.S. Army Intelligence Center and School (USAICS)
Software Analysis and Management System USAMS

Cross-Correlation: Statistics, Templating, and Doctrine

AD-A155 624

February 29, 1984

Best Available Copy

DTIC FILE COPY

JUN 1 0 1985

A

National Aeronautics and
Space Administration

JPL

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

This document has been approved
for public release and sale; its
distribution is unlimited.

85 5 23 167

2

U.S. ARMY INTELLIGENCE CENTER AND SCHOOL
Software Analysis and Management System

CROSS-CORRELATION: STATISTICS, TEMPLATING AND DOCTRINE

February 29, 1984

Authors:

Martha Ann Griesel

Martha Ann Griesel, Subgroup Leader
Algorithm Methodology Subgroup

James W. Gillis
Algorithm Methodology Subgroup

John R. Radbill

John R. Radbill, Consultant
Algorithm Methodology Subgroup

SFC Nicky L. Sizemore, Consultant
USAICS

Concur:

Edward J Records

Edward J. Records, Manager
USAMS Task Team

J. P. McCl

J./P. McClure
Ground Data Systems Section

Yukio Nakamura

Yukio Nakamura, Manager
Defense Information Systems Program

DTIC
S ELECTE D
JUN 1 8 1985
A

This document has been approved
for public release and sale; its
distribution is unlimited.

JET PROPULSION LABORATORY
California Institute of Technology
Pasadena, California

ACKNOWLEDGMENT

The following people have contributed to this document:

P. Babby

A. M. Farny

S. Feinstein

I. Kahn

K. McLaughlin

V. Nibler

N. Palmer

P. South

D. Yarbrough

The suggestions of Dr. Carl Verhey on the content of this report are gratefully acknowledged.

Accession For
NTIS GRA&I ☒
DTIC TAB □
Unannounced □
Justification
By
Distribution/
Availability Codes
Dist
A-1

DTIC
COPY
INSPECTED
3

CONTENTS

APPENDIXES

Figures

__Tables__

SECTION 1

INTRODUCTION

1.1 PURPOSE

This report is one of a series which describes the findings of the Algorithm Analysis Subtask working on the US Army Intelligence Center and School (USAICS) Software Analysis and Management System (USAMS) task studying algorithms in intelligence systems. It deals specifically with cross-correlation, a process for analyzing a collection of sitings of enemy units and equipment to identify and link associated "children" and "parent" entities, such as a radar and its artillery battery. In the report individual algorithms from proposed and developed Army Intelligence and Electronic Warfare (I/EW) systems are analyzed to determine their underlying mathematics, their military function, and the assumptions which tie mathematics and function together. The doctrine which governs what the observed military objects look like and how they behave is also discussed, together with the mathematics that associates intelligence observations with those military objects.

1.2 BACKGROUND

Each of the more than 40 intelligence systems under USAICS cognizance employs several types of algorithms to carry out its gathering and processing of intelligence data. The USAMS Algorithm Analysis Subgroup is studying and reporting on selected types of these algorithms; the studies already completed are listed in Appendix D. Two types of algorithms, geographic transformation and self-correlation. were chosen for analysis during last year. The former translates grid-zone locations from one system to another, for example, from latitude-longitude into Universal Transverse Mercator. The latter, using mainly statistical procedures, resolves many individual sitings into militarily recognizable units and equipment for intelligence and target development. Work this year has focused on cross-correlation, which further synthesizes the results of self-correlation.

In these studies "algorithm" means any set of rules for carrying out a single conceptual operation on a set of data. The conceptual models on which these procedures rest can usually be presented simply, cleanly and logically; however, presenting their software implementation is often quite complex. Algorithms are often hierarchical, lower-level algorithms being used to describe higher-level ones, thereby illuminating the underlying structure. The results from one algorithm thus often become data for another. This occurs extensively for the correlation algorithms; identifying the assumptions correctly that link the hierarchical levels is critical. USAICS is interested in algorithms that perform intelligence data processing functions central to its I/EW systems' missions and those that perform crucial support functions common to a number of systems, such as geographic location and transformation. Data management and mathematical function algorithms, although vital to the efficient functioning of the systems, are not being treated in this series of studies.

The algorithms examined in this report come from four processing systems. The Battlefield Exploitation and Target Acquisition system (BETA) is a joint Army-Air Force test bed for correlating data received from a large array of battlefield, air, and national sensors. The Template-Assisted Intelligence

Fusion Program (TEMPRO) is an interactive system that helps the operator/analyst develop, identify, and locate units by using stored templates. TCAC(D) is an Army Quick Reaction capability for COMINT and ELINT only. Marine Air/Ground Intelligence System (MAGIS) is the Marine Corps command-level analysis system which will eventually interact with the Army and Air Force All Source Analysis System (ASAS). In this report "ASAS" will be used only as a generic term referring to the software needed to support an All Source Analysis Center (ASAC). Documentation available for this analysis has ranged from general mathematical descriptions to flow charts and design documents to actual code, with only some of these available for each system.

1.3 USER BENEFITS

These analyses are being conducted to increase USAICS understanding and control of the software under its cognizance. The catalogue of existing algorithms now being assembled can preclude having algorithms unintentionally redeveloped for new systems from first principles. Analysis of individual algorithms may even. in a few cases, identify deficiencies worth correcting on the next system revision. The collection of selected and analyzed algorithms that has evolved from these analyses will form a library of intelligence algorithms with their associated computer subroutines analogous to the Collected Algorithms of the Association for Computing Machinery (ACM). The creation of such a library is in the spirit of Ada, the DOD language for embedded systems, and the Ada environment.

Comparing algorithms performing the same function in different systems is leading to guidelines for developing or selecting algorithms to include in new and revised systems. Prototype guidelines for algorithm analysis are now being set down, and one report dealing solely with a methodology for analyzing I/EW algorithms has already been published (see Appendix D).

1.4 STUDY ORGANIZATION

The report will first consider the military requirements for cross-correlation processing, both Blue force functional requirements and Red force templating requirements. The second section is thus a qualitative discussion of the general role of corps-level intelligence analysis (by Army doctrine, corps is the level where strategic and tactical intelligence information is combined). The section will also discuss Warsaw Pact doctrine and force structure issues that affect the Red force templates used by cross-correlation to identify units. The third section will concentrate on inputs to cross-correlation, mapping the overall data flow through successive intelligence processing systems. This requires considering message interfaces and ways in which previous processing (at the sensor or in self-correlation) affect the inputs cross-correlation sees. The fourth section discusses cross-correlation algorithms and their mathematical and system architecture requirements for the four systems. The last section contains general observations and specific conclusions.

SECTION 2

DOCTRINAL CONSIDERATIONS

Cross-correlation is the first function in the sequence of automatic I/EW information processors which deals chiefly with complex enemy unit structures (as opposed to technical equipment parameters) and whose information is used to initiate tasking for a spectrum of friendly weapons systems (artillery, rocket, and air resources), as well as for general corps and division-level tactical planning. The topics of this section are the role of Blue doctrine in shaping cross-correlation to generate information for supporting both intelligence analysis and assets management, and the role of Red doctrine in defining templates for unit identification.

2.1 BLUE TACTICS AND TARGETING APPLICATIONS

An intelligence processing system can be used either for generating target nominations (target development) or for situation analysis (intelligence development). Both these operations require the same inputs from the intelligence system:

(1) Unit identification (sitings reconciled into militarily recognizable entities, e.g. tanks, radars, motorized rifle regiments, command posts).

(2) Unit location.

(3) Unit behavior.

The three "quality of information" factors corresponding to these are:

(1) Certainty of unit identification.

(2) Accuracy of unit location.

(3) Timeliness.

This section will explore the relative importance of these three factors for intelligence and target development missions. Targeting will be considered first because its requirements are more technical and serve as a basis for later discussion.

2.1.1 Target Development

Targeting requirements differ according to the weapon system being supported. Those systems usually based farther forward and directly supporting the fire battle have relatively limited effects. They make different claims on the intelligence system than those weapon systems deployed farther back which have greater effectiveness per strike. Targeting requirements will thus be discussed separately for each of the three categories of weapon systems possibly supported by division or corps intelligence: artillery, air, and rockets.

2.1.1.1. Artillery. Artillery has two missions, direct and general support. Each mission has its own intelligence requirements, and will be discussed individually.

Direct support artillery, located with a brigade, will fire mainly in response to forward observers and in support of commander combat plans. Any counterbattery fires are directed by organic sensors, i.e., counterbattery and countermortar radars that use ballistic backtracking algorithms. These systems would probably not even query a division or corps-level intelligence processing system, but will provide it with information on enemy unit positions and movement. Commander combat plans rely on general intelligence development.

General support artillery, at division or corps, will fire counterbattery and also in support of commander combat plans (e.g. to concentrate fires). Counterbattery fires would probably still be supported by organic sensors, although targets could be checked by or received from the ASAC, if such tasking could be timely.

Either direct or counterbattery fires require precise target locations rapidly since enemy tanks and self-propelled artillery are often moving. Only purely suppressive or barrage fires need less precise locations. Thus, for artillery, location accuracy and timeliness are paramount.

2.1.1.2. Rockets. These more costly and more powerful systems are usually deployed at corps. They require "absolutely" certain target identification (to avoid wasting a scarce resource), but usually do not require the precise location or immediate response so important to artillery. However, among the battlefield interdiction and deep-strike targets against which a rocket system is effective, rockets rather than air strikes will be employed against those targets that are time sensitive. Thus, rocket systems, with their requirement for the fairly rapid "sure" target identification obtained by fusing information from many sources, are a classical application for ASAC targeting.

2.1.1.3. Air Strikes. As for rocket systems, certain target identification is important for air strikes, although the "man in the loop" decreases its totally paramount importance. Timeliness is not crucial as it was for artillery, because close air support sorties are made up no more often than every two hours, and battlefield interdiction or deep-strike missions are usually planned daily. Occasionally, close air support will be diverted to support particular ground operations in a changing tactical situation; but this is in response to a command decision based on general intelligence and is tactics, not targeting.

Most units targeted by air strikes move slowly, if at all. Units in assembly areas are not moving. Units moving down a road present a problem, and it is not clear how often they would be targeted. However, in either case precise target locations are not required. Hard targets, such as bridges, require precise locations, but their locations are known without the aid of tactical intelligence systems.

2.1.2 Intelligence Development

The emphasis for situation analysis differs from that for targeting. Precise location of units no longer plays a major role; unit identification, including subordination and behavior, does. Current behavior and organizational structure often define and signal enemy intent. Timely deciphering of enemy intent or indication of change in it is crucial. Such information must be as accurate as possible, especially if it is incomplete. Information which is both sparse and inaccurate can lead to devastatingly incorrect interpretations.

2.1.3 Overall Criteria

Targeting requirements for less powerful systems require timely processing of reams of data to obtain precise locations. This requirement is incompatible with all other targeting and situation analysis requirements. The prevailing requirements thus become timely identification of enemy deployment and intent. Even these may be incompatible, for either sure identification or determination of intent may require many reports giving complementary characteristics of the unit being observed. For example, both the identification and mission of a radar are deduced from its location relative to other equipment and its signal characteristics. Any analysis requiring much information to be processed quickly results in a conflict of system priorities.

This problem of conflicting resource allocation can be somewhat mitigated by proper sensor tasking, trying to optimize the value of each additional sensor report on the same entity by having the sensors look in intelligence "holes." This raises still another trade-off, for when sensors look at a specific place, they will most likely not be positioned to pick up the unexpected enemy activity found by more general scans. Quantifying and evaluating these decisions can be supported by modeling the sequence of information received and studying the probabilities associated with such sequences. Such modeling uses many mathematical concepts, such as optimal stopping times and experimental design, and is beyond the scope of this report.

Algorithms can thus be assessed according to these same three criteria:

(1) Certainty and accuracy of unit identification.

(2) Timeliness.

(3) Precision of unit location.

The above order reflects the priority suggested when looking at trade-offs among the criteria.

2.2 RED FORCE TEMPLATING

For this report templating means the art of describing a military entity in terms of observables that uniquely define it. These observables can be:

- Specific types of equipment, perhaps in special configurations.
- The deployment pattern, especially distances.
- The time sequence in which equipment or units appear and duration they are observed in one place.

Preferably, the unit characteristics used should be observed by NATO intelligence as easily as possible. However, descriptions which lead to positive identification are often difficult to construct. This section will present the basic elements governing template construction, namely doctrine and situation (including terrain, weather and vegetation), from the perspective of constructing militarily viable templates that can be used for mathematical analysis of intelligence data. The examples will focus on radars, the units to which they are organic, and these units' position in the force structure, thus keeping a more uniform view of hierarchical template construction. This will become important when analyzing BETA and TEMPRO.

2.2.1 Doctrinal Templating

Doctrine dictates what equipment constitutes a unit of particular type and echelon. For example, a self-propelled artillery battery usually has

6	self-propelled howitzers
3	URAL-375 trucks
2	Armored Combat Reconnaissance Vehicles

as well as radios and radars, and will follow 500 to 1000 m behind the motorized rifle company in whose direct support it is firing. Thus, doctrine also gives the position relative to (thus distance from) both the Forward Edge of Battle Area (FEBA) and other units. Finally, doctrine also defines the usual deployment of a unit, be it a fixed unit (as Figure 2-1, SA-2 battery), or a moving unit advancing, or a unit engaged in combat. This set of properties is usually used to define templates. Those properties also used in automatic processing are:

- Equipment constituting the unit.
- Intra-entity distance (usually a radius).
- Inter-entity distance.

Distance from the FEBA or spatial relation to other units is usually left to decisions by the analyst and is not accommodated in the automatic processing.

As an example, consider the problem of identifying an SA-2 battery. One indication of its presence, is its radar (a Fan Song). Certainly the battery reveals itself by firing, but it is preferable to identify it before it fires. If the radar is unique to SA-2 batteries (Table 2-1, Soviet Air-Defense *Radars, associates radars and units) or can* by some spatial (distance) measure be differentiated from other such radars (Figure 2-2 indicates how far apart the units with their radars might be deployed), it can be used to form the basis for

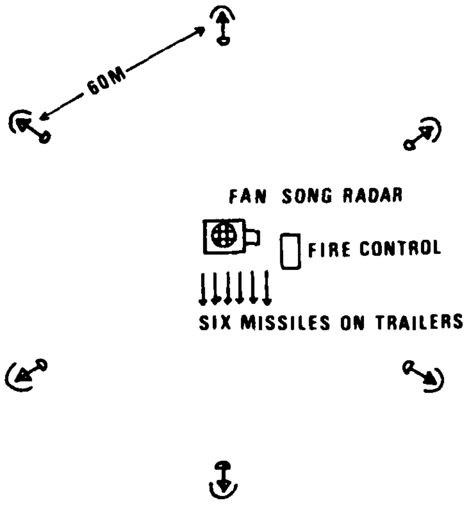
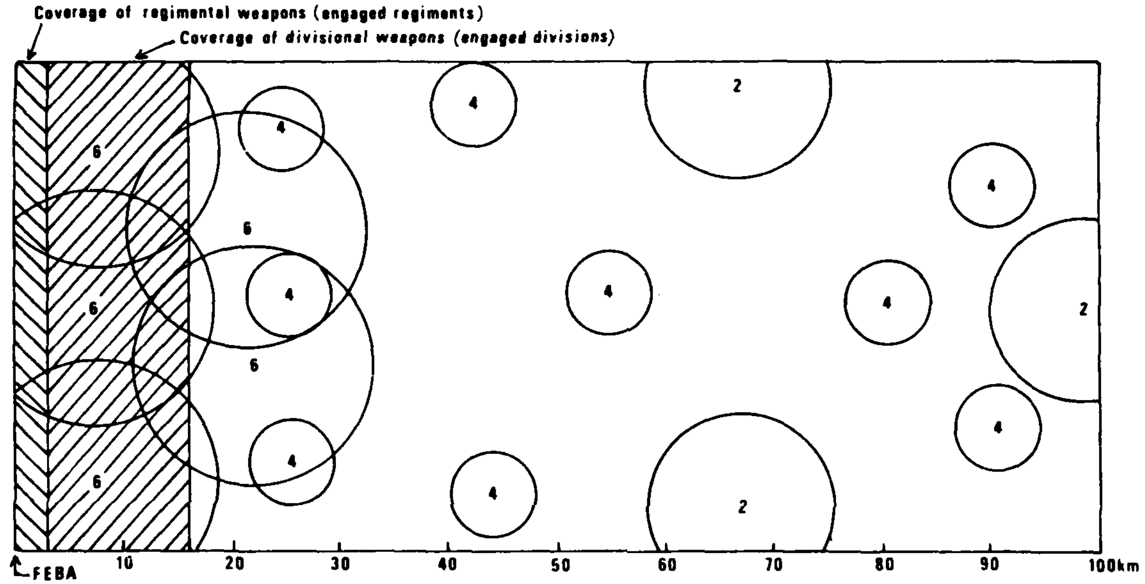


Figure 2-1. SA-2 Battery Deployment

From: D. C. Isby. Weapons and Tactics of the Soviet Army. London: Jane's, 1981, page 248.


Coverage of regimental weapons (engaged regiments)
Coverage of divisional weapons (engaged divisions)
6
6
6
6
6
4
4
4
4
4
4
4
4
4
2
2
2
FEBA
10
20
30
40
50
60
70
80
90
100km

This diagram shows the air-defence cover of a typical Soviet Army sector extending 100km behind the forward edge of the battle area (FEBA) and along 45km of frontage. In addition to the missile and AAA defences shown (each type of SAM is represented by the appropriate numeral), the Soviets position point-defence systems not only on the front lines but throughout the sector. Second-echelon regiments and divisions will be behind the FEBA, and their ZSU-23-4s, S-60s, SA-8s, SA-9s and SA-7s will fill in any gaps under the area-defence missile systems. The radii shown are the US Air Force's "avoidance radii," which they will not penetrate unless required. If maximum range were used, the whole sector would be covered.

Figure 2-2. Air-Defense Cover of an Army Sector

From: D. C. Isby. Weapons and Tactics of the Soviet Army. London: Jane's, 1981. page 221.

Table 2-1. Soviet Army Air-Defense Radars

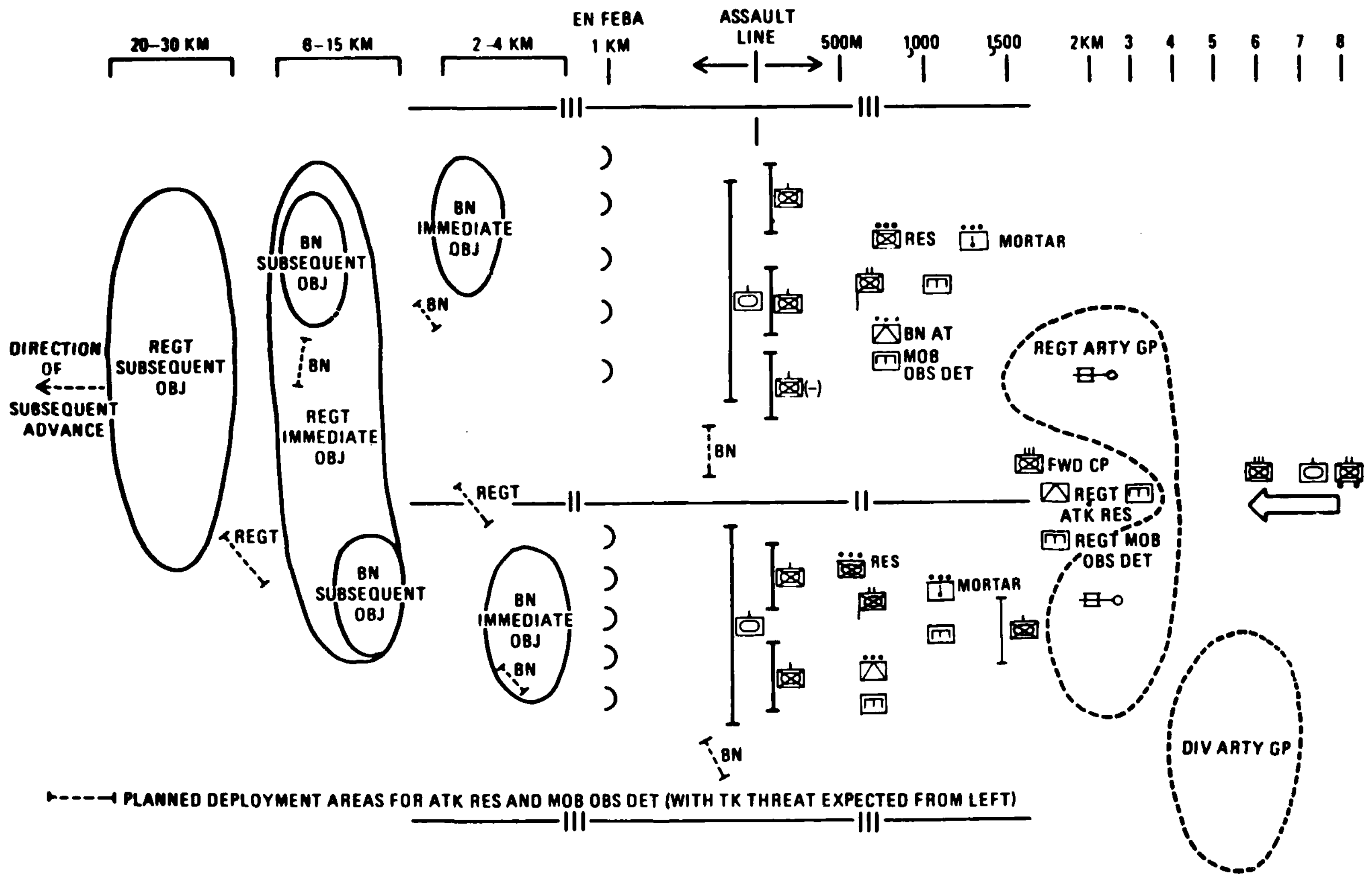
Name	Power	Frequency	Systems
Flat Face	400-500 kW	C band	Tk/MR Div AD Bde Nat'l AD
Squint Eye	400-500 kW	C band	Tk/MR Div AD Bde Nat'l AD
Spoon Rest	350 kW	A band 147-167 MHz	
Long Track		E band 2600 MHz	SA4,6,8 Bde/Regt SA4 Bn SA4,6 Bde/Regt HQ
Thin Skin A & B		H band	
Side Net		2560-2710 MHz	Tk/MR Div AD Bde Nat'l AD
Gun Dish	100-135 kW	J band	ZSU-23-4
SON-94 Fire Can	300 kW	E band 2700-2900 MHz	S-60 Btry
Flap Wheel		I/J band	S-60 Btry 130 mm AA gun
Fan Song A/B	600 kW	E/F band 2965-2990 MHz (A) 3025-3050 MHz (B)	SA-2 Btry
Fan Song D/F	1.5 MW	C band 4910-4990 5010-5090 MHz	SA-2 Btry
Low Blow	250 kW	I band 9000-9460 MHz	SA-3 Btry
Pat Hand		H band 6-8 GHz	SA-4 Btry
Straight Flush		D,G,H & I/J band	SA-6 Btry
Land Roll		G/H, I/J band 14200-14800 MHz	SA-8 launch vehicle

constructing an SA-2 template. Discrimination of any unit based on radars must deal with these ubiquitous air-defense radars.

Of course, a more complex entity, such as a command post, would require investigating inter- and intra-entity distances for a variety of equipment types -- radars, radios, vehicles, helicopters, etc. Missing equipment may be at least as important as sighted equipment, for an artillery battery may possess a radar of a type not used by the command post although they may be on the same radio net.

Having identified an artillery battery (perhaps by its radar type and location), templates are now needed to place that battery in its parent unit. This may be the regimental artillery group of a motorized rifle regiment (see Figure 2-3). Establishing proper unit subordination may be as important to correctly identifying the parent unit as it is to the proper placement of the subordinate unit in the force structure. The echelons (levels of grouping) at which templates for new units are defined create the hierarchical structure of the database that will eventually hold the intelligence derived using those templates. Thus stepping up one level, the regimental artillery group reappears in the motorized rifle division deployment illustrated in Figure 2-4.

Putting together the entire picture in which these templates must identify units, Figure 2-5 illustrates what NATO anticipates one standard German Scenario may be; it comes from SCORES IIA. A single corps would only be facing a portion of this picture; but even this is clearly a formidible task.



ATK RES = anti-tank reserve; TK = tank; MOB OBS DET = engineer obstacle detachment.

Figure 2-3. Motorized Rifle Regiment Deployment
Definitions of military symbols are given in Appendix E.
From: D.C. Isby. Weapons and Tactics of the Soviet Army.
London: Jane's, 1981, page 44.

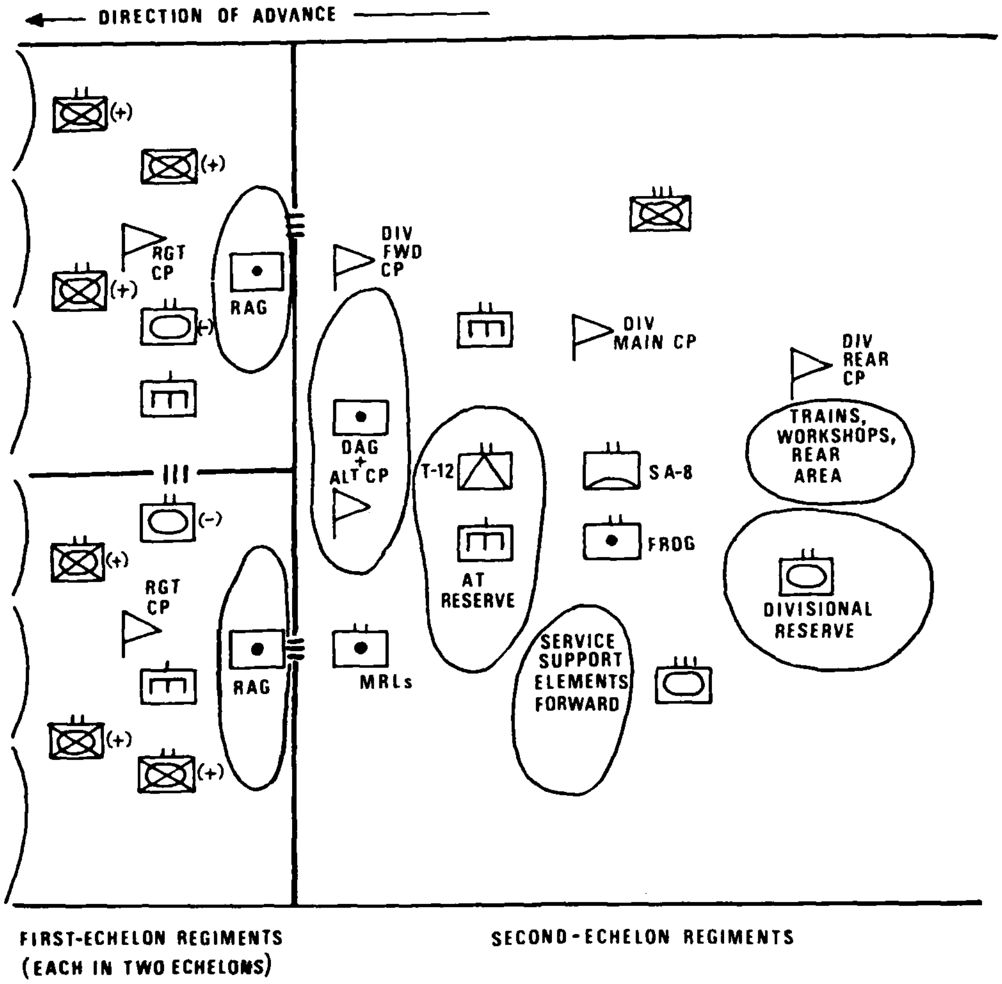


Figure 2-4. Motorized Rifle Division Deployment
Definitions of military symbols are given in Appendix E.
From: D.C. Isby. Weapons and Tactics of the Soviet Army.
London: Jane's, 1981, page 41.

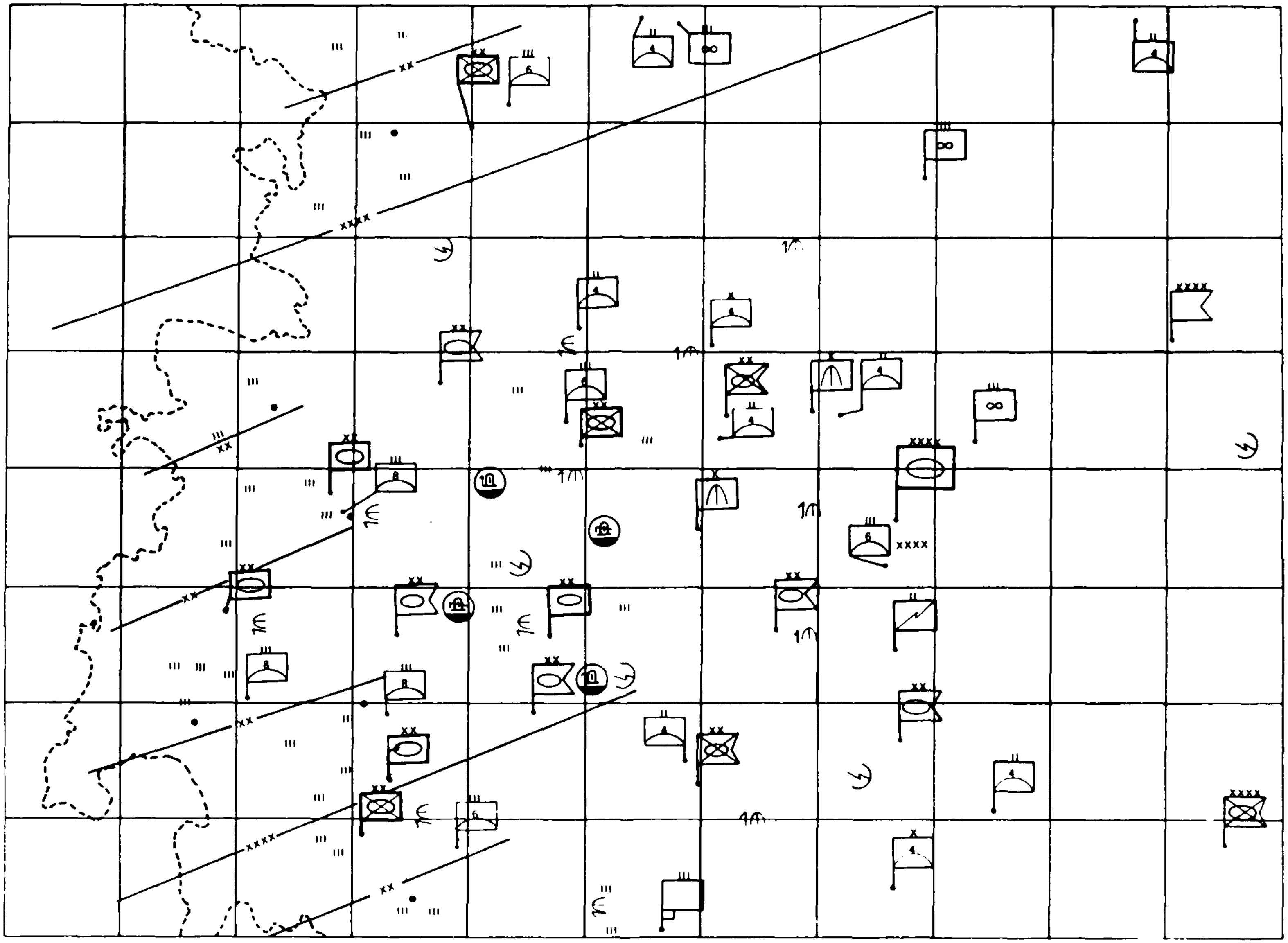


Figure 2-5. Enemy Ground Force Scenario (SCORES II-A)
Definitions of military symbols are given in Appendix E.
From: D.H. Howell. Final Report for Template-Assisted Intelligence Fusion Program (TEMPRO). Redondo Beach: TRW, 1979, page 11.

2.2.2 Situational Templating

For purposes of this report terrain, vegetation, and weather templating will be grouped into this category. In the preceding discussion, emphasis was placed on the difference between inter- and intra-entity distances. This difference is relied on for unit identification when using statistical separation tests, either alone or as one of a group of tests. In developed I/EW processing systems the intra-entity distance is usually given by a radius based on deployment prescribed by doctrine. However, a specific unit deployment will be accommodated to the terrain, leaving a range of possible radii. Futhermore, deployment of some emitters may be constrained by required wiring. If some of the larger intra-entity approach inter-entity distances, and it has been surmised they may for radars, identification algorithms may work poorly. To ameliorate this problem, other inter-entity distance measures may be chosen, promising better accommodation to specific terrain without losing doctrinal distinctions in overall unit "size." An example is the length of the minimum spanning tree. In other words, the mathematical form of the decision tests can inherit some of the burden of situational templating.

Moving units, in particular their vehicles, pose another problem. Doctrine will prescribe an inter-unit distance, for example, 3 to 5 km between maneuver regiments. Also, the number and type of major vehicles (thus their weight) are known; for example a tank regiment has 95 main battle tanks, 14 light trucks, and 119 standard and specialized trucks. But the intelligence picture will not be a "worm" of vehicles moving down the road, for Soviet doctrine also emphasizes cover and concealment. Thus, vehicles may regularly "pop-up" in a clearing in the foliage, giving scant information on which to identify a tactical march by fitting a doctrine-based template. Any hope of deciphering this march lies in good terrain and vegetation templating. Also, association with other activities, such as acquiring the SON-9 or Flap Wheel radar of the overwatching S-60 anti-aircraft gun (if this is part of a division movement), may aid in identification. This illustrates the time dimensions associated with templates, both a duration and a sequential separation of events. Of course, times also have both doctrinal and situational aspects.

2.3 REFERENCES

Isby, D.C., Weapons and Tactics of the Soviet Army. London: Janes', 1981.

Intelligence Preparation of the Battlefield. USAICS. 66000-A SupR, June, 1980.

Soviet Army Operations and Tactics. USAICS. 66444-1, SupR, November, 1982.

Soviet Army Troops Organization and Equipment. USAICS, 66444-3 SupR, November 1982.

SECTION 3

REVIEW OF DATAFLOW IN I/EW SYSTEMS

Cross-correlation is but one step in I/EW information processing and analysis. How data has been processed in previous steps affects the analysis performed by cross-correlation. This overall I/EW analysis, including the role of messages, will now be considered.

3.1 BRIEF REVIEW OF THE ENTIRE PROCESS

Figure 3-1 illustrates the general I/EW data collection and analysis process. Raw data is collected by sensors and processed at the associated analysis stations. This "single source" processing can be done automatically. by an operator/analyst, or, more often, by a combination called "operator aided" (the operator is aiding the computer by making some of the decisions). It may consist of data smoothing or determining a fix for more technical sensors, or report collection for HUMINT. For ELINT this initial data processing has been called "separation" as it separates out those reports associated with each emitter. This name may be found adequate or may be modified for the general case - "segregation" has been suggested by some authors.

The processed information is then passed into an all source processing system for integration with information from other sensor systems. The first operation is self-correlation; the integration of all data received about a unit into a record describing that unit. Since the original data was imprecise due to sensor error (totally ignoring deception), deciding which reports refer to the same battlefield entity is an art guided by some statistics and decision theory. Self-correlation is also carried out by some "intermediate" systems,

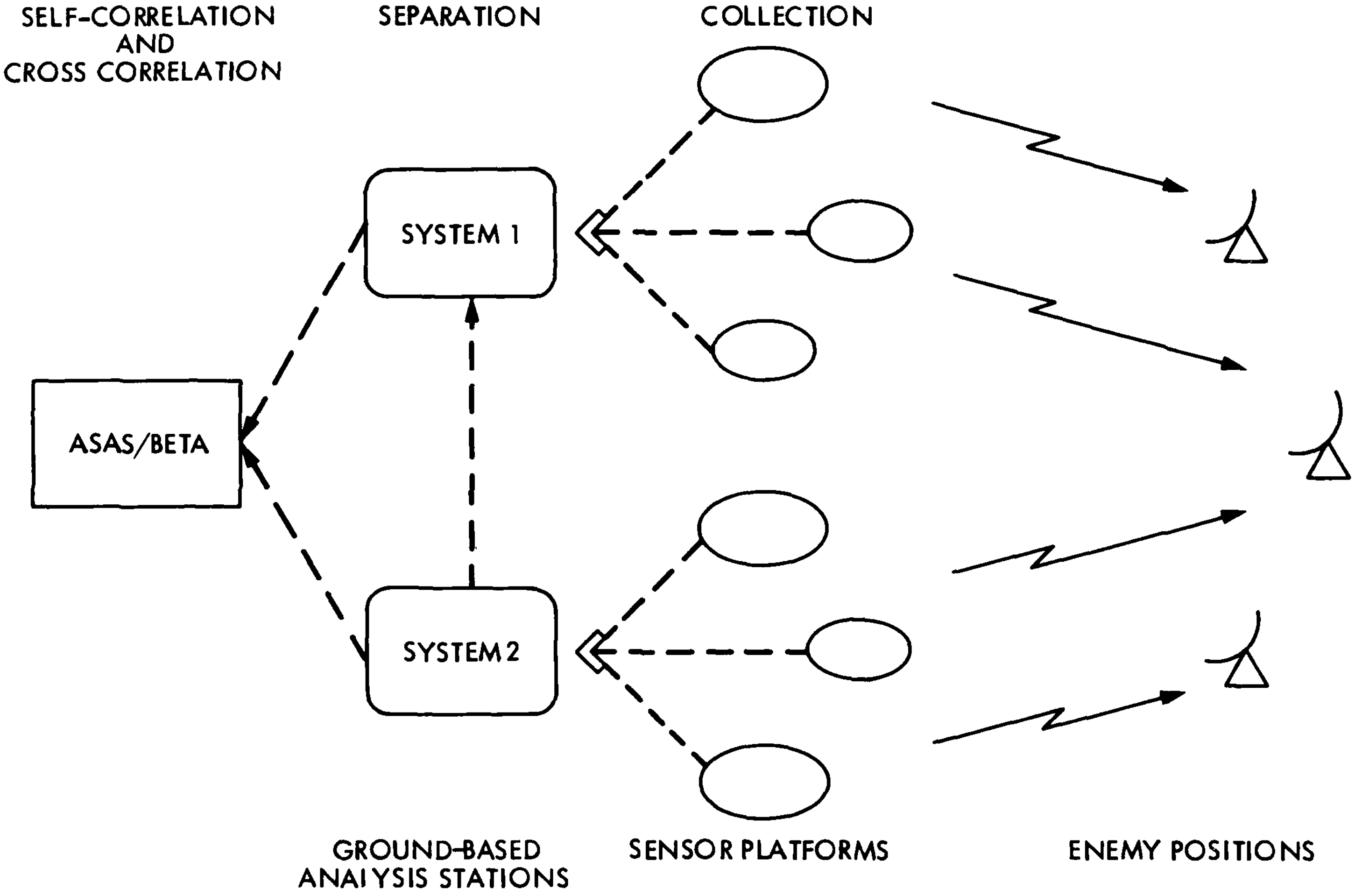


Figure 3-1. I/EW Data Processing Algorithms Overview

such as the Automated Ground Transportable Emitter Location and Identification System (AGTELIS), that use ınput from their own sensors and from analysis stations of other sensor systems. These "intermediate" systems, in turn, communicate their identified unit locations and characteristics to an ASAS.

After self-correlation is complete, cross-correlation begins. Records representing entities, such as radios and radars, or lower echelon units, are compared with those representing complex entities, such as command posts or higher echelon units, using templates to find the parent unit for the entity. Cross-correlation determines, for example, which radio belongs to which command post. The templates used are based on NATO perception of Soviet doctrine (different templates being required for a different adversary) and define a complex entity by the characteristics of the simple entities it contains. The dual process to cross-correlation, component collection, searches for simple entity "children" given a complex entity "parent." In the systems studied, the mathematics of component collection was the same as that of cross-correlation.

For cross-correlation, both parent and child entities are already defined; only the proper subordination is being determined. The third operation, aggregation, is naturally associated with cross-correlation. It is the initial identification of a parent complex entity from a group, or cluster, of children simple entities. Aggregation actually creates a newly identified unit from previously orphan entities. Aggregation will be covered in a later report.

As was discussed above, the entire tasking activity is beyond the scope of this report and has been ignored in this overview. The focus here is the "backward-flowing" data analysis activity, and, in particular, its associated algorithms, not the "forward-flowing" tasking activity.

3.2 INFORMATION INTERFACES

The information that survives to each stage of the intelligence process delimits the analysis that credibly can be performed at that stage. This applies to either the scope of analysis at any given level of data abstraction, or to the level itself. For example, if frequency is known only imprecisely at the sensor level, the scope or quality or reliability of the analysis is limited. However, if a positive identification of equipment item type can still be made, ASAS-level analysis may be only marginally affected. Some measure of the validity of such converted information should be made and transmitted with the information. Thus, every time data is processed - usually involving some sort of averaging - or whenever it is not sent from one processing stage to the next, information can be lost. Message formats are as important to the final results as the analytic processing. Messages must simultaneously satisfy two often conflicting criteria. They must:

(1) Contain all information ever needed.

(2) Be short enough to survive transmittal intact.

This section will first look at the information transmitted in the standard messages, thus input to an ASAS, then the "within ASAS" information flow.

3.2.1 Message Formats

In this section only Common, Intelligence, and Operations Control messages of the Joint Interoperability of Tactical Command and Control Systems (JINTACCS) will be considered. These messages, briefly described in Appendix C, are used for conveying a variety of intelligence activity reports.

Information carried in these reports from the following areas is commonly available to cross-correlation:

- The source as a coded identity.
- Source platform location, if applicable.
- General information about the siting:
  - Time.
  - Location.
  - Type of unit or equipment.
- Technical information (parameters) about the Signal of Interest (e.g., PRI).
- Enemy unit and activity:
  - Details about the unit being reported on (e.g., speed and direction of movement).
  - Activities of special significance.
- Free text.

Table 3-1 shows the pieces of this information conveyed by each message.

The above reflects but a small part of the message content. Most of the information supports general intelligence development and operations. Very few fields carry the more specific information used by cross-correlation: time, location, and type of unit sighted. Still fewer fields contain the specific technical information, for example describing signal parameters, used by self-correlation. As outputs from self-correlation are often inputs to cross-correlation, the availability of information to self-correlation effects the quality of information available to cross-correlation.

Table 3-1. Message Contents

	DISUM	INTREP	INTSUM	JRSRR	MISREP	SENREP	TACELINT	TACREP
The source as a coded identity		Y	Y		Y		Y	Y
Source platform location, if applicable				Y		Y	Y	
General information about the siting								
- Time		Y	Y		Y	Y	Y	Y
- Location		Y	Y		Y	Y	Y	Y
- Type of unit or equipment		Y	Y		Y		Y	Y
Technical information (parameters) about the Signal of Interest (e.g., PRI)							Y	Y
Enemy unit and activity								
- Details about the unit being reported on (e.g., speed and direction of movement)	Y	Y	Y		Y	Y		Y
- Activities of special significance	Y	Y	Y		Y	Y		Y
Free text	Y	Y	Y	Y	Y	Y	Y	Y

3.2.2 Interfaces Within an ASAS

An ASAS can talk to itself and its clones. Its input and output data cross the same information interface. Its output also becomes input for other ASAC-level systems, such as MAGIS. Figure 3-2 focuses on the functions and data passing interfaces in the overall I/EW system. The functions enclosed in the solid box represent those usually found in an ASAS; those in the dotted box are occasionally found in an ASAS system. "Identification" means identifying a radar from its signal parametrics as well as a unit from its components. Automatic creation and identification of a unit from many, possibly unrelated, components fits in "Aggregation." Note that the interface between self- and cross-correlation lies in both boxes. Comparing Figure 3-2 with Figure 3-1, which was heavily influenced by what systems perform the different functions, shows there is no clear match between system and function level. Different systems perform different, usually overlapping, functions.

An ASAS system may require technical inputs for self-correlation that are not required for cross-correlation. Indeed, self-correlation assumes some of the burden of the initial translation of technical characteristics into militarily recognizable objects, refining this identification as more information is collected. Thus, the difference between self- and cross-correlation is reflected in their respective information requirements. Cross-correlation is more concerned with generic, qualitative, descriptive characteristics of elements than with very technical quantitative ones. Characteristics usually considered (e.g. in BETA) are:

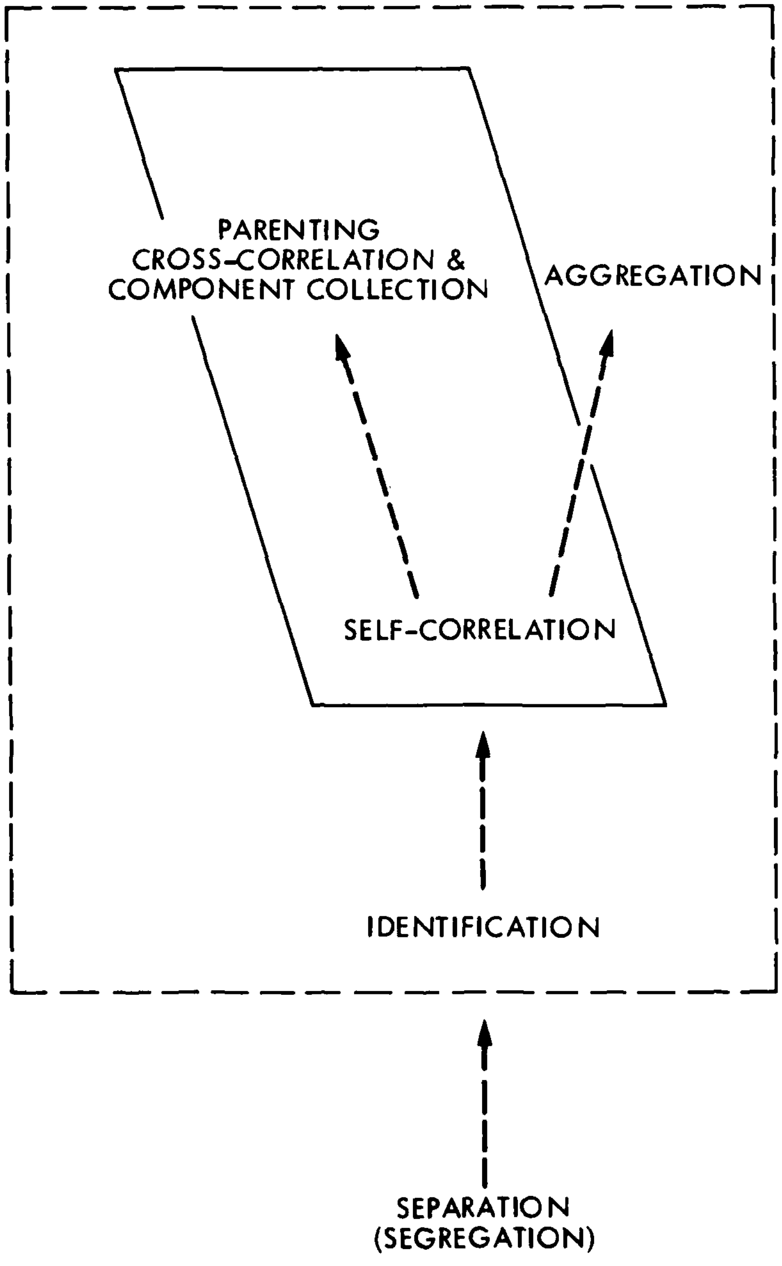


Figure 3-2. ASAC Level Information Flow

(1) Entity location.

(2) Entity type (e.g. Fan Song radar, artillery battery).

(3) Time of sighting (oldest, most recent, etc.).

Processing within the cross-correlation module usually does not affect the above information. It only establishes the links reflecting subordination between units.

The inputs to cross-correlation are thus the entity for which a parent (or child) is to be sought and the database of complex entities. Its outputs are that database, with the new links established if appropriate, and a message, if the additional information significantly increases the military importance of the parent complex.

3.3 EFFECT OF PREVIOUS PROCESSING

Previous processing by automatic I/EW systems significantly influences the type and quality of information input to cross-correlation. Since the kind of algorithms chosen for a system depend as much on the information available as on the answers required, it is important to understand the effects of this previous processing. This section will focus first on functions usually performed in sensor-level systems, then on those usually associated with self-correlation. Some of these issues have been reported on in greater depth in the report on self-correlation.

3.3.1 Sensor System Processing

Stationary sensor platforms are composed of multiple antennas, each with its own biased error distributions, with associated ground (single-source analysis) stations to perform separation (or segregation). Locations are derived from D.F. fixes from the many antennas, perhaps having different receiver and line-of-bearing indicator equipment and different operators. Non-stationary sensor platforms are composed of single antenna structures, again with a ground station for processing. D.F. fixing is accomplished by processing lines-of-bearing taken by the same antenna structure from different sensor platform positions. For stationary sensor platforms, the statistical measurement characteristics of the individual antennas (and support equipment and operator variability) may differ, whereas the moving sensor platform has "fixed" statistical measurement characteristics. Any measurement-induced biases will affect any sensor platform separation and segregation processing but will be transparent to the user of the messages generated by such processing.

Different bias distributions are obtained from ground-based and airborne configurations. In addition to individual operator and equipment biases, ground sensors also will be subject to biases resulting from terrain: varying absorbtion and reflection depending on soil, vegetation, and topography, and masking effects. Terrain factors remain relatively constant for the life of a given sensor deployment configuration, thus the reports generated are internally consistent. However, all the biasing environmental influences change when the sensor moves to a new location; and since different sensors move at different times, some have constant error distributions while that of others (perhaps looking at the same enemy entity) have changed. For a given airborne mission, these terrain factors may be viewed as either constant or, often, random. However, airborne sensors have a different source of platform- and mission-dependent bias. Since the platform is moving as the line-of-bearing is being determined, the aircraft position is known with an error partly dependent on the track and speed of the aircraft.

Thus, several potential sources of data degradation, unreported in standard messages, arise through initial processing and are seen to "add on" to each other, so earlier problems cannot be corrected later. These sources include:

- The (seemingly) unreported, shifting biases.
- Reporting average or estimated values, not distributions of the actual data sensed and the number of sensings.
- The increased correlation among the data as it progresses through the separation and segregation process.

These can make the usual tacit assumptions (statistically independent samples and known [sample] variances) made in self-correlation inappropriate. More complete analysis would require that statistical descriptors accompany estimated parameter values in messages. The appropriateness of such analysis depends on the accuracy required by the higher level processors; current systems seem to seek greater accuracy than the input data warrants.

3.3.2 Self-Correlation

Since the information entering an ASAS is first processed by self-correlation, any problems systemic to self-correlation are inherited by cross-correlation. As just discussed, sensor processing can also introduce a few red herrings and its own specific biases, and these biases are unique to each sensor system. But sensor software is outside an ASAS and not available to its control or correction. Self- and cross-correlation, on the other hand, both reside in an ASAS and should be mutually supportive. This section examines several potential sources of error.

A primary role of self-correlation is to match enough reports to give the operator a readable screen, that is, a graphical display not too dense with sightings. At various junctures operators use their graphic displays to intervene and make complex decisions, often aided by graphics. These graphics usually consist of the entity data record information being mapped on the display, overlaying a terrain map of the region. If self-correlation has not correctly matched reports to entities, then either too many units will be displayed (as several existing simple entities really refer to the same unit) or too few are displayed (as records were judged to refer to the same entity that in fact did not). In the first case, the operator may be able to resolve the inaccuracy, if his screen is not too crowded with all the extra entities to see anything. What he will probably see is a few units clustered around the unit's actual location. The second case is worse. Records from different units are combined and their locations integrated, making the location shown on the screen a phantom; also, several locations that should be shown are most likely not. This situation is usually neither retrievable, nor even detectable by the operator; units located in unlikely terrain may indicate phantoms.

When using graphical displays as an aid in the more complex cross-correlation decisions, either of the above cases is harmful; and the integration of unlike records is probably not correctable except where the sightings supporting each location have been kept and linked to it by pointers. Whether it is feasible either to store so many reports or take the time to unravel and reform locations is open to question.

Another problem arises when parametric information from two sitings of the same entity is combined. Means of like things usually combine into meaningful averages despite a poor environment. Thus, as long as entity characeristics are kept as like means, they probably are good. Variances, however, require more careful handling and may easily be biased if the proprieties, in the form of assumptions, are not observed. Improper variances can lead to improper associations in cross-correlation, if they have not already in self-correlation.

If the "means" are not like - that is, if two different measures of central tendency are used by two sensor systems - then combining them may lead to nonsense. The nonsense will be within the range of the data, and look quite real to the analyst, who has no indication that the software is misleading him. But this "mean" value will no longer be a measure of central tendency, and thus will not represent anything real and may be misleading when used in decision making. This again leads to cross-correlation receiving faulty information. This problem can be addressed by self-correlation knowing what type of measures (of central tendency and dispersion) to expect from each sensor system, and keeping more than one measure to describe an entity, for example mean, mode, and interquartile range.

Another problem arises from the implicit assumption that all sensors are equally likely to see a target - that sighting is a type of random event. In reality, different sensors within a given category of target, e.g. TRQ-30, TRQ-32, and TRAILBLAZER, will often have different coverage envelopes in terms of, for example, frequency range, duty cycle, and modulation type capability. This affects the likelihood of intercept. A cross-correlation algorithm could require information from the collection management process in order to properly analyze data from such a sensor suite. To take into acccount, for example, the likelihood of intercept for different node signature components requires both making the correct assumptions when developing the algorithm, and the availability of information on what parameter envelopes are being covered at a given time and by how many sensors.

SECTION 4

ANALYSIS OF CURRENT SYSTEMS

As indicated in Section 2, there are two aspects to implementing cross-correlation: the algorithms themselves and their interface with the system, especially the databases of entities and templates. This section will therefore look at three aspects of cross-correlation as it is implemented: first, the functioning of the cross-correlation systems as they have been developed; second, the type of mathematics they use; and third, the interface with and demands on system and database architecture.

4.1 ALGORITHMS IN CURRENT SYSTEMS

BETA cross-correlation "algorithms in standard form" (written in Pascal) are given in Appendix A. Database entries and structure reports for the cross-correlation modules of BETA and TEMPRO are given in Appendix B.

4.1.1 BETA

Figure 4-1 illustrates the flow of the BETA cross-correlation process. This discussion will focus on processing one entity data record (EDR) of the several a message may contain. The input-modified flag, dealing with data protection in a multi-user environment, will not specifically be considered. The implementation of BETA studied was that for which source listings were available (see references in Section 4.1.5).

The BETA cross-correlation process searches for a potential parent for an input EDR. First, the list of all potential parent templates, that is,

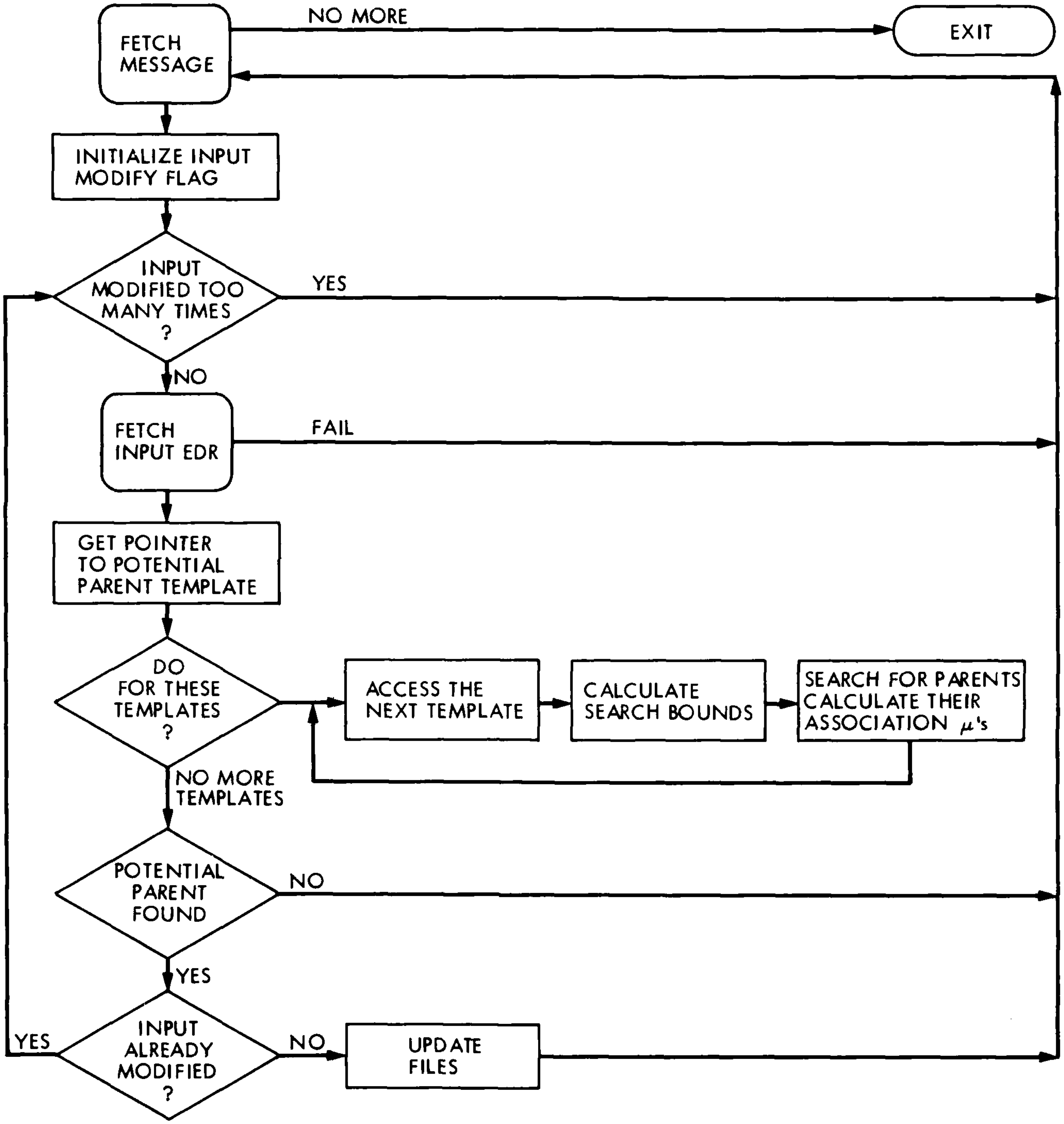


Figure 4-1. BETA Cross-Correlation Module

templates that have the subject type (radar. radio, artillery battery, etc.) of the input EDR as a possible subordinate, is fetched. For each template, the error ellipses (see Figure 4-2) and resulting search bounds are calculated. These bounds depend on both the amount of uncertainty in the input EDR location as reflected by the major and minor error ellipse axes, and the expected deployment of a unit described by the minimum and maximum radii given in the template. Next, the database is searched for EDRs of the template subject type that fall within the search bounds. An association measure is calculated for each such potential parent. If this association measure exceeds the acceptance threshold given in the template, the potential parent EDR is a candidate for the "acceptable" list; if it falls below the reject threshold, processing continues, looking at the next potential parent; and if it falls between the two thresholds, it is a candidate for the "ambiguous" list. If the acceptance threshold is set equal to the rejection threshold, there are no ambiguous candidates. The acceptable candidate with the highest association measure is usually linked with the input EDR, becoming its parent.

The association measure for complexes (e.g., command posts) is a weighted sum of three figures of merit: an affiliation score, a time decay score, and a location score. The weights are given in the template. The association measure score for a compound has no affiliation component. For example, a tank is compounded of a tracked vehicle, a radio, and a gun. It is interesting (and probably accidental) that these three figures of merit can be related back to the three criteria discussed in Section 2:

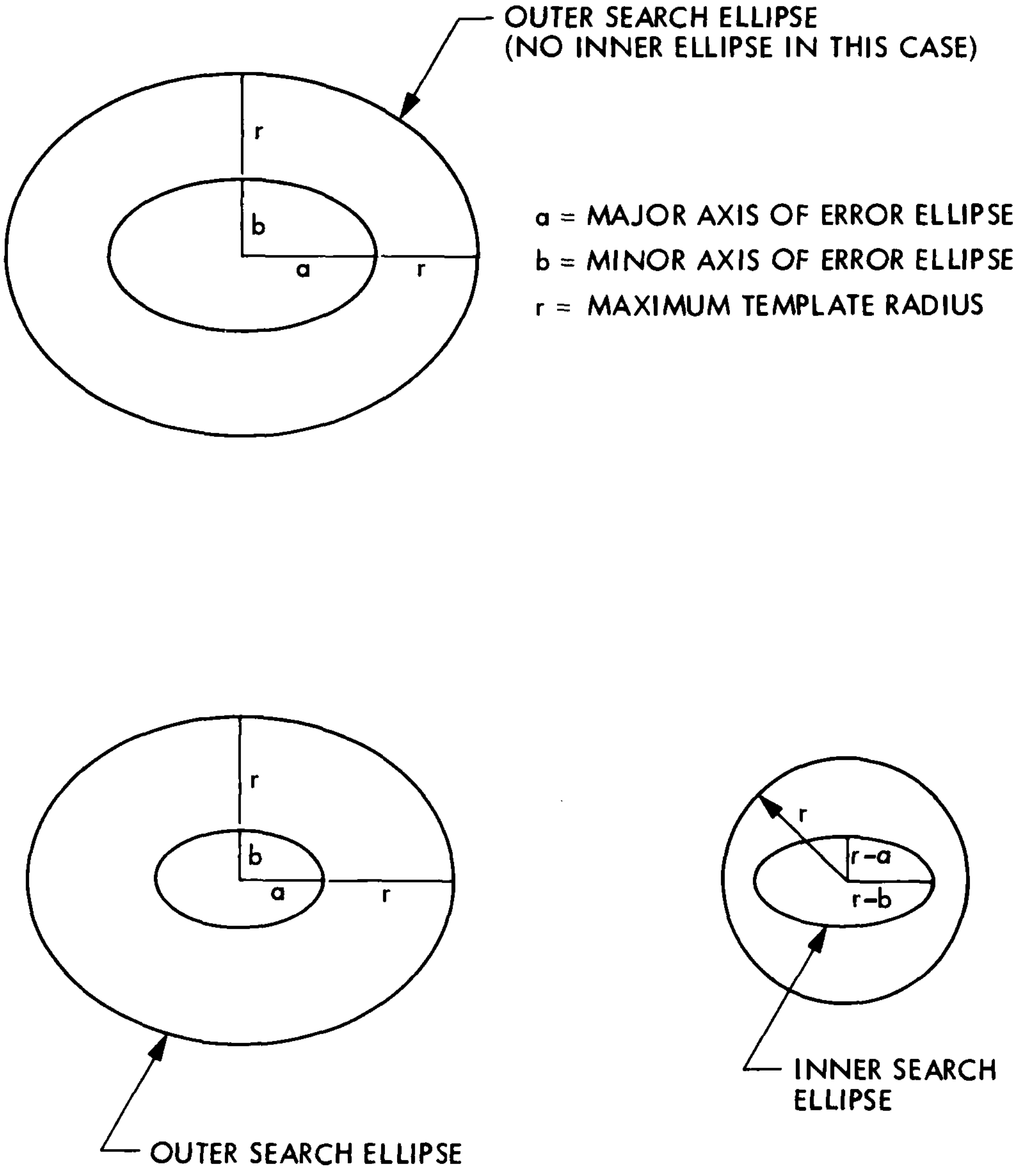

OUTER SEARCH ELLIPSE
(NO INNER ELLIPSE IN THIS CASE)
r
b
a
r
a = MAJOR AXIS OF ERROR ELLIPSE
b = MINOR AXIS OF ERROR ELLIPSE
r = MAXIMUM TEMPLATE RADIUS
r
b
a
r
r
r-a
r-b
INNER SEARCH
ELLIPSE
OUTER SEARCH ELLIPSE

Figure 4-2. BETA Error Ellipses

Accuracy of unit location	:	location score
Timeliness of response	:	time decay score
Certainty of unit identification	:	affiliation score

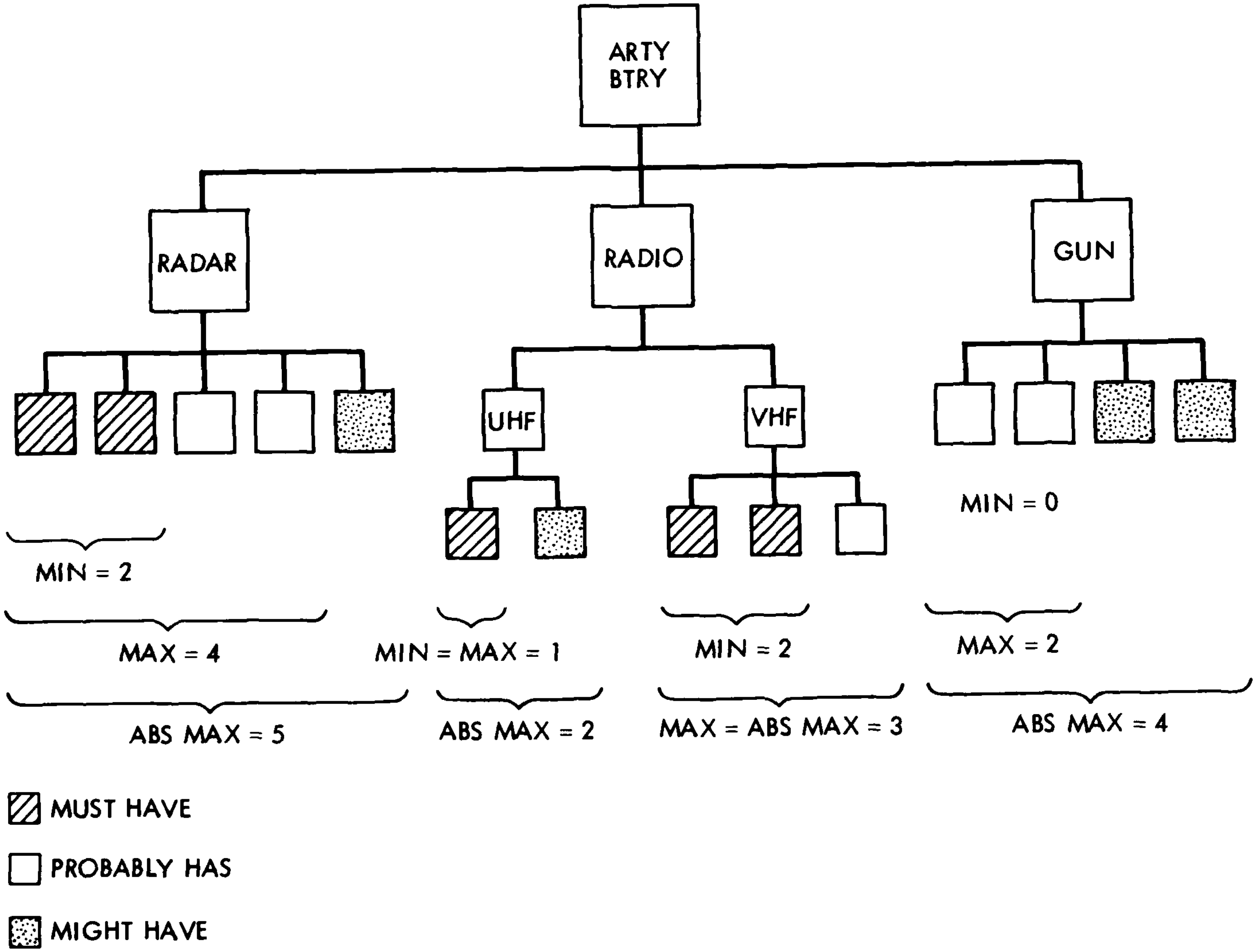
Each of these scores will now be discussed separately.

The affiliation score reflects the probability that the parent entity indeed has another element of the input type that has not been found. If an artillery battery is "known" by doctrine and specified in the template to have at least two radars, and none have been found, then the probability it has some still to be found is very high. Conversely, if it is "known" to have at most four radars, and at least four have been found then, assuming the four radars already found have been associated correctly with that parent, the probability there is still another radar to find is very low. This relation between affiliation score and the unit template is illustrated in Figures 4-3(a) and (b). These illustrations were developed for explanation only, and do not reflect any real TO&E.

To calculate the affiliation score, let max be the maximum number of things of the input type expected and min the minimum number, both values taken from the template. Assume the input EDR does actually belong to the candidate parent, and let n be the resulting number of children of the input type. Then

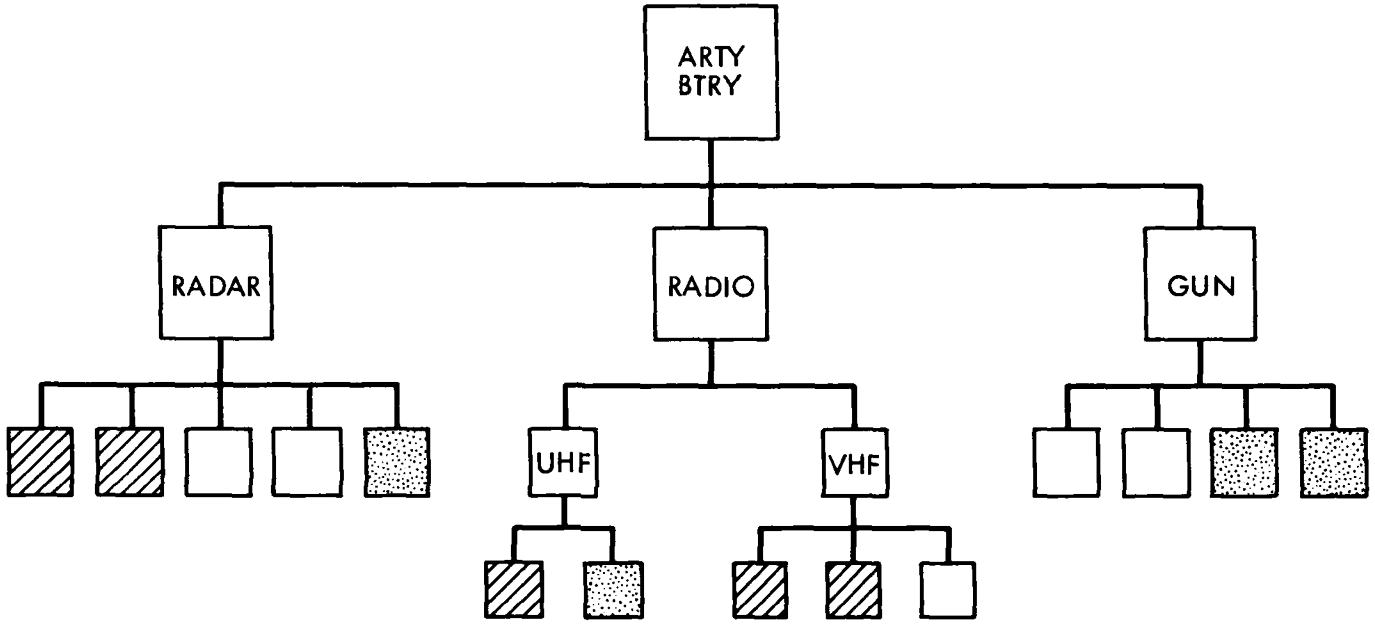
$$\text{affiliation score} = (\text{max}-n)/(1+\text{max}-\text{min})$$

when n falls between max and min, 0 if n is greater than or equal to max, and 1 if it is less than min. Thus, in the example (Figure 4.3), suppose a new radar is found and a candidate parent already has 2 radars. From the template,



(a) EXAMPLE TEMPLATE DEFINITION

Figure 4-3(a). Affiliation Score



PARENT HAS:	0	1	2	3	4	0	1	0	1	2	0	1	2	3
SCORE IS:	1	1	0.33	0	0	1	0	1	1	0.5	0.33	0	0	0

(HIGHER COMPONENT COUNTS ALL RESULT IN ZERO SCORES)

(b) EXAMPLE SCORE: PICTORIAL REPRESENTATION

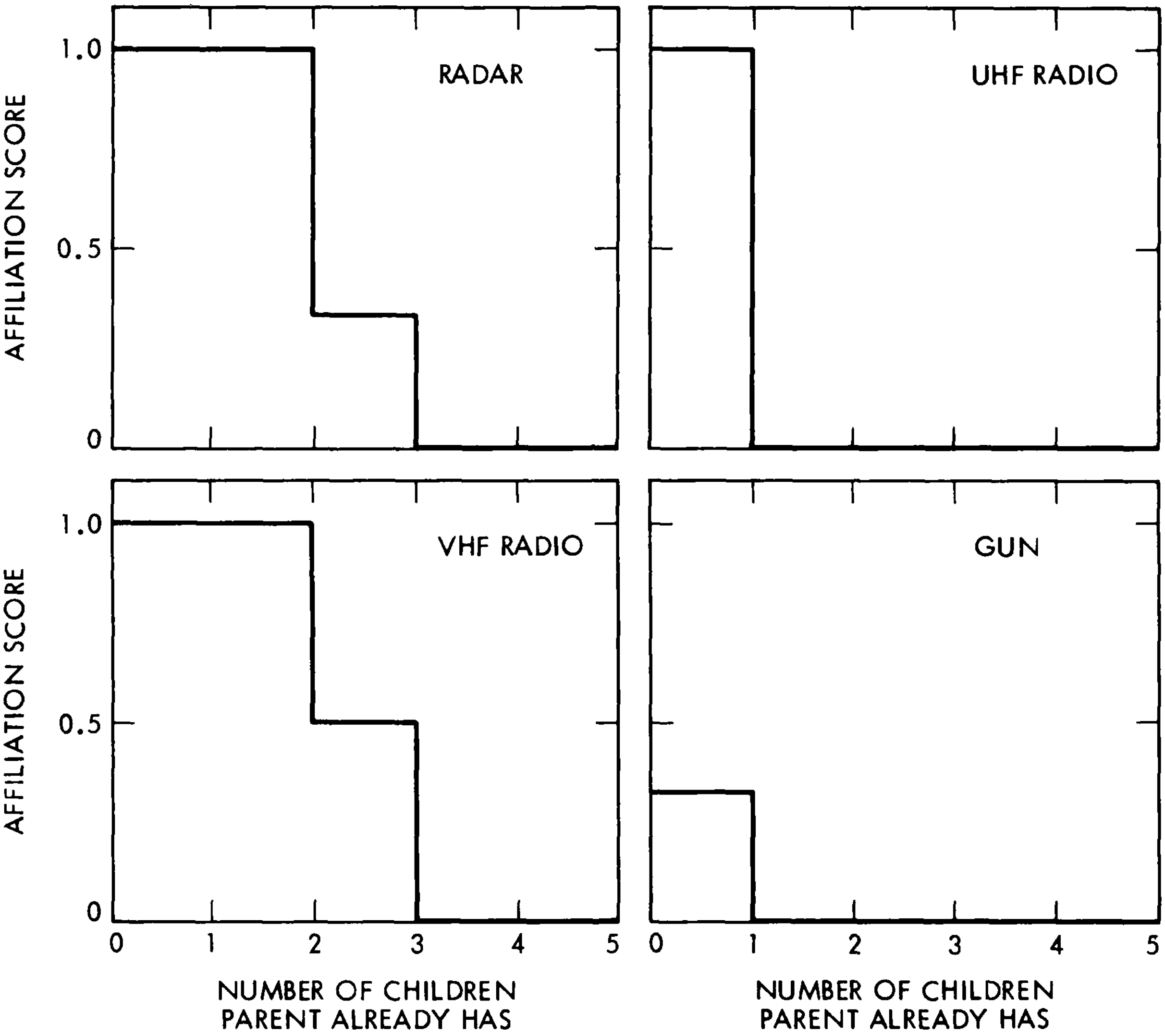
Figure 4-3(b). Affiliation Score

min = 2 and max = 4, so

$$\text{affiliation score} = ((2+1)-2)/(4-2+1) = 0.33.$$

For VHF radios, since min = 2 and max = 3, if a candidate parent has zero or one VHF radio, with a new sighting it has less than or equal to two VHF radios so the affiliation score is 1; if it already has two or more, then with a new sighting it has greater than or equal to three VHF radios and the affiliation score is zero. As these examples indicate, the inclusion of the "+1" in the denominator and the "or equal to" underlined above is interesting, as it makes the affiliation score "jump" to zero when n is still within the template-specified limits. This is illustrated in Figure 4-3(c). The resulting non-linearity in the score means it can not be interpreted as coming from a uniform distribution; that is. it is not considered equally likely that any number of entities in the template-defined range have been observed, but more likely that fewer have.

In some cases, because both input and parent EDRs are complexes. there appears to be a way to represent them by a subordinate object type e.g., an artillery battery in a regimental artillery group (RAG) may still be associated with its radars. In this case, the above rules for calculating the affiliation score would still hold, but total radars in the RAG would be counted instead of total artillery batteries. A multiplicative factor, depending on the number of objects in the parent. is used to convert the template-determined minimum and maximum number of batteries to bounds on the number or radars. The interested reader is referred to "Procedure ASCEAM" in Appendix A. In the code, this multiplicative factor is IQUANY, "number of objects represented by the candidate EDR." Remember this candidate is the parent. Since the number of children



(c) EXAMPLE SCORE: GRAPHICAL REPRESENTATION

Figure 4-3(c). Affiliation Score

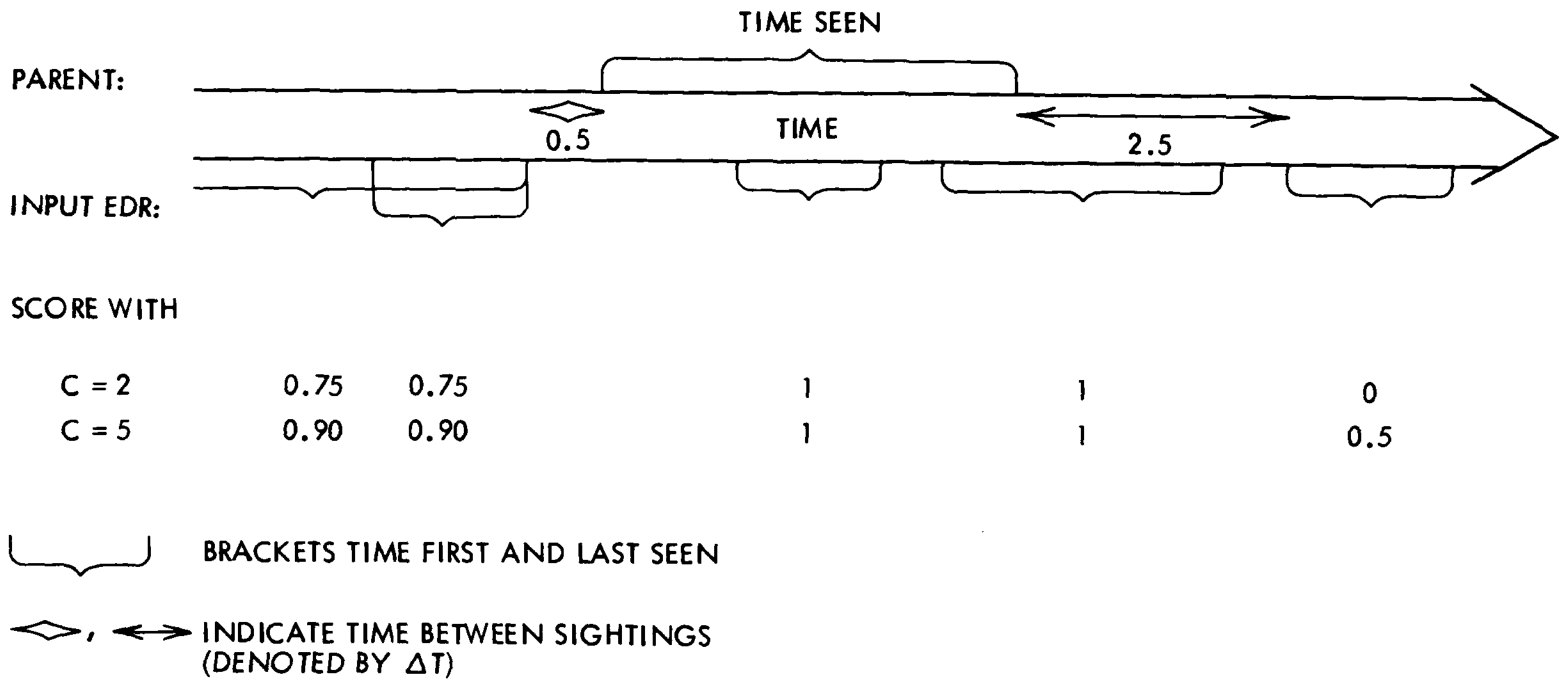
is multiplied by IQUANY, "number of objects in a specific type of child" would seem a more consistent definition. As it stands, the code is difficult to interpret.

The time decay score is based on the closed intervals representing the elapsed time over which the entity has been observed for both candidate and input EDR. As illustrated in Figure 4-4(a), if these intervals for the input and potential parent EDR overlap, the time decay score is 1. If not, and the elapsed time between the two intervals, t, is less than the time constant C given in the template,

$$\text{time decay} = 1 - t/C$$

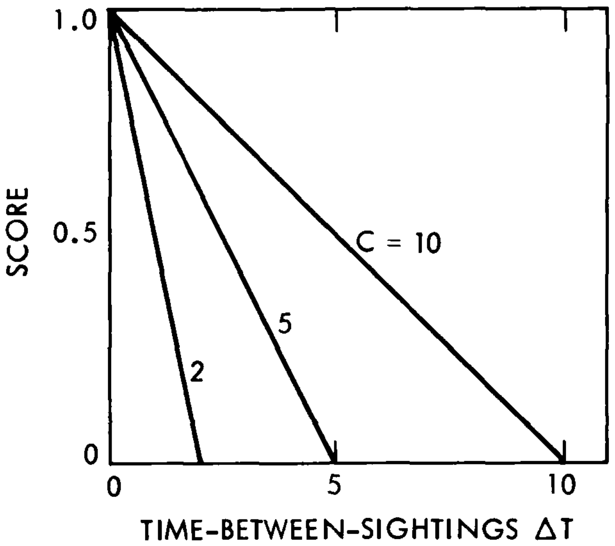
If t is greater than C, this score is zero. Figure 4-4(b) shows the influence of this constant on the time decay score. The time constants in the templates and the reported sighting times are assumed to be in the same units.

The location score is calculated by one of two methods, again governed by the template. The first, used if a non-zero maximum deployment radius for that parent relative to that subordinate type is given, is illustrated in Figure 4-5. For this method, if the distance between the estimated locations of the input and candidate parent EDRs, d, lies between that maximum radius and the template supplied minimum deployment radius, the location score is 1. If d is less than the minimum radius (min r)

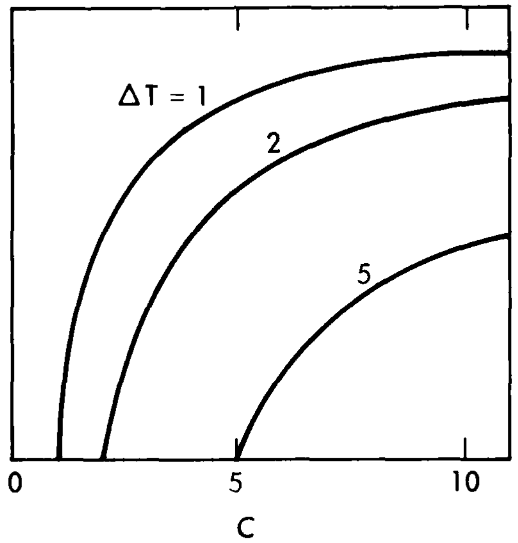


(a) EXAMPLE SCORES

Figure 4-4(a). Time Decay Score



BEHAVIOR OF AFFILIATION SCORE FOR FIXED TEMPLATE TIME FACTORS C AS THE TIME-BETWEEN-SIGHTING INTERVAL ΔT VARIES



BEHAVIOR OF AFFILIATION SCORE FOR FIXED TIME-BETWEEN-SIGHTINGS INTERVALS ΔT AS THE TEMPLATE TIME FACTOR C VARIES

(b) DEPENDENCE ON THE CONSTANT C

Figure 4-4(b). Time Decay Score

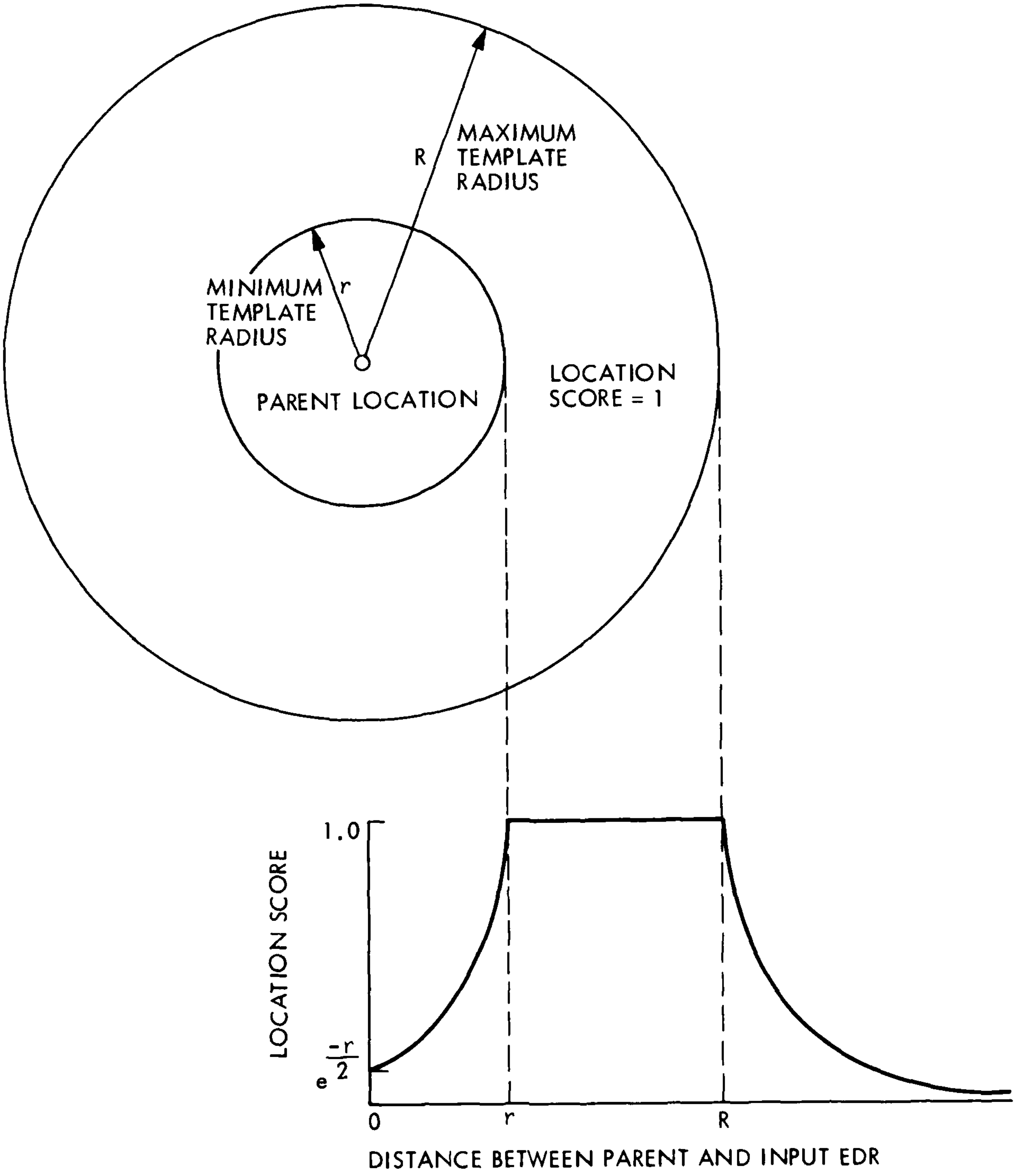


Figure 4-5. Location Score

$$\text{location} = \exp\,[-(\min r - d)/2],$$

and if it is greater than the maximum radius (max r),

$$\text{location} = \exp\,[-(d-\max r)/2].$$

This score uses the dispersion (uncertainty in the location estimate) of neither the parent nor subordinate entity.

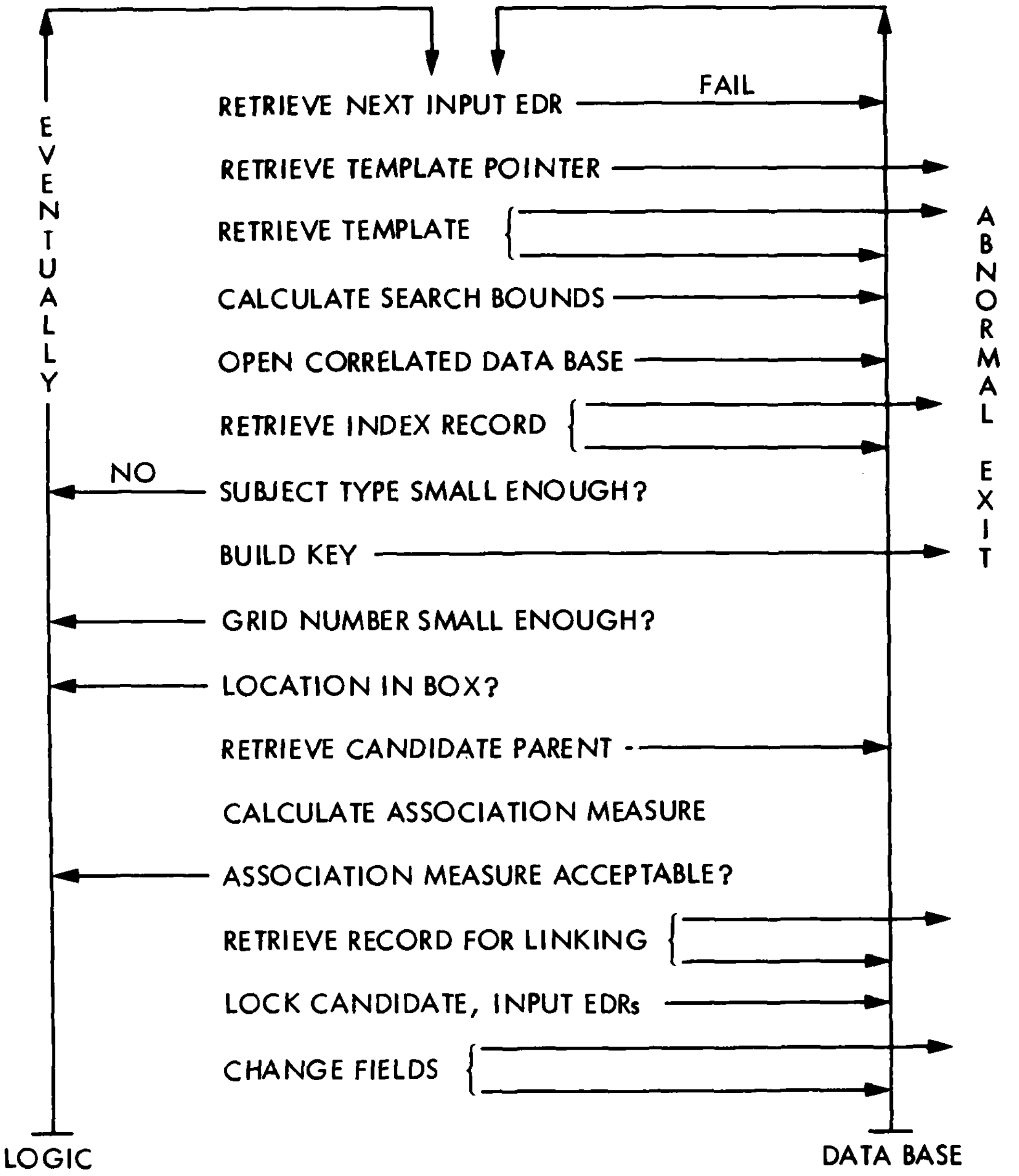
If no maximum deployment radius is given, then the second method is used, setting

$$\text{location} = \exp\,(-X/2)$$

where X is the usual quadratic form d^TCd, C the inverse of the sum of the covariance matrices for the EDR and parent. Although this method takes into account the dispersion (although ignoring differences in sample size), it assumes that the parent's deployment radius is indeed essentially zero.

The location score is completely dependent on the units in which distances are reported.

As shown in Figure 4-6, these figures of merit are embedded in interlocking cycles of database accesses and decision logic. The entire process is at the mercy of the database management software; there are far more places where processing may abort due to an unsuccessful attempt to access the database than there are terminations in the decision logic due to unsuitable candidate



EVENTUALLY
RETRIEVE NEXT INPUT EDR
FAIL
RETRIEVE TEMPLATE POINTER
RETRIEVE TEMPLATE
CALCULATE SEARCH BOUNDS
OPEN CORRELATED DATA BASE
RETRIEVE INDEX RECORD
NO
SUBJECT TYPE SMALL ENOUGH?
BUILD KEY
GRID NUMBER SMALL ENOUGH?
LOCATION IN BOX?
RETRIEVE CANDIDATE PARENT
CALCULATE ASSOCIATION MEASURE
ASSOCIATION MEASURE ACCEPTABLE?
RETRIEVE RECORD FOR LINKING
LOCK CANDIDATE, INPUT EDRs
CHANGE FIELDS
ABNORMAL EXIT
LOGIC
DATA BASE

Figure 4-6. BETA Sequences Terminating Processing of an EDR

attributes. Moreover, in some of the abort sequences the input message seems likely lost for good. Figure 4-6 also shows that an ordered structure is assumed for the correlated database of parents, as grid number and subject type searches are discontinued when the candidate parent value exceeds the search bound.

Now consider once more the hierarchical levels defined by the templates. If the database/template designer chooses many levels, more directories and pointers must be kept for data storage and retrieval. This will increase both storage space and processing time, especially for searches. If only a few levels are chosen, with many subordinate entity "children" at each level, it is important that all the children of the same parent are at a similar hierarchical level. In particular, complex units and equipment should never be mixed as children of the same parent. Figure 4-7 illustrates one problem that could arise if equipment and units are mixed. It may be difficult to determine what equipment belongs to a child, and what to the parent. This can easily lead to improper affiliation scores, and perhaps even to misidentified complexes. Even a "unique" piece of equipment can be assigned to some home (a CP, a Recon Company), and the logic and mathematics of the association process will operate more accurately and efficiently without these equipment orphans.

4.1.2 TEMPRO

Figure 4-8 illustrates the operation of TEMPRO Sequential Association. This process attempts to find the following items for a unit (for whom there is new information):

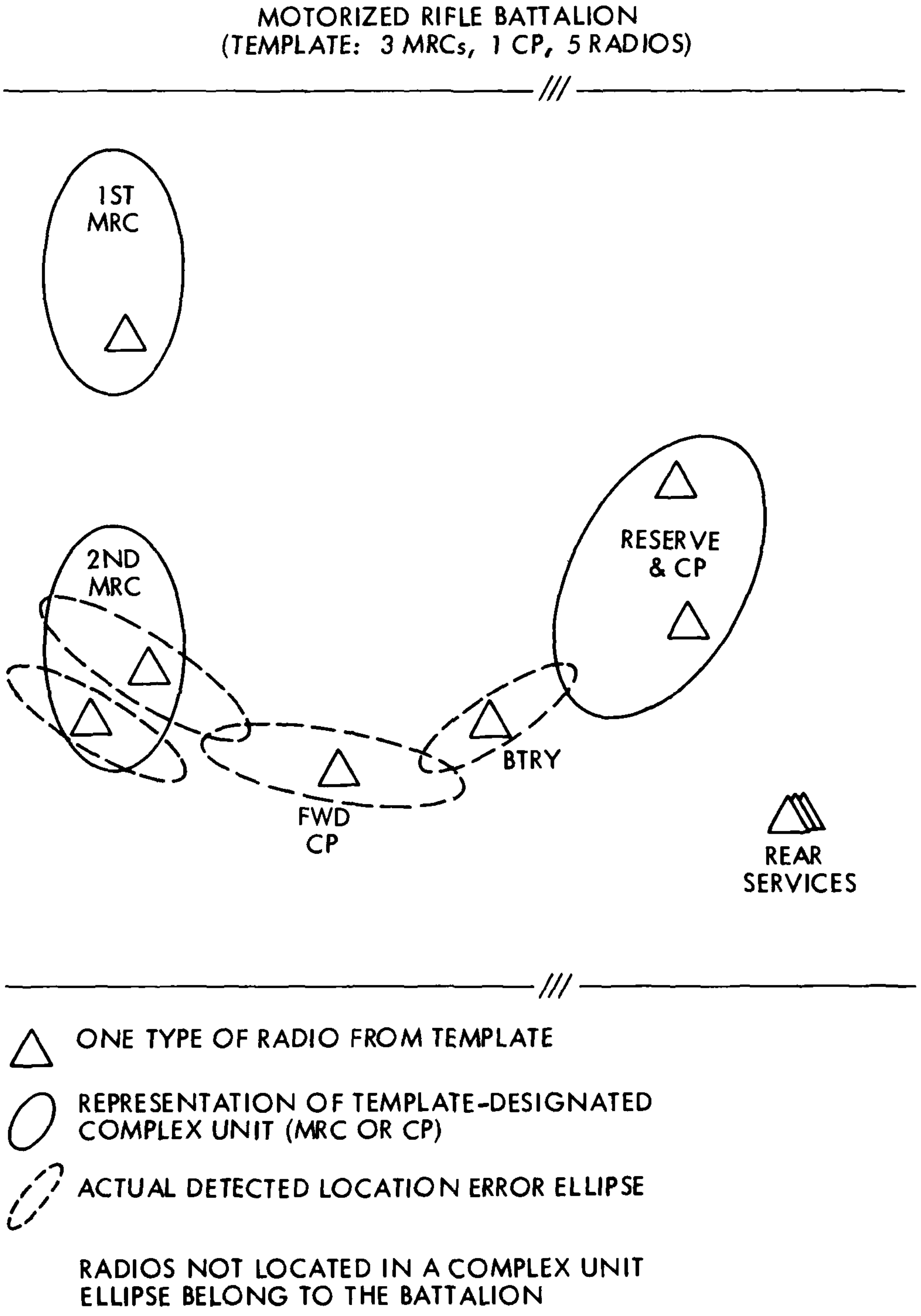

MOTORIZED RIFLE BATTALION
(TEMPLATE: 3 MRCs, 1 CP, 5 RADIOS)
1ST MRC
2ND MRC
RESERVE & CP
BTRY
FWD CP
REAR SERVICES
ONE TYPE OF RADIO FROM TEMPLATE
REPRESENTATION OF TEMPLATE-DESIGNATED COMPLEX UNIT (MRC OR CP)
ACTUAL DETECTED LOCATION ERROR ELLIPSE
RADIOS NOT LOCATED IN A COMPLEX UNIT ELLIPSE BELONG TO THE BATTALION

Figure 4-7. Mixed Level Template Syndrome

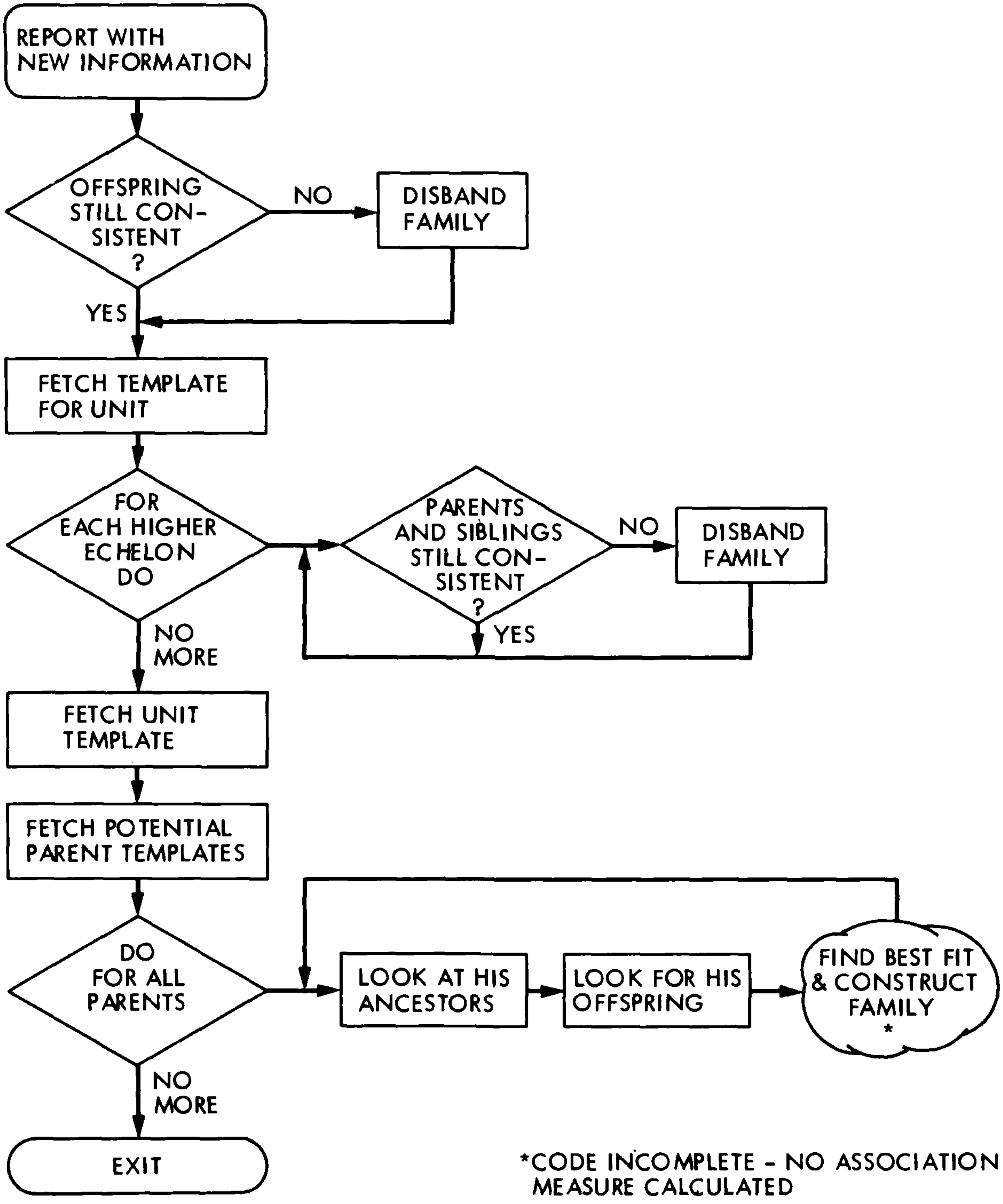

REPORT WITH NEW INFORMATION
OFFSPRING STILL CON-SISTENT ?
NO
DISBAND FAMILY
YES
FETCH TEMPLATE FOR UNIT
FOR EACH HIGHER ECHELON DO
PARENTS AND SIBLINGS STILL CON-SISTENT ?
NO
DISBAND FAMILY
YES
NO MORE
FETCH UNIT TEMPLATE
FETCH POTENTIAL PARENT TEMPLATES
DO FOR ALL PARENTS
LOOK AT HIS ANCESTORS
LOOK FOR HIS OFFSPRING
FIND BEST FIT & CONSTRUCT FAMILY *
NO MORE
EXIT
*CODE INCOMPLETE - NO ASSOCIATION MEASURE CALCULATED

Figure 4-8. TEMPRO Sequential Association Module

- Offspring.
- A parent.
- New siblings.

There are two association tests, performed sequentially:

(1) A consistency test.

(2) A measure of association.

If the parent and siblings group found are inconsistent, the one with the highest association measure with the input unit is kept.

Consistency is determined by matching type, subtype (see Figure 4-9), and level (Figure 4-10). An activity descriptor (Figure 4-11) is also kept but does not seem to be used in consistency checking. If these three parameters do not match, the units are not consistent. The parameters are sometimes considered matching when they are not equal. For example, if "tank" is the type, any headquarters unit may be the offspring of a "tank" unit of another subtype.

The mathematical measures of associations are not given in the 1978 source code referenced. Their subroutines are stubs.

The classification of units by type, subtype, level, and activity deserves further attention. Such a multidimensional system gives greater flexibility in consistency rules and database structure than the more commonly seen one-dimensional tree structures. Updating and refining information is also easier when any one dimension can be changed separately. Even the activity code could be used in consistency checking, as long as time is taken into account.

TYPE CODES

1 = MR - Motorized Rifle
2 = TNK - Tank
3 = ART - Artillery
4 = AD - Air Defense
5 = AIR ASSOC'D UNITS

SUBTYPE CODES (NONUNIQUE)

FOR MR:

1 = WHEELED (BTR)
2 = TRACKED (BMP)
3 = LEAD WHEELED
4 = LEAD TRACKED
5 = VANGUARD WHEELED
6 = VANGUARD TRACKED
10 = HQ BMP
11 = HQ BTR
12 = MAIN CP
13 = READ CP

FOR TNK:

1 = ITB
2 = TANK REG.
3 = MRR
4 = VANGUARD
5 = LEAD TANK
10 = HDQTRS WITH MRR
11 = HDQTRS (TNK REG)
12 = MAIN CP
13 = REAR CP

FOR ART:

1 = SELF PROPELLED
2 = TOWED
10 = HEADQUARTERS

FOR AIR DEFENSE:

1 = SA-1
2 = SA-2
3 = SA-3
4 = SA-4
5 = SA-5
6 = SA-6
7 = SA-8
8 = SA-9
9 = EWC
10 = SA-6 CP
11 = SA-1 CP
12 = SA-4 CP
13 = SA-8 CP

FOR AIR ASSOC'D UNITS

1 = MILITARY AIRFIELD
2 = FORWARD AIR CONTROLLER
10 = AIR ARMY MCP
11 = AIR DIVISION MCP

Figure 4-9. TEMPRO Type and Subtype Designators

LEVEL CODES

1 = COMPANY
2 = BATTERY
3 = BATTALION
4 = REGIMENT
5 = BRIGADE
6 = DIVISION
7 = CORPS
8 = ARMY
9 = BATTALION HDQTRS
10 = REGIMENT HDQTRS
11 = BRIGADE HDQTRS
12 = DIVISION HDQTRS
13 = CORPS HDQTRS
14 = ARMY HDQTRS

Figure 4-10. TEMPRO Level Description

STATE CODES

1 = ASSEMBLED DISPERSED (STATIC 1)
2 = AT REST (STATIC 2)
3 = DEPLOYED FOR COMBAT (STATIC 3)
4 = MARCHING (DYNAMIC)
5 = STATIONARY (CAST IN CONCRETE; IMMOBILE)

Figure 4-11. TEMPRO Activity Designators

4.1.3 TCAC(D)

As this system analyses ELINT and COMINT reports only, the COMINT/ELINT correlation is the only cross-correlation activity. Sets of entities with common parameters (e.g., location and time) parameters are developed and displayed. Any additional correlation is done by the analyst based on these displays.

4.1.4 MAGIS

MAGIS searches only on location, displaying to the operator entities in his specified area of interest. All association and aggregation are done visually by the operator. The developers are currently considering the addition of a cross-correlation module.

4.1.5 References

The following documents contain information on the cross-correlation activities of the above systems.

BETA: *BETA Correlation Center Application Computer Program Configuration Item Development Specification* (No. SS42-43E, Part 1). Los Angeles: TRW, 1980 (uncl.)

BETA Correlation Center Application Computer Program Configuration Item Development Specification, Volume I and Appendix II [Correlation Processing CPC] (No. SS22-43, Part II). Los Angeles: TRW (uncl.).

MAGIS: System specification for the Computer System, Digital AN/TYQ-19(V), (Spec. No. SS700000B, FSCM 07609). 29 August 1978 (uncl.)

TCAC(D): Version 3 Subsystems Specifications (ADS SS), Volume 1, (SDRL Item HOOR-1). RCA, 15 December 1981 (conf.)

TEMPRO: TEMPRO Programmer and User Manual, (CDRL Item A0101), Redondo Beach: TRW, 1978 (uncl.).

Howell, D.H.: Final Report for Template-Assisted Intelligence Fusion Program (TEMPRO). Redondo Beach: TRW, 1979 (uncl.).

4.2 MATHEMATICS OF ALGORITHMS IN THESE SYSTEMS

As seen in the preceding section, current systems make automatic "parenting" decisions in cross-correlation based on three characteristics: location, time, and affiliation. The mathematics and its assumptions underlying these decision-making processes, for the most part, will be the familiar ones from self-correlation. Some of the points discussed in the report on self-correlation will be repeated here for completeness.

4.2.1 Chi-square and Normal Distributions

Location estimates received from sensors, which presumably took many measurements to get them, and from self-correlation, where reports were combined from many sensors, are assumed bivariate normally distributed with known variance. If used, the velocity $\underline{v}$ of a moving unit is also assumed to be normally distributed. For a potential child located at $\underline{x}$ and parent located at $\underline{u}$ (or at

$\underline{u}$ + *time** $\underline{v}$), *with* C *the inverse of the sum of their respective covariance* matrices (including velocity if appropriate), then

$$X = ([\underline{x} - \underline{u}])^T C[\underline{x} - \underline{u}]$$

(where T denotes the transpose of a matrix) is Chi-square distributed, and L = exp[-X/2] gives the complement of that distribution on the unit interval. L is the usual location measure used in self-correlation and sometimes used in cross-correlations. To apply these measures to determine if two entities are co-located (as is done in self-correlation) requires assuming that:

- Position (and velocity) measurements are independent.
- If velocity is explicitly used, position and velocity are independent.
- Different sensor systems do not introduce different biases (hence the chi-square non-centrality parameter is zero).
- The variance of the population from which the sensor system measurements are a sample is known for each sensor system.
- Distributions are invariant over time.

The application of these assumptions to self-correlation is discussed in the earlier report, and apply to cross-correlation as discussed in Section 3.3.

This location measure could be modified to take into account children falling within the doctrinal deployment radius of a parent unit by "moving out" the distribution and putting a large atom at the middle (e.g., using [$\underline{x}$ - $\underline{u}$ - r] for a doctrinal radius r). Then the test looks like one to determine if the entities are r units apart; that is, that one lies on a circle of radius r centered at the other. This score could be turned into a measure of whether the entities are at most r apart by, for example, setting L to one, whenever the distance between $\underline{x}$ and $\underline{u}$ is less than r, and doubling L otherwise. However, as the preceding survey of actual systems indicates, when doctrinal radii are introduced, so often are other measures for the location measure.

4.2.2 Uniform and Atomic, and Characteristic Distributions

Time decay and affiliation are usually represented by uniform or atomic distributions, or a combination of them. As was illustrated in Section 4.1.1 and Figure 4-4, the time-decay score is one, if the time interval [oldest siting, most recent siting] for the input/entity sufficiently overlaps that for the potential associate, be that associate a parent, offspring, or sibling. The score falls off linearly as the time intervals overlap less, or are farther apart, being zero either when they are disjoint or separated by a given distance. Affiliation can either be measured by a similar linear function, or by atoms, or by a characteristic function which is one if another child helps the parent match its doctrinal template, and zero if the parent already has enough children. Consistency tests fall into this latter group, affiliation "succeeding" if the parent-offspring or sibling relationship is permissible.

The only assumption underlying such use of the uniform distribution, beyond a qualitative conviction that it properly penalizes out-of-range time and affiliation values, is that uniform distributions do minimize worst case decision errors in the case of ignorance. In this respect such an approach is the farthest possible from Bayesian. In fact, it may even be said that ignorance is assumed just as strongly as independence. An advantage of uniform distributions is that they are easy to calculate, even when summing several.

4.2.3 Exponential Distributions

When linear cost is considered inappropriate. for example, for time and sometimes for location, one alternative is the exponential distribution exp[-ax] for the appropriate constant a. For time this may reflect some feeling about how long it will take a sensor system to find an entity, once that entity is in position, for the exponential represents "time to stopping" for independent trials. If used this way, the earlier comments on coverage envelopes for different systems are appropriate here, for a represents the average likelihood of success for each "trial," 1/a the mean time to detection. For location, read "distance" for "time," where a distance represents all points on the circle of radius d+r centered at one entity (usually the new or input report), where r is the template-defined radius.

To see what assumptions must be satisfied for exp[-ax] to be interpreted in this way, consider the Poisson process from which it can be derived. Let X(t) be the number of sitings of an entity by a sensor in time [0,t]. Then

$$\text{Prob}\ (X(t)=n) = [(at)^n/n!]\ \exp\ [-at].$$

Thus, in classical reliability theory, the reliability R(t) is

$$R(t) = \text{Prob}\ (X(t) = 0) = \exp\ [-at].$$

Both the above equations are given for a constant parameter a (not time varying a(t)). In the above case, a would equal 1/2. The assumptions for the Poisson process are:

- The number of sightings in non-overlapping time intervals is independent.
- The distribution of the number of sightings in any time interval depends only on the length of the interval.

This describes a point process with stationary, independent increments.

References:

Breiman, L. Probability. Reading, Massachusetts: Addison-Wesley, 1968.

Breipohl, A.M. Probabilistic Systems Analysis. New York: John Wiley and Sons, 1970.

4.2.4 Linear Discriminants

Decisions are based on a linear discriminant which is a weighted sum of location, time, and affiliation scores. The weights define how "important" each of these characteristics is to intelligence fusion. Current system implementations tend to weight them equally, but this is one of the few things that can be changed by the system manager, if conditions warrant. If all are uniformly distributed (and even L can be if $\underline{x}$ and $\underline{u}$ represent the same location), the distribution of the linear discriminant can be calculated. In cross-correlation, the standard derivation of a linear discriminant from normal distributions with equal covariance matrices does not seem applicable.

Regardless of the distribution of the individual scores, the linear JJscriminant will describe a hyperplane whose dimensionality is the number of independent variables in the weighted sum. Dependent parameters contribute to the same dimension, and the extent of contribution of each is determined by its weight. In self-correlation, such hyperplanes are used to distinguish between two (or many) populations of sitings. With the introduction in cross-correlation of template-defined constants (radius, minimum number of components, time decay constant, etc.), the "clusters" separated by the hyperplanes are more dispersed, their structure is more obscure, and the interpretation of a linear discriminant is quite murky.

4.2.5 Aggregation and Independence

The above methods usually assume independence of the underlying random variables. By the time a unit location, for example, is an average of many sightings by many sensor systems of the unit or its constituents, independence with a new siting may well be lost, especially if one sensor puts out two reports, one for the unit and one for the potential child. It would be very hard to establish independence between random variables representing characteristics of complex units identified by sitings of its constituents.

In the case illustrated by Figure 4-12, independence is violated because the same ELINT sensor information is being reported to the ASAC from two different sources. In one instance it is fused with COMINT data, in the other with imagery information. Such occurrences could arise in self-correlation, but more rarely. In cross-correlation, where fused data representing complex entities is being compared, it will happen quite naturally.

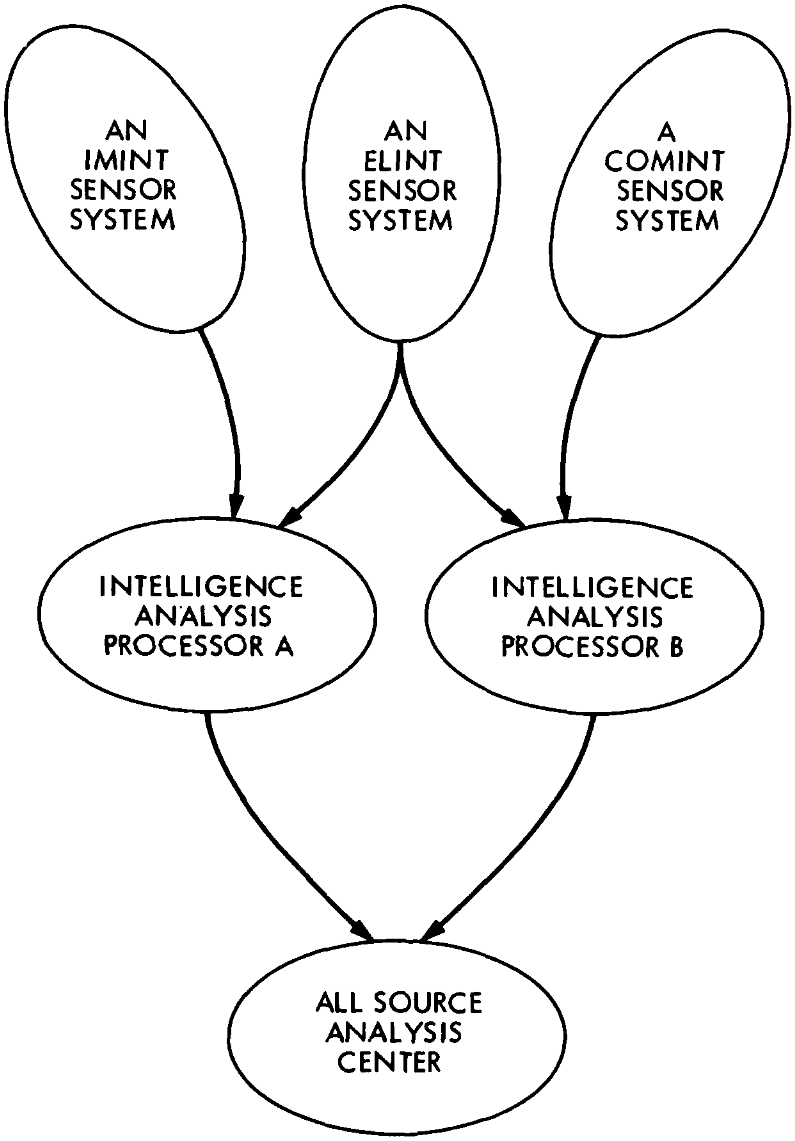

AN IMINT SENSOR SYSTEM
AN ELINT SENSOR SYSTEM
A COMINT SENSOR SYSTEM
INTELLIGENCE ANALYSIS PROCESSOR A
INTELLIGENCE ANALYSIS PROCESSOR B
ALL SOURCE ANALYSIS CENTER

Figure 4-12. A Dependent Data Path

4.3 SYSTEM ARCHITECTURE/ALGORITHM INTERFACES

This section focuses on the interface between algorithms and the system architecture. Since the code that embodies the algorithms constitutes only a few percent of the total system, these algorithms often have been written last, almost as an afterthought, with first consideration being given to major portions such as input conversion, database design, and displays. This tendency to adapt the algorithms to the system architecture is quite natural and reinforced by modern programming approaches such as top-down or outside-in. Since the choice of algorithms is often governed by outside factors, such as available input data and output information requirements, the algorithms often cannot be tailored to perform optimally within an already specified system structure. Therefore, when evaluating the performance of an algorithm in a system, the appropriateness of these interfaces must also be considered. This section will examine three major interface areas in light of the systems surveyed: message interfaces, database structure, and interaction with the operator (or, more likely, the lack thereof).

4.3.1 Message Interfaces

As seen from Section 3, the information required for the execution of an algorithm must match the information content of the messages input to the algorithms. This information in the messages from sensor systems external to the analysis system is largely beyond the control of the ASAS, but the information passed from one stage to the next within the system, i.e. from self-correlation to cross-correlation, is under its control (or would be if the system were designed as a whole rather than sequentially). Trimming information received in messages or as it crosses internal interfaces to reduce subsequent processing and storage volume is common. However, it is not always done so as to reflect appropriately the level of abstraction of information (the type of a radar in place of the parameter values used to identify the type). It is important to match the level of data abstraction and the type of algorithms used.

4.3.2 Database Structure

Considerations of cost and schedule usually argue strongly for the choice of an existing database processor. Again, the impact of this choice on the performance of the algorithms can be assessed. This section will focus on three major areas where the database management system significantly affects the algorithm performance.

4.3.2.1 Supporting Searches. The process of associating simple entities with established complex entities will require many database searches. The practical feasibility of the algorithms will depend on how efficiently certain types of searches are supported by the database system. Two primary types of access to the database that must be supported will be on the type of entity and by geographical location, and on the intersection of these keys. Exact locations of entities are not only not known precisely (they are described by an error ellipse), but also they are not important. Only the approximate distance from other nearby entities is important for purposes of cross-correlation. This suggests that entities might be indexed and retrieved by northing-easting blocks, and these blocks should be sufficiently large relative to the length of the error ellipses.

4.3.2.2 Linking Units. The database system must have the capability of linking or indexing the entity records that have been associated with a complex, and linking complexes that have been associated with higher order entities. The links should be traceable in both directions to aid in relinking as new reports are received, and delinking when units are reassigned or perceived to have changed their mission.

4.3.2.3 Storing Information. The primary purpose of an ASAS is to support decisions of the commander and his staff. The database system must be organized to provide decision information at a number of levels. On the lowest level, the cross-correlation algorithms organize and filter data for the analyst to aid him in making identification decisions. The database system also must provide a convenient structure to store the output of the algorithms so that it can be reviewed by the analyst with a minimum of time and effort. Furthermore, the database system must accept inputs from the analyst: either suggestions for an interactive algorithm to follow or results of analysis to be passed to another analyst looking at a "bigger-picture."

4.3.3 Interactive Control of the Analysis

The implementation of algorithms in the analysis system must be sufficiently flexible to allow the analyst to choose tools appropriate to the situation. The choice may reside with the individual analyst, with a system operations manager, or some combination of the two. Perceptions of the expected battlefield situation based on the best intelligence may be swept away by unpredicted occurrences. An extreme example of this is the recent experience of the British in the Falklands. Most systems surveyed, although they allow changing the weights used in the linear discriminants, do not allow any meaningful changes once the system is compiled.

Statistical techniques are fixed. No operator or system manager can choose between statistical and non-statistical algorithms. Yet density of reports (operational variables) and predominant types of reports affect the trade-off between statistical and non-statistical algorithms. When there are a large number of reports and the reports contain large error ellipses, the statistical algorithms are preferred, for example, for a large number of ELINT reports. When there are relatively few reports or they are very precise, the basis of the statistical methods does not apply, for example, with IMINT reports. Subject to workload constraints. even using more than one tool might be appropriate to give different insights. This is in the spirit of pattern recognition programs that apply a series of recognizers and compare the results on the basis of a figure of merit.

Furthermore. the operator often cannot even look at the results of these fixed tests individually; he can only see their combined result in the system simultaneous test. The choice between sequential and simultaneous tests depends on the quantity and quality of the report data (again operational variables) as well as on the nature of the algorithms. If the target characteristics are statistically independent or if they are linear (in the case of a nonstatistical test), sequential testing is mathematically correct. Where the parameters are partially correlated, but are assumed not to be, the results are unpredictable.

There are several advantages to the sequential approach. First, a large part of the tested entities can often be filtered out at an early state, thus reducing the computation. Second, the sequential testing is simpler to implement. Third, the analyst can see what part of the test failed rather than knowing only that it failed.

SECTION 5

OBSERVATIONS AND CONCLUSIONS

The first two conclusions apply generally to the cross-correlation process. The remaining observations concern specific aspects of that process, from template construction through database interfacing.

5.1 GENERAL

Cross-correlation in current systems is shaped by the decision tests that are merely variants of those used in self-correlation, being inherited or derived from them. Yet both the level of abstraction of the data analyses and the aim of cross-correlation are different from those of self-correlation. The basis of the mathematics of cross-correlation should be the aggregation of many simpler entities to identify more complex ones. The appropriate mathematical decision rules will then follow from those developed for aggregation and draw on a much wider range of mathematical disciplines.

However. no matter how carefully decision rules are constructed, as long as independence is destroyed, as in the example of Figure 4-12, there is little hope for good results.

General algorithms for doing the analysis and their interaction with the data retrieval system should be considered during the initial design of the system architecture. Not doing so can result in substantial costs. In particular, selection of hardware, which will constrain data retrieval, should not be made without reference to the impact on the database management/algorithms coupling.

5.2 SPECIFIC

Associating some unique piece of equipment, a combination of equipment types, or operational mode with a unit offers an excellent basis for constructing a template. Thus, properly observing and relaying in messages the characteristics of the equipment which facilitate its identification becomes increasingly important. But unlike doctrine, which tends to change very slowly, equipment is continually upgraded and modernized, requiring changes in both templates and correlation software. Flexibility should therefore be built into the system to accommodate these changes.

Much of the time and effort spent including situational templating in automatic analysis may be saved by choosing the proper mathematical tests instead of, for example, storing great quantities of terrain data.

Using doctrine-based equipment or unit counts to calculate affiliation scores ignores enemy losses and malfunctioning equipment, assuming he will always be at full doctrinal strength.

Current algorithms do not account for a unit's activity - even something as simple as whether or not the unit is moving at a specific time. Taking such information into account could significantly add to the power and flexibility of association tests.

The analyst should have the option of using the best mathematical decision tool, statistical or non-statistical, for the situation. In particular, the system should present the option of using either sequential or simultaneous analysis.

However, the best decision tests will be of no help to an analyst if the data base management system is faulty. For example, if the appropriate entity, template, etc. cannot be fetched, those decision tests are never performed. In some cases, processing may even be aborted. In all cases, information can be lost.

APPENDIX A

ALGORITHMS IN STANDARD FORM

```
TYPE
   position =
      RECORD
         east, north : real;
      END,
   matrix = array[1..3] of real;
   time =
      RECORD
         firstseen, lastseen : integer;
      END,
   rectangle =
      RECORD
         minimum, maximum : position;
      END;
   ellipse =
      RECORD
         semimajoraxis : real;
         semiminoraxis : real;
         orientationangl : real;
         focallength : real;
         focus1, focus2 : position;
      END;


   string2 = packed array[1..2] of char ;
   string6 = packed array[1..6] of char ;
   string30 = packed array[1..30] of char;
   index = array[1..12] of integer;
   workarea = array[1..4544] of integer;

VAR
 {V ACCASV : association measure data items}
   corrresult : integer;    {CORRLV : 1 = accept : updated
                                      2 = accept : no update
                                      3 = ambiguous
                                      4 = reject}
   assocmeasur : real;    {ASMEAV}
   bestmeasur : real;    {BESTMV}
   bestcandkey : integer;    {IBESCV}
   candambiguous : integer;    {NAMBCV}
   candprocessed : integer;    {NCANPV}
   timedecayscore : real;    {TIMSCV}
   locationscore : real;    {LOCSCV}
   inpmodify : integer;    {MODFLV}
   candaccepted . integer,    {NACEPV}
   adjustedmeasur . real;    {ADJAMV}

{B ACCBLD : variables for building the candidate list}
   candlist : array[1..6] of real;    {CLISTB}
   directpntr : array[1..6] of integer;    {DPNTRB}
   masterkey : array[1..6] of integer;    {MKEYB}
   modcount : array[1..6] of integer;    {MODCTB}
   cdbfilno . array[1..6] of integer;    {NOCDB}
   subjtyp : array[1..6] of integer;    {STYPEB}
   candindex : integer;    {INDEXB}

{Y ACCCAN : EDF fields retrieved from the candidate EDR}
   candmasterkey : integer;    {MASTRY}
   candgridno : integer;    {IGRIDY}
   candedr : position;    {IEASTY, NORTHY}
   candcovar : matrix,    {COVARY}
   candtime . time,    {ITIM1Y, ITIM2Y}
   candsgridno : integer;    {ISUBGY}
   compcount : integer;    {NCOMPY}
   candquantity : integer;    {IQUANY}
   candsubjtyp : integer;    {ISUBJY}
   candmodcount : integer,    {MODIFY}
   candcompentry : array[1..288] of integer;    {ENTRYC}

{C ACCCMP : data derived from the components of a candidate EDR.
         50 is the maximum number of components an EDR may have.
         The current number is XEDR}
   noofobjects : array[1..50] of integer;    {NOBJEC}
   compsubjtyps : array[1..50] of integer;    {STYPEC}

{Z ACCERR : status returns}
   controlstatus : integer;    {CONTRZ : 1 = successful
                                          2 = candidate retrieved
                                         -1 = failed : terminate processi
                                                       of this EDR
                                         -2 = failed : exit task
                                         -3 = no template retrieved
                                         -4 = no candidate found}
  {rsxstatus : integer;}    {RSXERZ}
   betastatus : integer;    {BETARZ}
   sarpstatus : integer;    {SARPRZ}

{X ACCINF : EDR fields retrieved from the input EDR}
   inpsubjtyp : integer;    {ISUBJX}
   inpedr : position;    {IEASTX, NORTHX}
   inpcovar : matrix;    {COVARX}
   inptime : time,    {ITIM1X, ITIM2X}
   inpquantity : integer;    {IQUANX}
   inpmodcount : integer;    {MODIFX}

{L ACCLOG : session log file event parameter}
   eventtype : integer;    {ITYPEL}

{N ACCLUN : logical unit numbers}
   corrtempfile : integer;    {CTFFLN}
   sarpfaplun : integer;    {FAPLUN}
   betaexeclun : integer;    {BEXLUN}
   cortemfillun : integer;    {CTFLUN}

{M ACCMSG : input data for EDR processing message}
   traceedrm : array[1..6] of integer;    {TRACEM}
   edrmmasterkey : integer;    {MASTRM}

{F ACCOCF : variables for opening and closing cdb}
   inppntr : integer;    {IPNTRF}
   candpntr : integer;    {CPNTRF}
   opencdbno : integer,    {IOPENF : 1 = radar file
                                     2 = radio file
                                     3 = mover file
                                     4 = shooter file
                                     5 = compound file
                                     6 = complex file}
   candcdbno : integer;    {ICANDF : as above}
   inpcdbno : integer.    {INPUTF : as above}

{R ACCRSA : variables for candidate search boundry}
   searchbound : rectangle;    {MINESR, MAXESR, MINNSR, MAXNSR}
   innerellipseflag: integer;    {INELLR : 0 = ellipse does not exist
                                           1 = ellipse exists}
   searchkey : integer;    {KEYSER}

{E ACCSEP : variables describing inner and outer ellipses}
   innersearch, outersearch : ellipse;    {EFOC1E, NFOC1E, EFOC2E, NFOC2E, A

{S ACCSFN : table of ASCII names and lengths of EDR fields}
  {fieldnames : array[1..6, 1..25] of integer;}    {FNAMES}
  {fieldlengths : array[1..25] of integer;}    {LENGTS}

{K ACCSKV : search boundries in grid/subgrid terms}
   mingrid : integer;    {MINGRK}
   maxgrid : integer;    {MAXGRK}
   minsubgrid : integer;    {MINSGK}
   maxsubgrid : integer;    {MAXSGK}

{G ACCSWA : SARP work area}
  {sarpworkarea : workarea;}    {SWA}
   actiontype : integer;    {ACTINQ : 0 = delete
                                      1 = add
                                      2 = update}
   tracenumber : array[1..6] of integer;    {TRACEQ}
   sepmasterkey : integer;    {MASTRQ}

{T ACCTMP : template entry record}
   tempsubj : integer;    {SUBJCT}
   affweight: real;    {AFFLWT}
   locweight : real;    {LOCAWT}
   timweight : real;    {TIMEWT}
   acceptthreshhold : real;    {ACCEPT}
   rejectthreshhold : real;    {REJECT}
   nocomptyp : integer;    {NCOMPT}
   compsubjtyp : array[1..16] of integer;    {CSUBJT}
   maxnocomp : array[1..16] of integer;    {NOMAXT}
   minnocomp : array[1..16] of integer;    {NOMINT}
   absmaxnocomp : array[1..16] of integer;    {ABMAXT}
   maxradius : array[1..16] of real;    {MAXRDT}
   minradius : array[1..16] of real;    {MINRDT}
   timeconst : array[1..16] of real;    {TIMECT}

{T ACCTMP : directory entry record}
   noparpntr : integer;    {NPARNT}
   parentpntr : array[1..58] of integer;    {PARENT}
   nosubjpntr : integer;    {NSUBJT}
   subjpntr : array[1..5] of integer;    {SUBPNT}
   noboolpntr : integer;    {NBOOLT}
   boolpntr : array[1..58] of integer;    {BOOLNT}

{P ACCTPD : template retrieval data}
   noparent : integer;    {NOTEMP}
   templatpntr : array[1..58] of integer;    {PRTEMP}
   counttemplates : integer;    {COUNTP}
   compindex : integer;    {INDEXP}



   tf : text;
   td : text;
   pf : text;
```

```
tdirname, tfilname, pfilname : string30;

candcompno : integer;

edrmaxnocomp : integer;    {XEDR}

N : integer;

stop, logged, leave : boolean;
```

```
PROGRAM ACPEPC(INPUT, OUTPUT, TD, TF, PF);
{Beta driver : determines the process flow of actions that attempt to correlat
 an input EDR with an existing EDR}

%INCLUDE 'common.pas'



PROCEDURE ACSESB; EXTERN;
PROCEDURE ACSEAM; EXTERN;
PROCEDURE ACSEFC; EXTERN;



PROCEDURE ACSELG;
   {prepares the data required for the event logging and then performs the
    logging}

BEGIN   {ACSELG}
END;   {ACSELG}



PROCEDURE ACSECL;
   {closes all files that were open and then exits the task}

BEGIN   {ACSECL}
   opencdbno := 0;
END;   {ACSECL}



PROCEDURE ACSEEX;
   {logs the abnormal termination of the cross correlation task and cleans
    up and exits the task}

BEGIN   {ACSEEX}
   IF eventtype <> 14   {cross correlation end}
      THEN
         BEGIN
            eventtype := 14;
            ACSELG   {perform logging}
         END;
   ACSECL   {clean up and exit}
END;   {ACSEEX}



PROCEDURE ACSERI;
   {retrieves the input EDR and extracts the data required for correlation}

VAR
   JJ : integer;
   continue : char;

BEGIN
```

```
      controlstatus := 1;    {successful}
      corrresult := 4;    {reject}
      assocmeasur := 0;
      bestmeasur := 0;
      bestcandkey := 0;
      candprocessed := 0;
      candambiguous := 0;
      candaccepted := 0;
      FOR JJ := 1 to 6 DO
         BEGIN
            candlist[JJ] := 0;
            subjtyp[JJ] := 0;
            cdbfilno[JJ] := 0;
            modcount[JJ] := 0;
            masterkey[JJ] := 0;
            directpntr[JJ] := 0
         END;
      IF inpmodify = 0 THEN
         BEGIN
            writeln('Input EDR? (y or n)');
            readln(continue);
            IF continue = 'n'
               THEN controlstatus := -1   {failed : terminate processing of this ED
               ELSE
                  BEGIN
                     writeln('Enter subject type, quantity, max no of components');
                     readln(inpsubjtyp, inpquantity, edrmaxnocomp);
                     writeln('Enter location (east, north)');
                     WITH inpedr DO
                       readln(east, north);
                     writeln('Enter covariance vector');
                     readln(inpcovar[1], inpcovar[2], inpcovar[3]);
                     writeln('Enter time firstseen, lastseen');
                     WITH inptime DO
                        readln(firstseen, lastseen)
                  END
         END
   END;   {ACSERI}


   PROCEDURE ACSETP;
      {retrieves from the correlation template file the directory entry that
       contains the ponters to all templates}

   VAR
      recordno : integer;    {RECRD}
      nocomp, nextp, ip : integer;
      nosubj : array[1..8] of integer;
      H, I, J, K : integer;

   BEGIN
      reset(td);
      reset(tf);
      recordno := inpsubjtyp;
      FOR H := 1 to (recordno - 1) DO
         readln(td),
      read(td, noparent);
      FOR I := 1 to noparent DO
         read(td, nosubj[I]);
   nextp := 1;
   ip := 1;
   WHILE NOT eof(tf) DO
      BEGIN
         readln(tf, nocomp, tempsubj);
         FOR J := 1 to noparent DO
            IF tempsubj = nosubj[I]
               THEN
                  BEGIN
                     templatpntr[ip] := nextp;
                     ip := ip + 1
                  END;
         FOR K := 1 to nocomp DO
            readln(tf);
         nextp := nextp + nocomp + 1
      END
END;  {ACSETP}



PROCEDURE ACSEPT;
   {uses a template pointer to retrieve a template from the correlations
    template file}

VAR
   recordno : integer;    {RECRD}
   I, J : integer;

BEGIN   {ACSEPT}
   reset(tf);
   controlstatus := 1;
   IF counttemplates <= noparent
      THEN
         BEGIN
            recordno := templatpntr[counttemplates];
            FOR I := 1 to (recordno - 1) DO
               readln(tf);
            readln(tf,nocomptyp,tempsubj,locweight,timweight,affweight,
                      acceptthreshhold,rejectthreshhold);
            compindex := 0;
            FOR J := 1 to nocomptyp DO
               BEGIN
                  readln(tf, compsubjtyp[J], maxnocomp[J], minnocomp[J],
                             absmaxnocomp[J], maxradius[J], minradius[J],
                             timeconst[J]);
                  IF compsubjtyp[J] = inpsubjtyp
                     THEN compindex := J
               END;
            IF compindex = 0
               THEN controlstatus := -3   {no template retrieved}
         END
      ELSE controlstatus := -3   {no template retrieved}
END;   {ACSEPT}



PROCEDURE ACSEDC;
   {opens the correlated data file (either compound or complex) that contains
    the EDR type specified by the template}
```

```
BEGIN   {ACSEDC}
END;    {ACSEDC}



PROCEDURE ACSEUP;
   {implements the modifications to the cdb when an acceptable cross
    correlation has been identified}

BEGIN   {ACSEUP}
END;    {ACSEUP}



BEGIN   {ACPEPC}
   {ACSEIN, initialize cross correlation edr processing task}
      tfilname := 'drc0:[jwg.cross]templat.dat    ';
      tdirname := 'drc0:[jwg.cross]tempdir.dat    ';
      pfilname := 'drc0:[jwg.cross]parent.dat     ';
   RESET(tf);
   stop := false;
   REPEAT   {input queue loop}
      WRITELN (' ENTER BETA STATUS (message queued)');
      READLN (betastatus);
      IF betastatus <> -13
         THEN
            BEGIN
               IF betastatus <= 0
                  THEN stop := true
                  ELSE
                     BEGIN
                        eventtype := 13;   {cross correlation begin}
                        ACSELO;   {perform logging}
                        inpmodify := 0;
                        leave := false;
                        REPEAT
                           ACSERI;   {retrieve input EDRand extract data}
                           logged := false;
                           IF controlstatus = -1   {failed : terminate processing of this EDR}
                              THEN
                                 BEGIN
                                    eventtype := 14;
                                    ACSELO;   {perform logging}
                                    leave := true;
                                    logged := true
                                 END
                              ELSE
                                 BEGIN
                                    ACSETP;   {get template pointers}
                                    counttemplates := 0;
                                    FOR N := 1 to noparpntr DO   {template processing loop}
                                       BEGIN
                                          counttemplates := counttemplates + 1;
                                          ACSEPT;   {retrieve template}
                                          IF controlstatus = -3   {no template retrieved}
                                             THEN N := noparpntr
                                             ELSE
                                                BEGIN
```

```
                                                   ACSESB;   {calculate search boundries}
                                                   IF controlstatus < 0
                                                      THEN N := noparpntr
                                                      ELSE
                                                         BEGIN
                                                            ACSEOC;   {open appropriate correlated data file}
                                                            IF controlstatus <> -1  {no file open error}
                                                               THEN
                                                                  REPEAT   {candidate retrieval loop}
                                                                     ACSEFC;   {find a qualified candidate}
                                                                     IF (controlstatus = 2) AND (controlstatus <> -4) THEN ACSEAM
                                                                  UNTIL controlstatus = 4  {no candidate found}
                                                         END
                                                END
                                       END;
                                    IF (candambiguous > 0) OR (candaccepted > 0)
                                       THEN
                                          BEGIN
                                             ACSEUP;   {update CDB}
                                             IF controlstatus = 6   {input modified}
                                                THEN inpmodify := inpmodify + 1
                                                ELSE leave := true
                                          END
                                       ELSE leave := true
                                 END
                        UNTIL (inpmodify = 2) OR (leave);
                        IF NOT logged
                           THEN
                              BEGIN
                                 eventtype := 14;
                                 ACSELG   {perform logging}
                              END
                     END
            END
   UNTIL stop;
   ACSECL   {clean up and exit}
END   {ACPEPC PROGRAM}
```

```
MODULE ESB(INPUT,OUTPUT);
{This module determines the boundries of the retrieval area where all possible
 parent candidates may be located}

%INCLUDE 'common.pas'

PROCEDURE ACSESB;    {Main routine of this module}

CONST
   pi = 3.14159265358;

VAR
   error : ellipse;
   covarstatus : integer;    {CSTAT}
   radian : real;    {DARAD}
   deltae : real;
   deltan : real;



FUNCTION MIN(x,y:real):real;
BEGIN
   MIN := y;
   IF x < y THEN MIN := x;
END;



FUNCTION MAX(x,y:real):real;
BEGIN
   MAX := y;
   IF x > y THEN MAX := x;
END;



PROCEDURE ACSEEP(VAR cov : matrix; VAR error : ellipse;
                 VAR covarstatus : integer);
   {Convert covariance matrix to confidence ellipse}

VAR
   temp, templ, a, b, theta : real;
   sinalpha, cosalpha, alpha : real;
   sinbeta, cosbeta, beta : real;

BEGIN
   IF (cov[1] > 0) AND (cov[3] > 0) AND (cov[1]*cov[3] - sqr(cov[2]) > 0)   {error ellipse exists}
      THEN
         BEGIN
            covarstatus := 0;   {success}
            temp := SQRT(SQR(cov[1] - cov[3]) + 4.0*SQR(cov[2]));
            a := SQRT(2.9955*(cov[1]+cov[3] + temp));
            b := SQRT(2.9955*(cov[1]+cov[3] - temp));
            IF (a-b < 1.0E-6)
               THEN theta := 0
               ELSE
                  BEGIN
                     temp := SQR(a) - SQR(b);
                     temp1 := MAX(5.991*(cov[3] - cov[1])/temp, -1.0);
                     cosalpha := MIN(temp1, 1.0);
                     sinalpha := SQRT(1.0 - SQR(cosalpha));
                     alpha := ARCTAN(sinalpha/cosalpha);
                     temp1 := MAX(11.982*cov[2]/temp, -1.0);
                     sinbeta := MIN(temp1, 1.0);
                     cosbeta := SQRT(1.0 - SQR(sinbeta));
                     beta := ARCTAN(sinbeta/cosbeta);
                     IF (beta > 0)
                        THEN theta := 28.64789*alpha
                        ELSE theta := -28.64789*alpha
                  END;
               WITH error DO
                  BEGIN
                     semimajoraxis := a;
                     semiminoraxis := b;
                     orientationangl := theta;
                  END
      END
  ELSE covarstatus := 1   {failure}
END;   {ACSEEP}



PROCEDURE ADSCNO(VAR xloc : position; VAR grid, subgrid : integer);

BEGIN   {ADSCNO}
END;   {ADSCNO}



PROCEDURE ACSESK;
   {builds the search key used to retrieve candidate EDRs}

BEGIN   {ACSESK}
   IF sarpstatus <> 1   {successful}
      THEN ACSEEX   {abnormal termination}
END;   {ACSESK}



BEGIN   {ACSESB}
   controlstatus := 1;   {successful}
   innerellipseflag := 0;
   ACSEEP(inpcovar,error,covarstatus);   {convert covariance matrix to confidence ellipse}
   WRITELN(' ERROR ELLIPSE PARAMETERS:');
   WITH error DO
      BEGIN
         WRITELN(' semimajoraxis =',semimajoraxis);
         WRITELN(' semiminoraxis =',semiminoraxis);
         WRITELN(' orientation angle =',orientationangl)
      END;
   IF covarstatus <> 0   {successful}
      THEN controlstatus := -1   {failed : terminate processing of this EDR}
      ELSE
         BEGIN
            WRITELN( 'outer search ellipse exists');
            WITH outersearch DO   { compute outersearch ellipse parameters}
```

```
               BEGIN
                  semimajoraxis := maxradius[compindex] + error.semimajoraxis;
                  semiminoraxis := maxradius[compindex] + error.semiminoraxis;
                  focallength := SQRT(SQR(semimajoraxis) - SQR(semiminoraxis));
                  radian := error.orientationangl*pi/180;
                  focus1.east := inpedr.east + focallength*SIN(radian);
                  focus1.north := inpedr.north + focallength*COS(radian);
                  focus2.east := inpedr.east - focallength*SIN(radian);
                  focus2.north := inpedr.north - focallength*COS(radian);
               END;   {WITH outersearch}
            IF (minradius[compindex] > error.semimajoraxis)
               THEN
                  BEGIN
                     innerellipseflag := 1;   {innerellipse exists}
                     WRITELN(' inner search ellipse exists');
                     WITH innersearch DO   {compute innersearch ellipse parameters}
                        BEGIN
                           semimajoraxis := minradius[compindex] - error.semiminoraxis;
                           semiminoraxis := minradius[compindex] - error.semimajoraxis;
                           orientationangl := error.orientationangl + 90;
                           focallength := SQRT(SQR(semimajoraxis) - SQR(semiminoraxis));
                           radian := orientationangl*pi/180;
                           focus1.east := inpedr.east + focallength*SIN(radian);
                           focus1.north := inpedr.north + focallength*COS(radian);
                           focus2.east := inpedr.east - focallength*SIN(radian);
                           focus2.north := inpedr.north - focallength*COS(radian);
                        END   {WITH innersearch}
                  END;
               WITH outersearch DO   {compute outer search ellipse rectangle}
                  BEGIN
                     deltae := (semimajoraxis*ABS(SIN(radian)) + semiminoraxis*ABS(COS(radian)));
                     deltan := (semimajoraxis*ABS(COS(radian)) + semiminoraxis*ABS(SIN(radian)));
               END;   {WITH outersearch}
            WRITELN(' SEARCH BOUND LIMITS ');
            WITH searchbound DO
               BEGIN
                  minimum.east := inpedr.east - deltae;
                  maximum.east := inpedr.east + deltae;
                  minimum.north := inpedr.north - deltan;
                  maximum.north := inpedr.north + deltan;
                  WITH minimum DO WRITELN(' min(east,north) ',east,north);
                  WITH maximum DO WRITELN(' max(east,north) ',east,north);
                  ADSCNG(minimum,mingrid,minsubgrid);   {convert min to min grid and subgrid}
                  minsubgrid := 0;
                  ADSCNG(maximum,maxgrid,maxsubgrid);   {convert max to min grid and subgrid}
               END;   {WITH searchbound}
            ACSESK;   {build search key}
         END
END   {ACSESB}
END.   {of MODULE ACSESB}
```

```
MODULE EFC(INPUT, OUTPUT);
{This module uses the search key to retrieve a candidate EDR index
 record.  The candidate EDR location is extracted and examined to
 determine whether or not the candidate lies within the search
 boundry.  If the candidate is within the boundry, the entire
 candidate EDR is retrieved.}

%include 'common.pas'

PROCEDURE ACSEFC;    {main PROCEDURE}

VAR
   gridlength : integer;    {LENGD}
   locflag : integer;    {LFLAG}
   candsubgrid : integer;    {STSGD}
   candsubj : integer;    {SUBCAN}
   candswapgridno : integer;    {STGRD}

   efcexit : boolean;
   cycle : boolean;




PROCEDURE ADSCGN(VAR grid : integer; VAR subgrid : integer;
                 VAR edr : position);
   {Convert grid/subgrid to easting/northing}

BEGIN   {ADSCGN}
END;   {ADSCGN}




PROCEDURE ACSELO( VAR searchellipse : ellipse; VAR locflag : integer);
   {Determines whether an EDR lies inside or outside the search annulus.
    The output variable locflag is set equal to 1 if the EDR is inside
    and equal to 2 if the EDR is outside the search ellipse.}

VAR
   distfoc1tocand : real;    {DIST1}
   distfoc2tocand : real;    {DIST2}
   sumdistance : real;    {SUMDIS}
   a, b, x, y : real;

BEGIN
   locflag := 0;
   WITH searchellipse DO
         BEGIN
            a := candedr.east - focus1.east;
            b := candedr.north - focus1.north;
            distfoc1tocand := SQRT(SQR(a) + SQR(b));
            x := candedr.east - focus2.east;
            y := candedr.north - focus2.north;
            distfoc2tocand := SQRT(SQR(x) + SQR(y));
            sumdistance := distfoc1tocand + distfoc2tocand;
            IF sumdistance <= 2*semimajoraxis
               THEN locflag := 1   {candidate EDR is inside ellipse}
               ELSE locflag := 2   {candidate EDR is outside ellipse}
         END
END;   {ACSELO}
```

```
BEGIN   {ACSEFC}
   WRITELN(' ENTER ACSEFC MODULE ');
   controlstatus := 0;
   efcexit := false;
   reset(pf);
REPEAT
   cycle := false;
   readln(pf, candsubjtyp, candcompno, candgridno, candsgridno,
              candedr.east, candedr.north, candtime.firstseen,
              candtime.lastseen, candquantity, maxgrid, mingrid,
              candcovar[1], candcovar[2], candcovar[3]);
   IF (sarpstatus = -10) OR (sarpstatus = -107)   {end of records or no records found}
      THEN
         BEGIN
           controlstatus := -4;   {no candidate found}
           efcexit := true
         END
      ELSE
         IF sarpstatus <> 1   {successful}
            THEN ACSEEX   {log abnormal termination}
            ELSE
               IF candsubjtyp > tempsubj
                  THEN
                     BEGIN
                        controlstatus := -4;   {no candidate found}
                        efcexit := true
                     END;
   IF NOT efcexit
      THEN
         BEGIN
            IF sarpstatus <> 1 THEN ACSEEX;   {log abnormal termination}
            IF candgridno >= maxgrid
               THEN
                  BEGIN
                     controlstatus := -4;   {no candidate found}
                     efcexit := true
                  END
               ELSE
                  BEGIN
                     ADSCGN(candgridno,candsgridno,candedr);   {convert grid/subgrid to easting/northing}
                     IF ( candedr.north < searchbound.maximum.north) AND
                        ( candedr.north > searchbound.minimum.north)
                        THEN
                           BEGIN
                              ACSELO(innersearch, locflag);   {determine candidate EDR location}
                              IF locflag = 2
                                 THEN
                                    BEGIN
                                       WRITELN(' CAND OUTSIDE ELLIPSE');
                                       cycle := true
                                    END
                                 ELSE
                                    BEGIN
                                       IF innerellipseflag = 1   {ellipse exists}
                                          THEN
                                             BEGIN
                                                ACSELO(outersearch, locflag);   {determine candidate EDR location}
                                                IF locflag = 1
                                                   THEN
                                                      BEGIN
                                                         WRITELN(' CAND INSIDE ELLIPSE');
                                                         cycle := true
                                                      END
                                             END
                                       IF NOT cycle
                                          THEN
                                             IF controlstatus <> -1   {failed : terminate processing of this EDR}
                                                THEN
                                                   BEGIN
                                                      controlstatus := 2;   {candidate EDR retrieved}
                                                      efcexit := true
                                                   END
                                    END
                        END
                  END
         END
JNTIL efcexit
END  {of MODULE ACSEFC}
```

```
MODULE EAM(INPUT, OUTPUT);
{This module calculates the association measure for a given subject EDR/
 candidate EDR pair and determines the cross correlation results}



%include 'common.pas'



PROCEDURE ACSEAM;
{Main routine of this module}

VAR
   affiliatscore : real;    {AFFSC}
   nocomp : integer;    {ICCNT}
   inpabsmaxnoobj : integer;    {IABMX}
   newcompcount : integer;    {NCCNT}
   newmaxnocomp : integer;    {MAXNO}
   newminnocomp : integer;    {MINNO}
   i : integer;



PROCEDURE ACSELS;
   {computes the figure of merit score based on the relative locations of the
    input and candidate EDRs and the template specified spatial relationship}

VAR
   maxrd : real;    {RDTMX}
   minrd : real;    {RDTMN}
   distance : real;    {DLENG}
   normaldistanc : real;    {DNORM}
   temp : real;    {SMALD}
   determinent : real;    {DETR}
   i : integer;
   tempadd : matrix;    {ADD11, ADD12, ADD22}
   differnce : position;    {IEAST, INORTH}

BEGIN
   maxrd := maxradius[compindex];
   minrd := minradius[compindex];
   differnce.east := inpedr.east - candedr.east;
   differnce.north := inpedr.north - candedr.north;
   IF maxrd <> 0
      THEN
         BEGIN
            distance := SQRT(SQR(differnce.east) + SQR(differnce.north));
            IF distance > maxrd
               THEN
                  BEGIN
                     temp := distance - maxrd;
                     locationscore := EXP(-temp/2)
                  END
               ELSE
                  IF distance >= minrd
                     THEN locationscore := 1
                     ELSE
                        BEGIN
                           temp := minrd - distance;
                           locationscore := EXP(-temp/2)
                        END
            END
         ELSE
            BEGIN
               FOR i := 1 to 3 DO
                  tempadd[i] := inpcovar[i] + candcovar[i];
               determinent := tempadd[1]*tempadd[3] - SQR(tempadd[2]);
               WITH differnce DO
                  BEGIN
                     normaldistanc := (tempadd[3]*SQR(east) - 2*east*north*tempadd[2] +
                                       tempadd[1]*SQR(north))/determinent;
                     locationscore := EXP(-normaldistanc/2);
{Error in original software!!!! east and north are not defined.}
                  END
            END
END;   {ACSELS}



PROCEDURE ACSETS;
   {computes a figure of merit time score based on the relative times of
    observation of the input and candidate EDR}

VAR
   timec : real;    {TIMEC}
   deltime : real;    {DELTA}

BEGIN   {ACSETS}
   timec := timeconst[compindex];
   timedecayscore := 0;
   IF candtime.lastseen < inptime.firstseen
      THEN
         BEGIN
            deltime := inptime.firstseen - candtime.lastseen;
            IF deltime > timec
               THEN timedecayscore := 0
               ELSE timedecayscore := 1 - deltime/timec
         END
      ELSE
         IF inptime.lastseen < candtime.firstseen
            THEN
               BEGIN
                  deltime := candtime.firstseen - inptime.lastseen;
                  IF deltime > timec
                     THEN timedecayscore := 0
                     ELSE timedecayscore := 1 - deltime/timec
               END
            ELSE   {cand EDR and input EDR times overlap}
               timedecayscore := 1
END;   {ACSETS}
```

```
    or accepted.   The criteria are the association measure and the acceptance
    and rejection threshholds}

VAR
   rsave : real;
   i, isave, iisave : integer;

BEGIN   {ACSERA}
   IF assocmeasur >= acceptthreshhold
      THEN candaccepted := candaccepted + 1
      ELSE
         IF assocmeasur <= rejectthreshhold
            THEN corrresult := 4   {candidate rejected}
            ELSE
               BEGIN
                  candambiguous := candambiguous + 1;
                  IF assocmeasur > bestmeasur
                     THEN
                        BEGIN
                           bestmeasur := assocmeasur;
                           bestcandkey := inppntr
                        END;
                  adjustedmeasur := assocmeasur / acceptthreshhold;
                  candlist[1] := adjustedmeasur;
                  directpntr[1] := candpntr;
                  modcount[1] := candmodcount;
                  cdbfilno[1] := candcdbno;
                  subjtyp[1] := candsubjtyp;
                  FOR i := 2 to 6 DO
                     BEGIN
                        IF candlist[i - 1] > candlist[i]
                           THEN
                              BEGIN
                                 rsave := candlist[i-1];
                                 candlist[i-1] := candlist[i];
                                 candlist[i] := rsave;
                                 isave := directpntr[i-1];
                                 directpntr[i-1] := directpntr[i];
                                 directpntr[i] := isave;
                                 iisave := modcount[i-1];
                                 modcount[i-1] := modcount[i];
                                 modcount[i] := iisave;
                                 iisave := cdbfilno[i-1];
                                 cdbfilno[i-1] := cdbfilno[i];
                                 cdbfilno[i] := iisave;
                                 isave := subjtyp[i-1];
                                 subjtyp[i-1] := subjtyp[i];
                                 subjtyp[i] := isave
                              END
                     END
               END;
   candprocessed := candprocessed + 1
END;   {ACSERA}



BEGIN   {ACSEAM}
   assocmeasur := 0;
   affiliatscore := 0;
```

```
    locationscore := 0;
    timedecayscore := 0;
    nocomp := 0;
    IF compcount < edrmaxnocomp
       THEN
          BEGIN
             inpabsmaxnoobj := absmaxnocomp[compindex]*candquantity;
             FOR i := 1 to compcount DO
                IF compsubjtyp[i] = inpsubjtyp
                   THEN nocomp := nocomp + noofobjects[i];
             newcompcount := nocomp + inpquantity;
             IF newcompcount <= inpabsmaxnoobj
                THEN
                   BEGIN
                      IF candcdbno = 6
                         THEN
                            BEGIN
                               newminnocomp := minnocomp[compindex]*candquantity;
                               newmaxnocomp := maxnocomp[compindex]*candquantity;
                               IF newcompcount <= newminnocomp
                                  THEN affiliatscore := 1
                                  ELSE
                                     IF newcompcount >= newmaxnocomp
                                        THEN affiliatscore := 0
                                        ELSE affiliatscore := 1 - (newcompcount - newminnocomp)/
                                                              (1 + newmaxnocomp - newminnocomp)
                            END;
                      ACSELS;        {compute location score}
                      ACSETS;        {compute time score}
                      assocmeasur := affiliatscore * affweight
                                     + locationscore * locweight
                                     + timedecayscore * timweight;
                   END
          END;
    ACSERA;
 END   {ACSEAM}
 END   {of MODULE ACSEAM}
```

APPENDIX B

DATA BASE REPORTS

This appendix contains repots from the USAMS database on BETA and TEMPRO cross-correlation algorithms. Following the list of algorithms in each section is a utilized (structure) report and dictionary report.

```
                                        U N C L A S S I F I E D
                                        JTFTB-A                                              15 MAR 84
                                   Joint Tactical Fusion Test Bed - Army
                                        Name Selection
    1     BE_ACPEPC                      PROCESS
    2     BE_ACSEAM                      PROCESS
    3     BE_ACSECF                      PROCESS
    4     BE_ACSEFC                      PROCESS
    5     BE_ACSELI                      PROCESS
    6     BE_ACSEPT                      PROCESS
    7     BE_ACSERI                      PROCESS
    8     BE_ACSESB                      PROCESS
    9     BE_ACSETP                      PROCESS
   10     BE_ACSEUF                      PROCESS
```

```
REPORT 190                                                                                  PAGE 001
                                        U N C L A S S I F I E D
```

```
Utilizes Structure

 COUNT LEVEL          NAME

   1   1.0      BE_ACPEPC
   2   1.1         BE_ACSFRI
   3   1.1.1          BE_ACSESB
   4   1.1.1.1           BE_ACSESK
   5   1.1.2          BE_ACSEFI
   6   1.1.3          BE_ACSETP
   7   1.1.4          BE_ACSEEF
   8   1.1.5          BE_ACSEFC
   9   1.1.5.1           BE_ACSELO
  10   1.1.5.2           BE_ACSERC
  11   1.1.5.3           BE_ACSEEC
  12   1.1.6          BE_ACSEAM
  13   1.1.6.1           BE_ACSELS
  14   1.1.6.2           BE_ACSETS
  15   1.1.6.3           BE_ACSERA
  16   1.1.7          BE_ACSEUP
  17   1.1.7.1           BE_ACSELI
  18   1.1.7.1.1            BE_ACSEEF
  19   1.1.7.1.2            BE_ACSEEC
  20   1.1.7.1.3            BE_ACSULK
  21   1.1.7.2           BE_ACSECF
  22   1.1.7.2.1            BE_ACSEGG
  23   1.1.7.2.2            BE_ACSECC
  24   1.1.7.2.3            BE_ACSECF   *
  25   1.1.7.2.4            BE_ACSEEC
  26   1.2         BE_ACSETP
  27   1.3         BE_ACSEPT
  28   1.4         BE_ACSESB
  29   1.4.1          BE_ACSESK
  30   1.5         BE_ACSEAM
  31   1.5.1          BE_ACSELS
  32   1.5.2          BE_ACSETS
  33   1.5.3          BE_ACSERA
  34   1.6         BE_ACSEUP
  35   1.6.1          BE_ACSELI
  36   1.6.1.1           BE_ACSEEF
  37   1.6.1.2           BE_ACSEEC
  38   1.6.1.3           BE_ACSULK
  39   1.6.2          BE_ACSECF
  40   1.6.2.1           BE_ACSEGG
  41   1.6.2.2           BE_ACSECC
  42   1.6.2.3           BE_ACSECF   *
  43   1.6.2.4           BE_ACSEEC
  44   1.7         BE_ACSEFC
  45   1.7.1          BE_ACSELO
  46   1.7.2          BE_ACSLRC
  47   1.7.3          BE_ACSEEC
```

```
COUNT LEVEL          NAME

  48  2.0     BE_ACSEAM
  49  2.1        BE_ACSELS
  50  2.2        BE_ACSETS
  51  2.3        BE_ACSERA
  52  3.0     BE_ACSECF
  53  3.1        BE_ACSEGG
  54  3.2        BE_ACSECC
  55  3.3        BE_ACSECF
  56  3.3.1         BE_ACSEGG
  57  3.3.2         BE_ACSECC
  58  3.3.3         BE_ACSECF  *
  59  3.3.4         BE_ACSEEC
  60  3.4        BE_ACSEEC
  61  4.0     BE_ACSEFC
  62  4.1        BE_ACSELO
  63  4.2        BE_ACSERC
  64  4.3        BE_ACSEEC
  65  5.0     BE_ACSELI
  66  5.1        BE_ACSEEF
  67  5.2        BE_ACSEEC
  68  5.3        BE_ACSULK
  69  6.0     BE_ACSEPT
PSA461: "BE_ACSEPT" has no SUBPARTS and does not UTILIZE any PROCESS.
  70  7.0     BE_ACSERI
  71  7.1        BE_ACSESB
  72  7.1.1         BE_ACSESK
  73  7.2        BE_ACSEPT
  74  7.3        BE_ACSETP
  75  7.4        BE_ACSEEF
  76  7.5        BE_ACSEFC
  77  7.5.1         BE_ACSELO
  78  7.5.2         BE_ACSERC
  79  7.5.3         BE_ACSEEC
  80  7.6        BE_ACSEAM
  81  7.6.1         BE_ACSELS
  82  7.6.2         BE_ACSETS
  83  7.6.3         BE_ACSERA
  84  7.7        BE_ACSEUP
  85  7.7.1         BE_ACSELI
  86  7.7.1.1          BE_ACSEEF
  87  7.7.1.2          BE_ACSEEC
  88  7.7.1.3          BE_ACSULK
  89  7.7.2         BE_ACSECF
  90  7.7.2.1          BE_ACSEGG
  91  7.7.2.2          BE_ACSECC
  92  7.7.2.3          BE_ACSECF  *
  93  7.7.2.4          BE_ACSEEC
  94  8.0     BE_ACSESB
  95  8.1        BE_ACSESK
```

```
COUNT LEVEL          NAME

 96  9.0    BE_ACSETP
FSA461 "BE_ACSETP" has no SUBPARTS and does not UTILIZE any PROCESS.
 97  10.0   BE_ACSEUP
 98  10.1      BE_ACSELI
 99  10.1.1       BE_ACSEEF
100  10.1 2       BE_ACSEEC
101  10 1.3       BE_ACSULK
102  10.2      BE_ACSECF
103  10.2.1       BE_ACSEQQ
104  10 2.2       BE_ACSECC
105  10.2.3       BE_ACSECF  *
106  10.2.4       BE_ACSEEC

* Loop encountered (a PROCESS is directly or indirectly UTILIZED by itself)
```

Utilizes Matrix

Explanation of the Utilizes Matrix:

The rows are input PROCESS names, and the columns are PROCESSES UTILIZED by (or a SUBPART of) the rows.

```
     (i, j) value   meaning
     ------------   ----------------------------------

          U         Column j is UTILIZED by Row i
          S         Column j is a PART of Row i
          B         Column j is both UTILIZED by, and a PART of Row i
```

Utilizes Matrix

```
                                           20 BE_ACSECC -------------------- /
                                         19 BE_ACSEGG -------------------- / |
                                       18 BE_ACSULK -------------------- / | |
                                     17 BE_ACSECF -------------------- / | | |
                                   16 BE_ACSELI -------------------- / | | | |
                                                                     | | | | |
                                15 BE_ACSERA -------------------- /  | | | | |
                              14 BE_ACSETS -------------------- / |  | | | | |
                            13 BE_ACSELS -------------------- / | |  | | | | |
                          12 BE_ACSEEC -------------------- / | | |  | | | | |
                        11 BE_ACSERC -------------------- / | | | |  | | | | |
                                                          | | | | |  | | | | |
                     10 BE_ACSELO -------------------- /  | | | | |  | | | | |
                    9 BE_ACSESK -------------------- / |  | | | | |  | | | | |
                  8 BE_ACSEEF -------------------- / | |  | | | | |  | | | | |
                7 BE_ACSEFC -------------------- / | | |  | | | | |  | | | | |
              6 BE_ACSEUP -------------------- / | | | |  | | | | |  | | | | |
                                               | | | | |  | | | | |  | | | | |
           5 BE_ACSEAM -------------------- /  | | | | |  | | | | |  | | | | |
         4 BE_ACSESB -------------------- / |  | | | | |  | | | | |  | | | | |
       3 BE_ACSEPT -------------------- / | |  | | | | |  | | | | |  | | | | |
     2 BE_ACSETP -------------------- / | | |  | | | | |  | | | | |  | | | | |
   1 BE_ACSERI -------------------- / | | | |  | | | | |  | | | | |  | | | | |
                                    | | | | |  | | | | |  | | | | |  | | | | |
----------------------------------+----------+----------+----------+----------+
1 BE_ACPEPC --------------------- | U U U U U| U U      |          |          |
2 BE_ACSERI --------------------- |   U U U U| U U U    |          |          |
3 BE_ACSESB --------------------- |          |       U  |          |          |
4 BE_ACSEFC --------------------- |          |         U| U U      |          |
5 BE_ACSEAM --------------------- |          |          |     U U U|          |
                                  +----------+----------+----------+----------+
6 BE_ACSEUP --------------------- |          |          |          | U U      |
7 BE_ACSELI --------------------- |          |     U    |   U      |     U    |
8 BE_ACSECF --------------------- |          |          |   U      |   U   U U|
----------------------------------+----------+----------+----------+----------+
Count Table for Row Names

 Row              Name               Type    SUBPARTS UTILIZES  Both

  1  BE_ACPEPC                       PROCESS    0         7       0
  2  BE_ACSERI                       PROCESS    0         7       0
  3  BE_ACSESB                       PROCESS    0         1       0
  4  BE_ACSEFC                       PROCESS    0         3       0
  5  BE_ACSEAM                       PROCESS    0         3       0
  6  BE_ACSEUP                       PROCESS    0         2       0
  7  BE_ACSELI                       PROCESS    0         3       0
  8  BE_ACSECF                       PROCESS    0         4       0

     Total                                      0        30       0
     Average                                   0.00      3.75
```

Count Table for Column Names

Column	Name	Type	PART OF	UTILIZED	Both
1 BE_ACSERI		PROCESS	0	1	0
2 BE_ACSETF		PROCESS	0	2	0
3 BE_ACSEPT		PROCESS	0	2	0
4 BE_ACSESB		PROCESS	0	2	0
5 BE_ACSEAM		PROCESS	0	2	0
6 BE_ACSEUF		PROCESS	0	2	0
7 BE_ACSEFC		PROCESS	0	2	0
8 BE_ACSEEF		PROCESS	0	2	0
9 BE_ACSESK		PROCESS	0	1	0
10 BE_ACSELO		PROCESS	0	1	0
11 BE_ACSERC		PROCESS	0	1	0
12 BE_ACSEEC		PROCESS	0	3	0
13 BE_ACSELS		PROCESS	0	1	0
14 BE_ACSETS		PROCESS	0	1	0
15 BE_ACSERA		PROCESS	0	1	0
16 BE_ACSELI		PROCESS	0	1	0
17 BE_ACSECF		PROCESS	0	2	0
18 BE_ACSULK		PROCESS	0	1	0
19 BE_ACSEQQ		PROCESS	0	1	0
20 BE_ACSECC		PROCESS	0	1	0
Total			0	30	0
Average			0.00	1.50	

```
  1 BE_ACPEPC                                                                      PROCESS

                  DESCRIPTION:
                              Monitor  the processing flow of input  self-correlation  module
                              Entity Data Records through the cross-correlation module.

                     KEYWORDS: algorithm                      cross-correlation
                               multiple

                      SOURCES: SS22-43 Part II

                     SECURITY: U

                      RESP PD: D. Yarbrough

                    ATTRIBUTE:                        VALUE:
                               generic-synonym                'BE_Driver'
                               type-of-source                 'document'
                               date-acquired                  '10/08/82'
                               validator                      'USAICS'
                               date-validated                 BE_date-validated
                               mathematical-field             'Event_Queueing'
                               tree-level                     'ROOT'
                               USAICS-term                    algorithm

  2 BE_ACSEAM                                                                      PROCESS

                  DESCRIPTION:
                              Calculates "association measure" for input EDR
                              candidate/parent pair.
                                   If component, count this type of parent < min,
                                       affiliation score =1
                                   If greater than max,
                                       affiliation score =0
                                   else compute affiliation score = linear interpolation.
                              ACSELS calculates spacial relation (location) score, and
                              ACSETS calculates time decay score.  Association measure
                              is convex combination of affiliation, location and time
                              decay scores.

                     KEYWORDS: algorithm                      cross-correlation
                               multiple

                      SOURCES: SS22-43 Part II

                     SECURITY: U

                      RESP PD: D  Yarbrough
```

```
        ATTRIBUTE:                       VALUE:
                 generic-synonym                    'BE_Figure_of_Merit_Tests'
                 type-of-source                     'document'
                 date-acquired                      '10/08/82'
                 validator                          'USAICS'
                 date-validated                     BE_date-validated
                 mathematical-field                 'Descriptive_Statistics'
                 tree-level                         'MIDDLE'
                 USAICS-term                        algorithm

3 BE_ACSECF                                                               PROCESS

      DESCRIPTION:
                 Changes fields in an Entity Data Record.

         KEYWORDS: algorithm                        cross-correlation
                   multiple

          SOURCES: SS22-43 Part II

         SECURITY: U

          RESP PD: D. Yarbrough

        ATTRIBUTE:                       VALUE:
                 generic-synonym                    'BE_DB/Event_Queue_Handling'
                 type-of-source                     'document'
                 date-acquired                      '10/08/82'
                 validator                          'USAICS'
                 date-validated                     BE_date-validated
                 mathematical-field                 'Encoding/Decoding'
                 tree-level                         'MIDDLE'
                 USAICS-term                        algorithm

4 BE_ACSEFC                                                               PROCESS

      DESCRIPTION:
                 Finds Entity Data Record (EDR) within good subgrid boundaries,
                 converts   grid-subgrid  to  easting-northing,   determines  if
                 candidate inside or outside search ellipse  (ACSELD).  retrieve
                 index  record  of candidate EDR (ACSERC) and retrieves  entries
                 from candidate EDR by field number (ACSEEC).

         KEYWORDS: algorithm                        cross-correlation
                   multiple

          SOURCES: SS22-43 Part II
```

```
              SECURITY: U

               RESP PD: D. Yarbrough

             ATTRIBUTE:                              VALUE:
                        generic-synonym                     'BE_Database_Search'
                        type-of-source                      'document'
                        date-acquired                       '10/08/82'
                        validator                           'USAICS'
                        date-validated                      BE_date-validated
                        mathematical-field                  'Searching'
                        tree-level                          'MIDDLE'
                        USAICS-term                         algorithm

5 BE_ACSELI                                                                         PROCESS

           DESCRIPTION:
                        Performs actual candidate/parent link.

              KEYWORDS: algorithm                           cross-correlation
                        multiple

               SOURCES: SS22-43 Part II

              SECURITY: U

               RESP PD: D. Yarbrough

             ATTRIBUTE:                              VALUE:
                        generic-synonym                     'BE_Entity_Handling'
                        type-of-source                      'document'
                        date-acquired                       '10/08/82'
                        validator                           'USAICS'
                        date-validated                      BE_date-validated
                        mathematical-field                  'Encoding/Decoding'
                        tree-level                          'MIDDLE'
                        USAICS-term                         algorithm

6 BE_ACSEPT                                                                         PROCESS

           DESCRIPTION:
                        Retrieves from the correlation templates file the  template
                        entry  of  possible parent subject  type  via a pointer
                        from the correlation template file directory entry.

              KEYWORDS: algorithm                           cross-correlation
                        multiple
```

```
         SOURCES: SS22-43 Part II

        SECURITY: U

         RESP PD: D. Yarbrough

       ATTRIBUTE:                        VALUE:
                  generic-synonym                 'BE_Template_Management'
                  type-of-source                  'document'
                  date-acquired                   '10/08/82'
                  validator                       'USAICS'
                  date-validated                  BE_date-validated
                  mathematical-field              'Encoding/Decoding'
                  tree-level                      'MIDDLE'
                  USAICS-term                     algorithm

7 BE_ACSERI                                                              PROCESS

     DESCRIPTION:
                  Select  Entity Data Records from Correlated Database based  on
                  master key and access fields from input EDR (using ACSEEF).

        KEYWORDS: algorithm                       cross-correlation
                  multiple

         SOURCES: SS22-43 Part II

        SECURITY: U

         RESP PD: D. Yarbrough

       ATTRIBUTE:                        VALUE:
                  generic-synonym                 'BE_Database_Search'
                  type-of-source                  'document'
                  date-acquired                   '10/08/82'
                  validator                       'USAICS'
                  date-validated                  BE_date-validated
                  mathematical-field              'Searching'
                  tree-level                      'MIDDLE'
                  USAICS-term                     algorithm

8 BE_ACSESB                                                              PROCESS

     DESCRIPTION:
                  For cross-correlation, calculates error ellipses (inner
                  and outer) for search for Entity Data Records of a
                  parent template type (location only), converts search para-
                  meters to grid and subgrid (ACSESK) and builds the
```

combined database (CDB) search key.

KEYWORDS: algorithm multiple cross-correlation

SOURCES: SS22-43 Part II

SECURITY: U

RESP PD: D. Yarbrough

ATTRIBUTE:	VALUE:
generic-synonym	'BE_Database_Search'
type-of-source	'document'
date-acquired	'10/08/82'
validator	'USAICS'
date-validated	BE_date-validated
mathematical-field	'Descriptive_Statistics'
tree-level	'MIDDLE'
USAICS-term	algorithm

9 BE_ACSEIP PROCESS

DESCRIPTION:

Get pointer to template of Entity Data Record subject type and to all potential parent subject types.

KEYWORDS: algorithm multiple cross-correlation

SOURCES: SS22-43 Part II

SECURITY: U

RESP PD: D. Yarbrough

ATTRIBUTE:	VALUE:
generic-synonym	'BE_Template_Management'
type-of-source	'document'
date-acquired	'10/08/82'
validator	'USAICS'
date-validated	BE_date-validated
mathematical-field	'Encoding/Decoding'
tree-level	'MIDDLE'
USAICS-term	algorithm

10 BE_ACSEUP PROCESS

```
DESCRIPTION:
            When tests passed, links candidate and parent Entity
            Data Records (ACSELI),  changes required  input  Entity
            Data   Record  Fields  (ACSECF)  and  retrieves  and
            modifies appropriate compound or complex element

   KEYWORDS: algorithm                         cross-correlation
             multiple

    SOURCES: SS22-43 Part II

   SECURITY: U

    RESP PD: D. Yarbrough

  ATTRIBUTE:                          VALUE:
             generic-synonym                  'BE_Database_Entity_Update'
             type-of-source                   'document'
             date-acquired                    '10/08/82'
             validator                        'USAICS'
             date-validated                   BE_date-validated
             mathematical-field               'Encoding/Decoding'
             tree-level                       'MIDDLE'
             USAICS-term                      algorithm
```

```
                              Name Selection

     1     TP_Amesur                      PROCESS
     2     TP_Bestfn                      PROCESS
     3     TP_Compar                      PROCESS
     4     TP_Consas                      PROCESS
     5     TP_F2fas                       PROCESS
     6     TP_F2subf                      PROCESS
     7     TP_Getemp                      PROCESS
     8     TP_Hierad                      PROCESS
     9     TP_Sequas                      PROCESS
```

```
                              USAICS/USAMS - VAX/VMS

                             Utilizes Analysis Report

  Utilizes Structure

   COUNT LEVEL          NAME

     1  1.0   TP_Amesur
  PSA461: "TP_Amesur" has no SUBPARTS and does not UTILIZE any PROCESS.
     2  2.0   TP_Bestfn
     3  2.1      TP_Amesur
     4  3.0   TP_Compar
     5  3.1      TP_Consas
     6  4.0   TP_Consas
  PSA461: "TP_Consas" has no SUBPARTS and does not UTILIZE any PROCESS.
     7  5.0   TP_F2fas
     8  5.1      TP_Getemp
     9  5.2      TP_Bestfn
    10  5.2.1       TP_Amesur
    11  5.3      TP_Hierad
    12  6.0   TP_F2subf
  PSA461: "TP_F2subf" has no SUBPARTS and does not UTILIZE any PROCESS.
    13  7.0   TP_Getemp
  PSA461: "TP_Getemp" has no SUBPARTS and does not UTILIZE any PROCESS.
    14  8.0   TP_Hierad
  PSA461: "TP_Hierad" has no SUBPARTS and does not UTILIZE any PROCESS.
    15  9.0   TP_Sequas
    16  9.1      TP_Getemp
    17  9.2      TP_Compar
    18  9.2.1       TP_Consas
    19  9.3      TP_F2subf
    20  9.4      TP_F2fas
    21  9.4.1       TP_Getemp
    22  9.4.2       TP_Bestfn
    23  9.4.2.1        TP_Amesur
    24  9.4.3       TP_Hierad
```

Utilizes Matrix

Explanation of the Utilizes Matrix:

The rows are input PROCESS names, and the columns are
PROCESSES UTILIZED by (or a SUBPART of) the rows.

```
     (i, j) value   meaning
     ------------   ------------------------------------

          U         Column j is UTILIZED by Row i
          S         Column j is a PART of Row i
          B         Column j is both UTILIZED by, and a PART of Row i
```

```
                 8 TP_F2fas -------------------------- /
               7 TP_F2subf ------------------------- / |
             6 TP_Compar ------------------------- / | |
                                                   | | |
          5 TP_Hierad ------------------------- /  | | |
        4 TP_Bestfn ------------------------- / |  | | |
      3 TP_Getemp ------------------------- / | |  | | |
    2 TP_Consas ------------------------- / | | |  | | |
  1 TP_Amesur ------------------------- / | | | |  | | |
                                        | | | | |  | | |
--------------------------------------+----------+------+
1 TP_Bestfn ------------------------- | U        |      |
2 TP_Compar ------------------------- |   U      |      |
3 TP_F2fas -------------------------- |     U U U|      |
4 TP_Sequas ------------------------- |     U    | U U U|
--------------------------------------+----------+------+
Count Table for Row Names

 Row                  Name                  Type   SUBPARTS UTILIZES  Both

  1  TP_Bestfn                              PROCESS   0        1        0
  2  TP_Compar                              PROCESS   0        1        0
  3  TP_F2fas                               PROCESS   0        3        0
  4  TP_Sequas                              PROCESS   0        4        0

     Total:                                           0        9        0
     Average:                                       0.00     2.25
```

```
Count Table for Column Names

 Column              Name               Type   PART OF  UTILIZED  Both

  1  TP_Amesur                          PROCESS    0         1       0
  2  TP_Consas                          PROCESS    0         1       0
  3  TP_Getemp                          PROCESS    0         2       0
  4  TP_Bestfn                          PROCESS    0         1       0
  5  TP_Hierad                          PROCESS    0         1       0
  6  TP_Compar                          PROCESS    0         1       0
  7  TP_F2subf                          PROCESS    0         1       0
  8  TP_F2fas                           PROCESS    0         1       0

     Total:                                        0         9       0
     Average:                                   0.00      1.13
```

```
                                 USAICS/USAMS - VAX/VMS

                                    Dictionary Report

  1 TP_Amesur                                                                    PROCESS

                        DESCRIPTION:
                               This is a stub for a routine to compare a unit and potential
                               parent and quantify the chance of their being related based on a
                               template.

                            SYNONYM: TP_Association_measure

                           KEYWORDS: algorithm                       dummy

                          ATTRIBUTE:                    VALUE:
                               USAICS-term                       algorithm
                               type-of-source                    'listing'
                               date-acquired                     '03/15/84'
                               date-of-source                    'June 1979'
                               date-validated                    TP_date-validated
                               processing-done                   'structure analyzed'
                               tree-level                        'leaf'
                               development-state                 'n EMSIM'

  2 TP_Bestfn                                                                    PROCESS

                        DESCRIPTION:
                               Although not precisely a stub, this algorithm is clearly designed
                               to link force-to-force assessment (F2fas) and the figure of merit
                               test (Amesur) which is a stub with no operating code.

                           KEYWORDS: algorithm                       cross-correlation

                          ATTRIBUTE:                    VALUE:
                               USAICS-term                       algorithm
                               type-of-source                    'listing'
                               date-acquired                     '03/15/84'
                               date-of-source                    'June 1979'
                               date-validated                    TP_date-validated
                               processing-done                   'structure analyzed'
                               mathematical-field                'linear equations'
                               tree-level                        'middle'
                               development-state                 'In EMSIM'

  3 TP_Compar                                                                    PROCESS

                        DESCRIPTION:
                               COMPAR controls the offspring consistency check for a parent unit
                               specified by number.  Its main function seems to be manipulating
                               pointers and fetching data.  The low robustness score is based on
                               (1) specific array elements are fetched without any array bounds
                               checking (except what may be done at run time), and (2) some vari-
                               ables appear confused within the context of this module, e.g.
                               CONCNT and CNTCNT
```

```
        SYNONYM:  TP_Compare

       KEYWORDS:  algorithm                        cross-correlation

      ATTRIBUTE:                          VALUE:
                  USAICS-term                      algorithm
                  type-of-source                   'listing'
                  date-acquired                    '03/15/84'
                  date-of-source                   'June 1979'
                  date-validated                   TP_date-validated
                  processing-done                  'structure analyzed'
                  mathematical-field               'qualitative methods'
                  tree-level                       'middle'
                  development-state                'In EMSIM'

4 TP_Consas                                                                PROCESS

    DESCRIPTION:
                  CONSAS checks type, subtype and level are consistent for two input
                   units, one the new report and one from the database.  For subtype
                   headquarters are handled separately as they can have several sub-
                  types as offspring.

        SYNONYM:  TP_Consistency_assessment

       KEYWORDS:  algorithm                        cross-correlation

      ATTRIBUTE:                          VALUE:
                  USAICS-term                      algorithm
                  type-of-source                   'listing'
                  date-acquired                    '03/15/84'
                  date-of-source                   'June 1979'
                  date-validated                   TP_date-validated
                  processing-done                  'structure analyzed'
                  mathematical-field               'logic'
                  requirements-performance         'leaf'
                  development-state                'In EMSIM'

5 TP_F2fas                                                                 PROCESS

    DESCRIPTION:
                  Force-to-force assessment finds the best parent for a sibling,
                  finding suitable bretheren using a parent.  This method can lead
                  to two possible "best" parents - that for this unit alone and that
                  for its family.  If this occurs one is chosen.  The robustness
                  rating is because pointer values arising from computations and
                  logic are not checked in the code prior to database access.

        SYNONYM:  TP_Force_to_force
```

```
                                   Dictionary Report

                 KEYWORDS: algorithm                      cross-correlation

                ATTRIBUTE:                          VALUE:
                           USAICS-term                    algorithm
                           type-of-source                 'listing'
                           date-acquired                  '03/15/84'
                           date-of-source                 'June 1979'
                           date-validated                 TP_date-validated
                           processing-done                'structure analyzed'
                           mathematical-field             'logic'
                           tree-level                     'middle'
                           development-state              'In EMSIM'

 6 TP_F2subf                                                                   PROCESS

              DESCRIPTION:
                           Force-to-subforce is a stub.

                  SYNONYM: TP_Force_to_subforce

                 KEYWORDS: algorithm                      dummy

                ATTRIBUTE:                          VALUE:
                           USAICS-term                    algorithm
                           type-of-source                 'listing'
                           date-acquired                  '03/15/84'
                           date-of-source                 'June 1979'
                           date-validated                 TP_date-validated
                           processing-done                'structure analyzed'

 7 TP_Getemp                                                                   PROCESS

              DESCRIPTION:
                           Getemp fetches a template  from the database that has been
                           requested by number.

                  SYNONYM: TP_Get_template

                 KEYWORDS: algorithm                      database-interface

                ATTRIBUTE:                          VALUE:
                           USAICS-term                    algorithm
                           type-of-source                 'listing'
                           date-acquired                  '03/15/84'
                           date-of-source                 'June 1979'
                           date-validated                 TP_date-validated
                           processing-done                'structure analyzed'
                           mathematical-field             'encoding/decoding'
                           tree-level                     'leaf'
                           development-state              'In EMSIM'
```

```
                              USAICS/USAMS - VAX/VMS

                                 Dictionary Report

   8 TP_Hierad                                                                  PROCESS

                    DESCRIPTION:
                          Hierad adds an offspring to the parents family.

                        SYNONYM:  TP_Hierarchical_addition

                       KEYWORDS:  algorithm                       database-interface

                      ATTRIBUTE:                       VALUE:
                          USAICS-term                     algorithm
                          type-of-source                  'listing'
                          date-acquired                   '03/15/84'
                          date-of-source                  'June 1979'
                          date-validated                  TP_date-validated
                          processing-done                 'structure analysed'
                          mathematical-field              'encoding/decoding'
                          tree-level                      'leaf'
                          development-state               'In EMSIM'

   9 TP_Sequas                                                                  PROCESS

                    DESCRIPTION:
                          Sequas performs the logic and searches to associate parent, off-
                          spring, and sibling units.  It determines which templates apply
                          and uses them to structure the searches.  When potential associa-
                          tions are made, offspring consistency is checked.  It does not
                          seem to be "used by" any routine in TEMPRO.

                        SYNONYM:  TP_Sequential_assessment

                       KEYWORDS:  algorithm                       cross-correlation

                      ATTRIBUTE:                       VALUE:
                          USAICS-term                     algorithm
                          type-of-source                  'listing'
                          date-acquired                   '03/15/84'
                          date-of-source                  'June 1979'
                          date-validated                  TP_date-validated
                          processing-done                 'structure analyzed'
                          mathematical-field              'qualitative methods'
                          tree-level                      'root'
                          development-state               'In EMSIM'
```

APPENDIX C

MESSAGE FORMATS

This appendix contains excerpts from the USAMS database document by Desiree Yarbrough, dated December 30, 1983, JINTACCS Message Report. The information is based on Intelligence Message Formatting and Procedures Users Handbook (DRSEL-SEI-ATU, Fort Monmouth, New Jersey, dated 21 September 1981).

```
                                   Name Type   Synonym                            USAICS Term


JI_AKNLDG                          INPUT       JI_Acknowledge_Message             message
JI_COMSPOT                           "         JI_Communications_Spot_Report        "
JI_DISUM                             "         JI_Daily_Intelligence_Summary        "
JI_ERTM                              "         JI_Elint_Requirement_Task_Msg        "
JI_IIR                               "         JI_Imagery_Interpretation_Rep        "
JI_INTREP                            "         JI_Intelligence_Report               "
JI_INTSUM                            "         JI_Intelligence_Summary              "
JI_JRSRR                             "         JI_JNT_REM_SENSOR_REP/REQ            "
JI_JTACSURVREQ                       "         JI_Joint_Tact_Surveill_Reqt          "
JI_MIJIFEEDER                        "         JI_MIJIFEEDER_Report                 "
JI_MISREP                            "         JI_Mission_Report                    "
JI_MSGCHANGEREP                      "         JI_Message_Change_Report             "
JI_RECCEXREP                         "         JI_RECON_Exploitation_Report         "
JI_RII                               "         JI_Req_for_Intelligence_Info         "
JI_RRII                              "         JI_Res_to_Req_for_Intel_Info         "
JI_SENREP                            "         JI_SENSOR_REPORT                     "
JI_TACELINT                          "         JI_Tactical_ELINT_Report             "
JI_TGTINFOREP                        "         JI_Target_Information_Report         "
```

```
JI_AKNLDG
DESCRIPTION;
   Full title is Acknowledge Message (AKNLDG).
   To acknowledge receipt of a message and indicate planned or
 accomplished action.  Transmittal of the messsage implies
 understanding of the received message.
   Standard Introductory Sets (SIS) must be filled out for each
 message. The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are as indicated below.
     SET        !
     IDENTIFIER ! INSTRUCTIONS
     ------------------------------------------
       REF      ! Occurrence in the message is (m)
       CANX     ! This set is not used here
;

JI_COMSPOT
DESCRIPTION;
   Full title is Communications Spot Report (COMSPOT).
   To advise of any situation which might cause significant
 disruption of tactical communications.
   Standard Introductory Sets (SIS) must be filled out for each
 message. The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are as indicated below.
   SET          !
   IDENTIFIER   !   INSTRUCTIONS
  ----------------------------------------------------
   REF          ! Occurrence catagory in message is (o,r)
   CANX         ! This set is not used here
   NARR         ! Not used in this message
;

JI_DISUM
DESCRIPTION;
 Full title is Daily Intelligence Summary (DISUM).
 The Commander Joint Task Force utilizes the DISUM to provide a
 summary of all significant intelligence produced/collected
 during the previous 24-hour period as specified.
   Standard Introductory Sets (SIS) must be filled out for each
 message. The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are as indicated below.
   SET          !
   IDENTIFIER   !  INSTRUCTIONS
  -------------------------------------------------------
   REF          !  Occurrence catagory here is (o,r)
   CANX         !  This set is not used
```

```
;

JI_ERTM
DESCRIPTION;
    Full title is ELINT Requirement Tasking Message (ERTM).
    For use by operational commanders to task resources under their
 operational control for the purpose of ELINT collection from
 sources outside their operational control.
   Standard Introductory Sets (SIS) must be filled out for each
 message. The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are as indicated below.

   SET          |
   IDENTIFIER | INSTRUCTIONS
 ---------------------------------------------------------------
   REF          | Occurrence catagory of message is (o,r)
;

JI_IIR
DESCRIPTION;
   Full title is Imagery Interpretation Report (IIR).
   The Imagery Interpretation Report (IIR) is a single messge
 format which can be used to report Initial Imagery Interpretation
 (IPIR),Supplemental Imagery Interpretation(SUPIR), or abbreviated
 formats for tactical reporting.  The purpose of each message
 variation is defined as:
 a. The IPIR is designed to provide Initial Imagery Interpretation
    or more detailed information than that provided by RECCEXREP.
 b. The SUPIR is accomplished as a comprehensive second phase
    exploitation.
   Standard Introductory Sets (SIS) must be filled out for each
 message. The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are as indicated below.
   SET          |
   IDENTIFIER | INSTRUCTION
 -------------------------------------------------------------
   REF          | Occurrence catagory of message is (o,r)
   CANX         | This set is not used here.
;

JI_INTREP
DESCRIPTION;
 Full title is Intelligence Report (INTREP).
 Unit to provide for the joint exchange of information obtained
 through tactical collection efforts.  The INTREP provides timely
 information regarding events that could have an immediate and
```

significant effection current planning and operations, or information that maybe of timely interest at the national level. This message is the primary means of reporting HUMIT/Counterintelligence information.

Standard Introductory Sets (SIS) must be filled out for each message. The instructions for completing the SIS are contained under the tab marked SIS. Modification to those instructions that apply to this message are as indicated below.

```
   SET         !
   IDENTIFIER  ! INSTRUCTIONS
 ---------------------------------------------------------
   REF         ! Occurrence category is (o,r)
   CANX        ! Not used here
;
```

JI_INTSUM
DESCRIPTION;

Full title is Intelligence Summary (INTSUM).
The INTSUM provides a brief summary of information of Intelligence interest covering a specific period of time, as specified by Joint Force Commander. The INTSUM provides a summary of the enemy situation in forward and rear areas, enemy operations and capabilities, and weather and terrain characteristics. The INTSUM reflects the intelligence staff officer's interpretation and conclusions, as to enemy capabilities and probable courses of action. It is prepared by components and lower echelons as directed and provides the major input for the DISUM.

Standard Introductory Sets (SIS) must be filled out for each message. The instructions for completing the SIS are contained under the tab marked SIS. Modification to those instructions that apply to this message are as indicated below.

```
  SET
  IDENTIFIER ! INSTRUCTIONS
 ---------------------------------------------------------
  REF        ! Occurrence in this message is (o,r)
  CANX       ! Not used in this message

;
```

JI_JRSRR
DESCRIPTION;

Full title is Joint Remote Sensor Report/Request(JRSRR).
Used to coordinate the use of, and assist in the management of, intelligence sensors among the components of the JOINT FORCE. It will normally be used to request sensor support or report on sensor support status. Additional usage includes reporting on sensor implantation, monitoring change/termination status, and sensor removal

Standard Introductory Sets (SIS) must be filled out for each

```
message. The instructions for completing the SIS are contained
under the tab marked SIS.  Modification to those instructions
that apply to this message are as indicated below.
  SET
  IDENTIFIER |  INSTRUCTIONS
--------------------------------------------------------------
  REF        |  Occurence is message is (o,r)
  CANX       |  Not used in this message
;


JI_JTACSURVREQ
)ESCRIPTION;
  Full name is Joint Tactical Surveillance Request (JTACSURVREQ).
  Purpose is to identify tactical surveillance requirements
against a specific target/area.
  Standard Introductory Sets (SIS) must be filled out for each
message. The instructions for completing the SIS are contained
under the tab marked SIS.  Modification to those instructions
that apply to this message are as indicated below.
  SET        |
  IDENTIFIER | INSTRUCTIONS
 -----------------------------------------------------------
  REF        | Occurence category in message is (o,r)
  CANX       | This set is not used in this message
;


JI_MIJIFEEDER
)ESCRIPTION;
 Full title is Meaconing,Intrusion,Jamming,and Interference Feeder
 Report (MIJIFEEDER).
 Used as a primary means of sharing MIJI incidents in a timely
 manner, and provides for joint exchange of tactical MIJI
 information, including electro-optic interference.
  Standard Introductory Sets (SIS) must be filled out for each
 message.  The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are indicated below.
  SET
  IDENTIFIER  |  INSTRUCTIONS
 ---------------------------------------------------------------
  REF         |  Occurance catagory in message is (o,r)
  CANX        |  This set is not used in this message
;


JI_MISREP
DESCRIPTION;
 Full title is Mission Report (MISREP).
 The MISREP is used by all air units to report the results of all
```

```
missions and significant non-imagery recorded sightings along the
flight route. It may also amplify an inflight report.
 Standard Introductory Sets (SIS) must be filled out for each
message.  The instructions for completing the SIS are contained
under the tab marked SIS.  Modification to those instructions
that apply to this message are indicated below.
  SET
  IDENTIFIER  |  INSTRUCTIONS
-------------------------------------------------------------------
  REF         |  Occurrence catagory in this message is (o,r)
  CANX        |  This set is not used in this message.
;

JI_MSGCHANGEREP
DESCRIPTION;
 Full title is Message Change Report (MSGCHANGEREP).
 To make amendments to the main text of previously transmitted
 messages and/or to cancel complete messages.
  Standard Introductory Sets (SIS) must be filled out for each
 message.  The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are indicated below.
  SET
  IDENTIFICATION  |  INSTRUCTIONS
-------------------------------------------------------------------
  REF             |  Occurrence catagory is (m)
  NARR            |  This set not used here
  CANX            |  This set not used here
;

JI_RECCEXREP
DESCRIPTION;
 Full title is Reconnaisance Exploitation Report.
   The RECCEXREPis a high priority report of time sensitive targets
 of significant tactical importance, used to alerting report
 to initiate immediate operational response.
   Standard Introductory Sets (SIS) must be filled out for each
 message. The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are as indicated below.
   SET         |
   IDENTIFIER  | INSTRUCTIONS
   ------------------------------------------------------
   REF         | Occurrence catagory is (o,r)
   CANX        | This set is not used in this message.
;
```

```
JI_RII
DESCRIPTION;
    Full title is Request for Intelligence Information (RII).  Used to
request intelligence information from other units.  It may  also be
used to request the status of an anticipated response to  another
request.
    Standard Introductory Sets (SIS) must be filled out for
each  message.  The instructions for completing the SIS are contained
under the tab marked SIS.  Modification to those instructions  that
apply to this message are indicated below.
   SET
   IDENTIFIER | INSTRUCTIONS
 ---------------------------------------------------------------
   REF         | Occurrence category in this message is (o,r)
   CANX        | This set is not used in this message
;


JI_RRII
DESCRIPTION;
 Full title is Response to Request for Intelligence Information,
 (RRII).  Used to reply to requests for Intelligence Information.
 If the information is contained in a previous message, the RRII
 should reference that message.
  Standard Introductory Sets (SIS) must be filled out for each
 message.  The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are indicated below.
   SET
   IDENTIFIER | INSTRUCTIONS
 --------------------------------------------------------------
   REF         | Occurrence in the message is (o,r)
   CANX        | This set is not used in the message
;


JI_SENREP
DESCRIPTION;
 Full title is Sensor Report (SENREP).
 Used to disseminate Intelligence obtained from information
 collected by remote ground sensors regarding enemy movements/
 activities, to support estimates of enemy capabilities and
 intentions, and to permit quick response attacks by Joint Force
 resourses.
  Standard Introductory Sets (SIS) must be filled out for each
 message.  The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are indicated below.
   SET
   IDENTIFIER | INSTRUCTIONS
 ---------------------------------------------------------------
```

```
   REF        | Occurrence category in this message is (o,r)
   CANX       | This set is not used in this message
;


JI_TACELINT
DESCRIPTION:
 Full title is Tactical ELINT Report (TACELINT).  TACELINT is used to
report time-critical operational ELINT and  parametric information.
Information contained therein may be used   for indications andd
warning, data base maintenance, Orders of  Battle, and strike planning.
ELINT collectors use this message  format as a reporting vehicle.  The
JTF Commander uses this  message format to advise the Joint Force of
updates to the ELINT  Order of Battle/Data Base.  NOTE: Sets SOI,EMLOC,
and PRM are repeatable as a segment.  The  PRM set is repeated to
report multiple discrete RFs or the limits   when RF is sweeping,
switching, sliding, or stepping.  The PRM  set also may be repeated to
report stagger intervals, jitter  limits etc.
  Standard Introductory Sets (SIS) must be filled out for each
message.  The instructions for completing the SIS are contained  under
the tab marked SIS.  Modification to those instructions  that apply to
this message are indicated below.
   SET
   IDENTIFICATIONS | INSTRUCTIONS
 --------------------------------------------------------------
   REF             | Occurrence in this message is (o,r)
   CANX            | This set is not used here.
;


JI_TGTINFOREP
DESCRIPTION:
 Full title is Target Information Report (TGTINFOREP).
 The TGTINFOREP provides the JTF Commander and staff information to
 permit a decision to attack, the manner of attack, and
 the weapons system to be used.  Components report new targets,
 results of attacks, overrun and inactive targets and no-strike
 requests via the TGTINFOREP.  The JTF HQ will decide whether to
 place the proposed target on the joint target list, or to delete
 or change the status of known targets.  If the Joint Force
 Commander decides to add/delete/change status of these reported
 he will advise all components via the TARBUL.
  Standard Introductory Sets (SIS) must be filled out for each
 message.  The instructions for completing the SIS are contained
 under the tab marked SIS.  Modification to those instructions
 that apply to this message are indicated below.
   SET
   IDENTIFIERS | INSTRUCTIONS
 -------------------------------------------------------------------
   REF         | Occurrence category is (o,r)
   CANX        | This set is not used here.
```

APPENDIX D

OTHER REPORTS IN THIS SERIES

Gillis, J.W., Griesel. M.A., and Radbill. J.R. Analysis of Geographic Transformation Algorithms. JPL Report D-181, July, 1982.

______________________________. Correlation Algorithm Report. JPL Report D-182, September, 1982.

______________________________, and Kuo, T.-J. Intelligence Algorithm Methodology I. JPL Report D-183, August 1983.

APPENDIX E

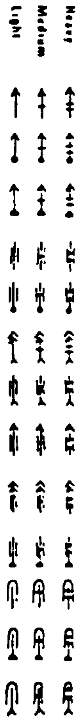
MILITARY SYMBOLS

The following list of military symbols is taken from Weapons and Tactics of the Soviet Army (David C. Isby, Jane's, London, 1981) page 10.

Unit, vehicle and other symbols

⊠	Airborne infantry
□	Air defence
⊡	Tank
□	Chemical
⊠	Naval infantry
□	Engineers
⊡	Artillery (towed or SP, weapon type shown at side)
⊠	Motorised rifle
⊠	Infantry (non-Soviet)
⊞	Medical
□	Anti-tank (any)
□	Anti-tank artillery
⊘	Reconnaissance
⊠	Special forces
→	Rocket or missile artillery
□	Service support element
◐	Supply installation (fixed)
□	Signals
SVCS	Service support
□(-)	Unit has had components detached from it
□(+)	Unit has been reinforced with non-organic assets
□HQ	Headquarters (while moving)
□	Headquarters (deployed)
□	Unit is an ad hoc or mission-specific grouping
△	Observation post
•	Squad or individual vehicle
••	Section (US usage of term)
•••	Platoon
I	Company or battery
II	Battalion
III	Regiment
x	Brigade
xx	Division
xxx	Corps

xxxx	Army
xxxxx	Front
xxxxxx	Theatre
⚐	Command post
⊡	Mortars
⊡	SAM launcher (tactical)
▯	ZSU-23-4
Ⅱ	Main battle tank
Ⅱ	Light tank
Ⅲ	Heavy tank
◊	APC or BMP
⓪	SP gun
Ⅲ	AVLB
◊	Engineer APC
⋯	Minefields
—II—	Unit boundary (here a battalion)
(…)	Unit defensive position (here a platoon)



Automatic infantry weapon

Mortar

Anti-aircraft machine gun

Artillery gun or gun/howitzer

Howitzer

Anti-tank rocket-launcher

Anti-tank gun

Recoilless rifle

Rocket launcher artillery

Anti-aircraft gun

Missile

Air-defence missile

Anti-tank missile

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER D-184	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) USAMS Cross-Correlation: Statistics, Templating and Doctrine		5. TYPE OF REPORT & PERIOD COVERED FINAL 6. PERFORMING ORG. REPORT NUMBER D-184
7. AUTHOR(s) Martha Ann Griesel, James W. Gillis, John R. Radbill, Nicky L. Sizemore		8. CONTRACT OR GRANT NUMBER(s) NAS7-918
9. PERFORMING ORGANIZATION NAME AND ADDRESS Jet Propulsion Laboratory ATTN: 171-209 California Institute of Technology 4800 Oak Grove, Pasadena, CA 91109		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS RE 182 AMEND # 187
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, USAICS ATTN: ATSI-CD-SF Ft. Huachuca, AZ 85613-7000		12. REPORT DATE February 29, 1984 13. NUMBER OF PAGES 136
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (Same as Item 11)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NONE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Dissemination

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

18. SUPPLEMENTARY NOTES

Prepared by Jet Propulsion Laboratory for the US Army Intelligence Center and School's Combat Developer's Support Facility

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

CROSS-CORRELATION, MULTIVARIATE STATISTICS, TEMPLATING, SIGINT, ELINT, IEW, CORRELATION ALGORITHM, BETA, TEMPRO, MAGIS, TCAC(D)

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

This is one of a series of algorithm analysis reports on work performed at the Jet Propulsion Laboratory for the US Army Intelligence Center and School covering selected algorithms in existing Intelligence and Electronic Warfare (IEW) systems. It focuses on cross-correlation algorithms in four fusion systems, especially on those dealing with information from Signals Intelligence (SIGINT) reports. These algorithms decide if one battlefield entity (e.g. a radar) belongs to or contains a higher or lower ech-

elon entity (e.g. an artillery battery). The decision tests mix some statistics (mainly Chi-square tests) with simple in-bounds, linear, and exponential tests based on doctrinal templates (deployment radii and Table of Organization and Equipment (TO&E) equipment lists). The mathematics of such hybrid tests is analyzed, illustrated by the tests implemented in BETA and TEMPRO (MAGIS and TCAC were also surveyed), and focusing on the underlying mathematical assumptions and military template structure. Effects of previous processing (e.g. by the sensor systems) and incoming message formats on the data distributions seen by the tests, and thus on the outcome of the tests, is also discussed.

(Battlefield Exploitation and Target Acquisition)

(Marine Air/Ground Intelligence System)

(Tempo [illegible] Program)