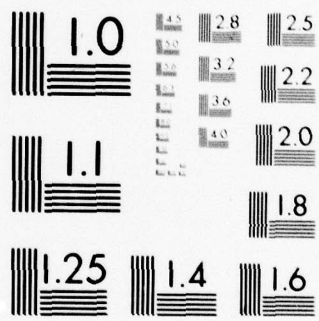
AD-A080 555 NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA F/G 9/4
HIGHER ORDER LOGIC FOR PLATFORM IDENTIFICATION IN A PRODUCTION --ETC(U)
OCT 79 R A DILLARD
UNCLASSIFIED NOSC/TD-288 NL

1 OF 1
AD
A080555

END
DATE
FILMED
3-80
DDC




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

LEVEL II

NOSC

NOSC TD 288



NOSC TD 288

Technical Document 288

# HIGHER ORDER LOGIC FOR PLATFORM IDENTIFICATION IN A PRODUCTION SYSTEM

## A method of extending the capabilities of a production system applied to tactical situation assessment

RA Dillard



17 October 1979



Prepared for
Naval Electronic Systems Command
(NAVELEX 330)
Washington DC 20360



NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

SL GUILLE, CAPT, USN
Commander

HL BLOOD
Technical Director

## ADMINISTRATIVE INFORMATION

Work was performed under Program Element 61153N, Project XRO1408, Task Area XRO140801 (NOSC Work Unit 824-CC25) by a member of the $C^2$ Information Processing Branch (Code 8242), for the Naval Electronic Systems Command (NAVELEX 330). The work documented was performed during FY 1979 and was approved for publication 17 October 1979.

Released by
RC Kolb, Head
Tactical Command Control
Division

Under authority of
JH Maynard, Head
Command Control Electronic Warfare
Systems and Technology Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

**1. REPORT NUMBER**
NOSC Technical Document 288 (TD 288)

**2. GOVT ACCESSION NO.**

**3. RECIPIENT'S CATALOG NUMBER**

**4. TITLE (and Subtitle)**
HIGHER ORDER LOGIC FOR PLATFORM IDENTIFICATION IN A PRODUCTION SYSTEM.
A method of extending the capabilities of a production system applied to tactical situation assessment.

**5. TYPE OF REPORT & PERIOD COVERED**
Technical document

**6. PERFORMING ORG. REPORT NUMBER**

**7. AUTHOR(s)**
R M Dillard

**8. CONTRACT OR GRANT NUMBER(s)**

**9. PERFORMING ORGANIZATION NAME AND ADDRESS**
Naval Ocean Systems Center
San Diego CA 92152

**10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS**
[redacted] XRO1408
XRO140801 (NOSC 824-CC25)

**11. CONTROLLING OFFICE NAME AND ADDRESS**
Naval Electronic Systems Command (NAVELEX 330)
Washington DC 20360

**12. REPORT DATE**
17 Oct[redacted]79

**13. NUMBER OF PAGES**
46

**14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)**
NOSC/TD-288

**15. SECURITY CLASS. (of this report)**
Unclassified

**15a. DECLASSIFICATION/DOWNGRADING SCHEDULE**

**16. DISTRIBUTION STATEMENT (of this Report)**

Approved for public release; distribution unlimited

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

**18. SUPPLEMENTARY NOTES**

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

Tactical analyses
Tactical data systems
Identification systems
Platform identification
Artificial intelligence
Production systems

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

Describes a method of extending the capabilities of a production system applied to tactical situation assessment, to enable the system, whatever its type, to perform the higher order logical reasoning required for the solution of many platform identification problems. The method is implemented through (1) production rules for constructing and updating an intermediate framework in the system's data base, a framework that supports the process of associating tracks with platforms, and (2) production rules for performing the association process by using, for example, the notion of complete subsets of platform files and track files, along with logical sequences of reasoning.

(Continued)

20. ABSTRACT (Continued)

Many of the system logic rules underlying this method have been implemented in an experimental production system and tested against hypothesized data. Two examples are given, along with a step-by-step description of the response of the system to each new piece of information.

The method as it is described uses only data having high confidence values. However, the conclusions which would logically follow from different assumptions about particular tracks or platforms could be determined by recording logical conclusions derived from less certain data, then reinitializing the data base by removing previous conclusions and replacing selected questionable assertions with alternative assertions.

## CONTENTS

## INTRODUCTION

In some platform identification problems, the conclusions must satisfy constraints that involve conjunctions, disjunctions, negations, combinatorial relations, etc. Their solution requires, for example, process-of-elimination reasoning. The accumulation-of-evidence reasoning and simple logical reasoning commonly used in production systems is inadequate for handling these problems. A production system capable of performing such limited reasoning can be extended, however, by the addition of system-logic rules, to perform higher order logic kinds of reasoning.

The particular problem addressed here is the matching of tracks to specific platforms. (A "track" primarily represents positional and movement information concerning a platform. It can result, for example, from a radar or sonar contact or a sighting by patrol aircraft.) Because this problem has no equivalent in nonmilitary areas of science, the many artificial intelligence techniques that have been developed for nonmilitary purposes are inapplicable or inadequate for solving it. While a specialized problem-solving technique probably could be developed which would provide the most efficient implementation of the solution process, the approach taken here was to find a way to perform the reasoning within the framework of a production system. The justification for this approach is that other types of tactical reasoning processes can be implemented in a production system (ref 1, 2), and the coordination of those processes with this logic-oriented process is best achieved in a single system where they can share a data base and a package of machine functions.

Production rules that enable the higher order logical reasoning required to associate tracks with platforms are listed in appendices A, B, and C. The implementation of rules for this purpose within a production system used for tactical situation assessment (such a system would include many other kinds of rules) is termed a Platform-Track Association Production Subsystem (PTAPS). Figure 1 gives an overview of a system extended with PTAPS. (Production systems are discussed in references 3–8.) Many of the assertions and rules needed to support the chains of logical reasoning in PTAPS are also individually useful in an unextended system – thus the overlapping areas in figure 1. Most of the PTAPS assertions are created by PTAPS rules.

While the incorporation of a PTAPS into a production system containing other kinds of reasoning has not yet been attempted, many of the PTAPS rules have been exercised for two scenarios in an experimental production system built from the network manipulation and rule evaluation functions of STAMMER (ref 1).

1. NOSC TD 252, STAMMER: System for Tactical Assessment of Multisource Messages, Even Radar, by RJ Bechtel and PH Morris, of Systems Development Corporation, May 1979.
2. Final Technical Report on TECA Development, SDC Integrated Services, Inc, 29 December 1978.
3. Artificial Intelligence, by PH Winston, Addison-Wesley, 1977.
4. Stanford AI Lab Memo AIM-270, Computer Science Dept Report STAN-CS-75-524, An Overview of Production Systems, by R Davis and J King, October 1975.
5. Stanford AI Lab Memo AIM-266, Computer Science Dept Report STAN-CS-75-519, Production Rules as a Representation for a Knowledge-Based Consultation System, by R Davis, BG Buchanan, and EH Shortliffe, October, 1975.
6. Computer-Based Medical Consultations: MYCIN, by EH Shortliffe, American Elsevier, 1976.
7. NOSC TR 364, New Methodologies for Automated Data Fusion Processing, by RA Dillard, September 1978.
8. STAMMER2. A Production System for Tactical Situation Assessment, by P Morris, D Kibler, and RJ Bechtel, SDC Integrated Services, Inc, August 1979. (Also submitted for publication as an NOSC Technical Document.)

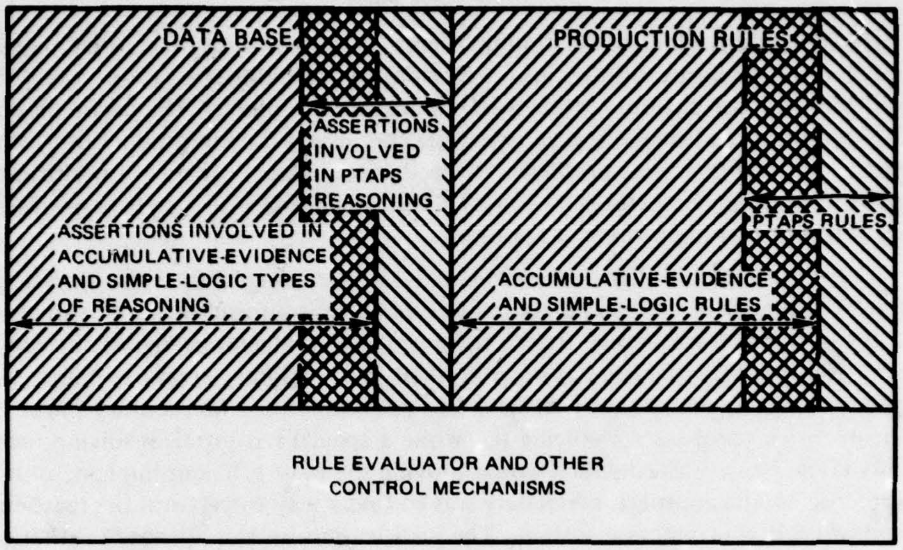



Figure 1. A production system with PTAPS, used for tactical situation assessment. Each assertion consists of two nodes (representing objects or properties) and a relation connecting them, and it can have an associated confidence. The rules are of the if-then form, and can also have confidences. Only assertions having high certainty are used by PTAPS rules.

## DATA BASE ORGANIZATION

For any defined geographical area, denoted here by "region," initial assertions are as follows (written here in the form (A is the/an R of B), where A and B are nodes and R is a relation):

RPF is the platform-file of region

RTF is the track-file of region

REF is the emission-file of region

These files receive members via PTAPS membership rules that create assertions having the relation "member." For example, ⟨platform node⟩ is a member of RPF. These and a number of other files described below constitute part of an "intermediate framework" built into the data base by PTAPS rules to permit chains of reasoning not otherwise possible.

The platform file RPF has as members all surface ships and submarines known to be or thought possibly to be in the region, with the exception of boats, aircraft, and own-force platforms. US Navy platforms and tracks are handled separately, and it is assumed in the discussions that their positions are known accurately enough that no track in RTF can be that of an own-force platform.

The primary purpose of the reasoning process implemented by the system logic rules is the matching of tracks in RTF to platforms in RPF. Initial assertions about a track, t, give its position and movement data and describe any observed physical characteristics. Ideally, each track, t, in a region's track file RTF would be associated (via an assertion: t is the track of p) with a platform, p, in the region's platform file; and p would be linked, via assertions, with its hull number, category (surface or subsurface), general type (carrier, cruiser/destroyer/frigate, amphibious, minelaying, . . ., submarine, or commercial), type (CV, CG, CLG, CA, AO, . . .), class (Kynda, Kashin, Krivak, Kara, . . .), and other identifying classifications. In practice the information about a platform in RPF often is incomplete, contact with a platform in RPF can be lost (in which case its track is eventually removed from RTF and becomes its "last-inactive-track"), some tracks may be those of platforms not in RPF, and various other inadequacies can exist.

An existing platform node, p, becomes a member of a region's platform file RPF whenever

(1) p is currently associated with a track, t (via an insertion: t is the track of p), and t is entering the region (t becomes a member of RTF at that time), or

(2) p was earlier associated with a track outside the region, now an inactive track, and p meanwhile could have entered the region.

A platform node, p, is created and becomes a member of RPF whenever

(1) a new track, t, inside the region cannot be that of any platform in RPF (t becomes a member of RTF and the assertion "t is the track of p" is made at that time), or

(2) a time-late report is received of a sighting, inside the region, of a platform which is clearly not in RPF. A track from a time-late report is temporarily treated as an active track, and the procedure is as in (1) above. The track is then removed from RTF and becomes the last-inactive-track of p if the time limit specified for active tracks has expired.

When a platform node, p, is created, all information about the physical characteristics of the platform is linked to the node via assertions.

It is assumed that no active track entered into the PTAPS data base is a false track, eg the result of a radar's false alarm. Also, no two active tracks can be the track of a single platform. The amount of time after contact is lost that must elapse before an active track is made an inactive track should depend on the situation and be specified by rules. When two contacts that are reported at different times or by different sources could possibly be of the same platform, only the most recent or most informative should be labeled an active track. However, if it is certain that they are contacts of the same platform, the data should be recorded under a single track.

Subsets of RPF and RTF are formed according to the characteristics that several platforms have in common. These subsets generally will be category-subsets, general-type-subsets, type-subsets, or class-subsets. Each subset is itself a platform file or a track file.

The status of a platform file is "complete" if every platform in the region of the kind to be kept in that file is a member of the file. (Note that members of a complete platform file need not actually be in the region.) The status of most platform files will be complete if the region is enclosed (eg the Persian Gulf, Red Sea, Mediterranean Sea) and the entrance/exit areas are continually patrolled or monitored.

A track file can also have a status of complete if every platform in that region (of that category, type, or class, if a subset of RTF) is being tracked. As a result of high-altitude surveillance, for example, a temporary status of complete can be given the surface category-subset of RTF.

When a new track or platform is submitted as data to the production system, a rule asserting that the status of a file is complete must not be allowed to fire before the membership rule for that file has been exercised. In an ordered-rule system, this is no problem. If rules are accessed in an irregular manner, an additional condition is needed in each such rule to inhibit it until the membership rule has been accessed. With the exception of the rules which assert completeness, the rules can be ordered in any manner and the final conclusions will be the same.

For convenience, it will be assumed that there is a single region of interest. Initially, RPF is declared to be a platform-file and RTF a track-file. (An assertion having the relation "is" will be expressed "B is an A," since the form "A is an R of B" sounds awkward in this case.) Also, the assertion

RTF is the corresponding-file of RPF

is made.

During rule evaluations, the variable PF can be bound to any platform file – that is, to RPF or to any platform file which is a subset of RPF. Similarly, the variable TF can be bound to RTF or any subset.

The emission file REF has, as members, emissions determined to have been emitted within the region but not from own-force emitters.

An "OR-file" is built by PTAPS rules for each member of RPF, RTF, and REF. The members of the OR-file of a platform are those tracks which have not been ruled out as the track of that platform. A platform is a member of a track's OR-file if that track has not been ruled out as a track of that platform. The OR-file of an emission has, as members, platforms which have not been ruled out as the emitting platform.

In some cases it is useful to assert impossible relationships between two nodes. For example, each track can have at most one assertion (of confidence certainty) having the relation "track." For example, t is the track of p, but it can have many assertions of the form: t is an impos-track of p.

The examples given in the next two sections illustrate the use of these files and special relationships. After each firing of a rule in the experimental system, an explanation of why the rule fired was printed. The explanation contained names of pertinent nodes involved in that particular firing. In the description in the two examples below, explanatory material is interspersed in a different manner. The rules used in the first example are contained in appendix A; those used in the second are contained in appendices A, B, and C.

## TWO-SUBMARINE EXAMPLE

An example of reasoning that is very simple for a human but somewhat difficult for a computer is the following:

> Only two submarines could be in the region – a Delta and an Echo II; and two subsurface tracks are reported. The acoustic signature of one track shows that it cannot be a Delta; therefore, it must be the Echo II, and the other track must be the Delta.

Figure 2 gives an illustration of such a situation. The sequence of reasoning steps used by PTAPS to reach this conclusion will depend upon the order in which the

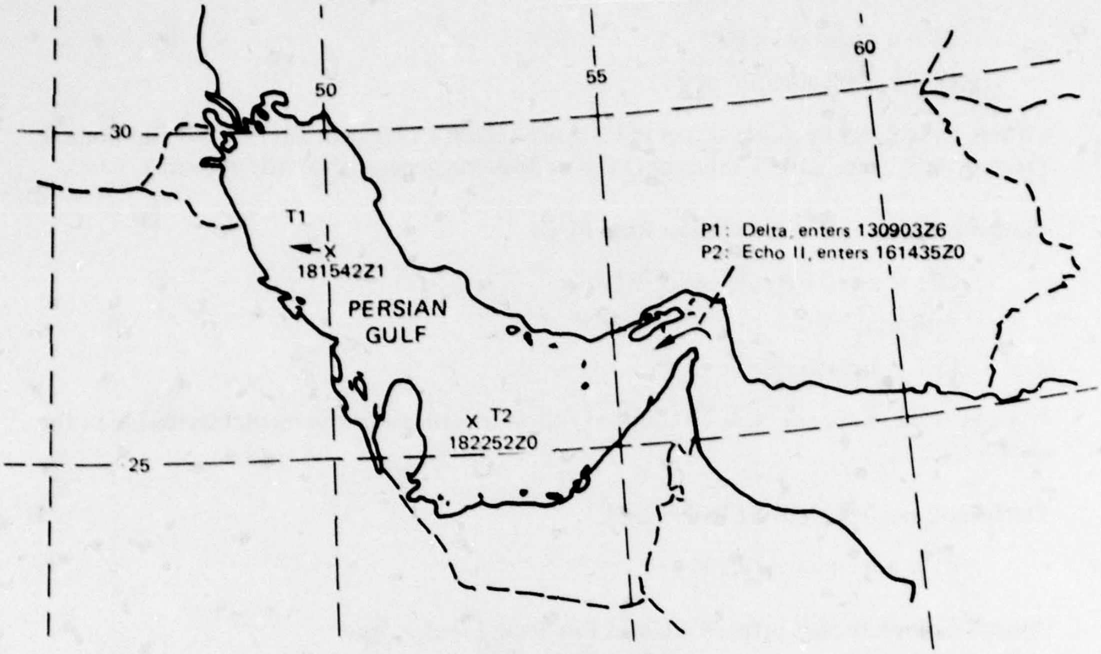



Figure 2. An example of two submarines entering an enclosed region and later being detected while inside. P1 and P2 are platform nodes, and T1 and T2 are track nodes representing active tracks at time 182252ZO. (In the date-time-group **AABBBBZC**, **AA** is the day, **BBBB** is the time, and C is the check sum.)

information (in the form of assertions) is presented to the system and upon the order in which the rules are exercised. The rules used in this example are contained in appendix A.

The information that there are two submarines, a Delta and an Echo II, can be expressed in the form of assertions as follows:

P1 is a platform

subsurface is the category of P1

Delta is the class of P1

P2 is a platform

subsurface is the category of P2

Echo II is the class of P2

P1 was the first of the two submarines to enter the region. At the time, it was associated with a track; that track is now the last inactive-track of P1. Assuming that there were no other submarines in the region at that time, the initial detection of P1 as it entered would have resulted in the creation of several additional assertions in the data base via the firing of several rules, as described below.

**Active-Track RPF Member Rule Fired . . .**

P1 is a member of RPF

ORF1 is the OR-file of P1

(When a platform becomes a member of the region's platform file, an OR-file node is created for it; here, ORF1 represents the node-name generated by the system.)

**Platform-Category Node Creation Rule Fired . . .**

PF1 is a category-subset of RPF

subsurface is the category of PF1

PF1 is a platform-file

(As discussed in appendix A, alternatively these assertions can permanently reside in the data base.)

**Platform-Category Member Rule Fired . . .**

P1 is a member of PF1

**Always-Complete Subsurface-Platform File Rule Fired . . .**

complete is the status of PF1

(The knowledge that no submarines could enter the region without being detected results from a very close monitoring of the region, using sensitive acoustic devices at the entrance to a nearly enclosed gulf or sea. This rule is not needed if PF1 permanently exists in the data base.)

P2 then entered the region; its track is now the last-inactive-track of P2. When P2 was detected entering the region, some of the additional assertions entered into the data base are as follows.

**Active-Track RPF Member Rule Fired . . .**

P2 is a member of RPF

ORF2 is the OR-file of P2

**Platform-Category Member Rule Fired . . .**

P2 is a member of PF1

The fact that there are only two submarines in the region is now represented by PF1's having a status of complete and having two members.

The platform-class node creation rule and the platform-class member rule fire, but their conclusions are not pertinent here.

The assertions in the data base at this point are shown in figure 3.

Next, consider the information "there are two subsurface tracks reported" and "the acoustic signature of one track shows that it cannot be a Delta." We assume first that the

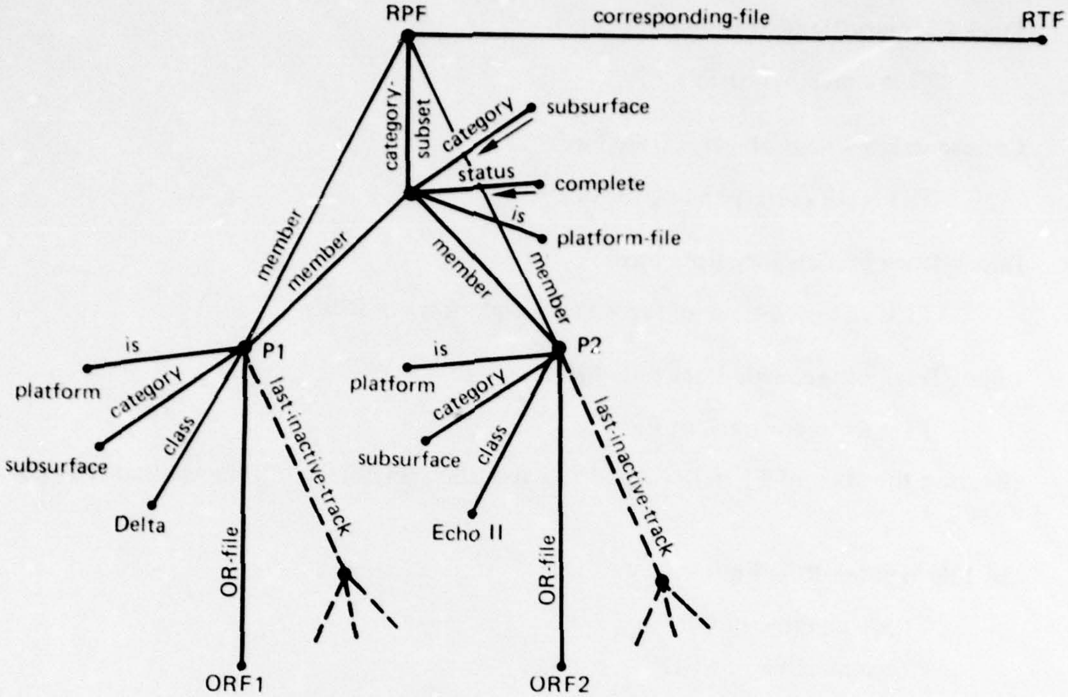



Figure 3. Pertinent assertions in data base just before track T1.

acoustic data are obtained with the first submarine track, T1. Among the assertions entered into the data base are the following:

> T1 is a track
>
> subsurface is the working-category of T1
>
> AD1 is the acoustic-data of T1
>
> ICF is an impos-class-file of AD1
>
> Delta is a member of ICF

Also, (inside-region T1) has the value "true."

Additional assertions are entered into the data base by the firing of the following rules.

**RTF Member Rule Fires . . .**

> T1 is a member of RTF
>
> FRO1 is the OR-file of T1

**Track-Category Node Creation Rule Fires . . .**

> TF1 is a category-subset of RTF
>
> subsurface is the category of TF1
>
> TF1 is a track-file

**Track-Category Member Rule Fires . . .**

T1 is a member of TF1

**Corresponding-File (Category) Rule Fires . . .**

TF1 is the corresponding-file of PF1

**Impos-Track by Category Rule Fires . . .**

T1 is an impos-track of (each surface platform in RPF)

**Impos-Track by Acoustic-Data Rule Fires . . .**

T1 is an impos-track of P1

(Because the class of P1 is Delta and the acoustic signature of T1 shows that it is not a Delta.)

**OR-File Member Rule Fires . . .**

T1 is a member of ORF2

P2 is a member of FRO1

**Complete Track-OR-File Rule Fires . . .**

Complete is the status of FRO1

(The OR-file of T1 is complete because T1 is a member of a track file (TF1) whose corresponding platform file (PF1) is complete.)

**And-Then-There-Was-One Platform Rule Fires . . .**

T1 is the track of P2

(Because the OR-file of T1 is complete and contains only P2.)

The assertions now in the data base are shown in figure 4. When the contact of another submarine is reported, new assertions are entered:

T2 is a track

subsurface is the working-category of T2

Also, (inside-region T2) has the value true. These enable further firing of the rules.

**RTF Member Rule Fires . . .**

T2 is a member of RTF

FRO2 is the OR-file of T2

**Track Category Member Rule Fires . .**

T2 is a member of TF1

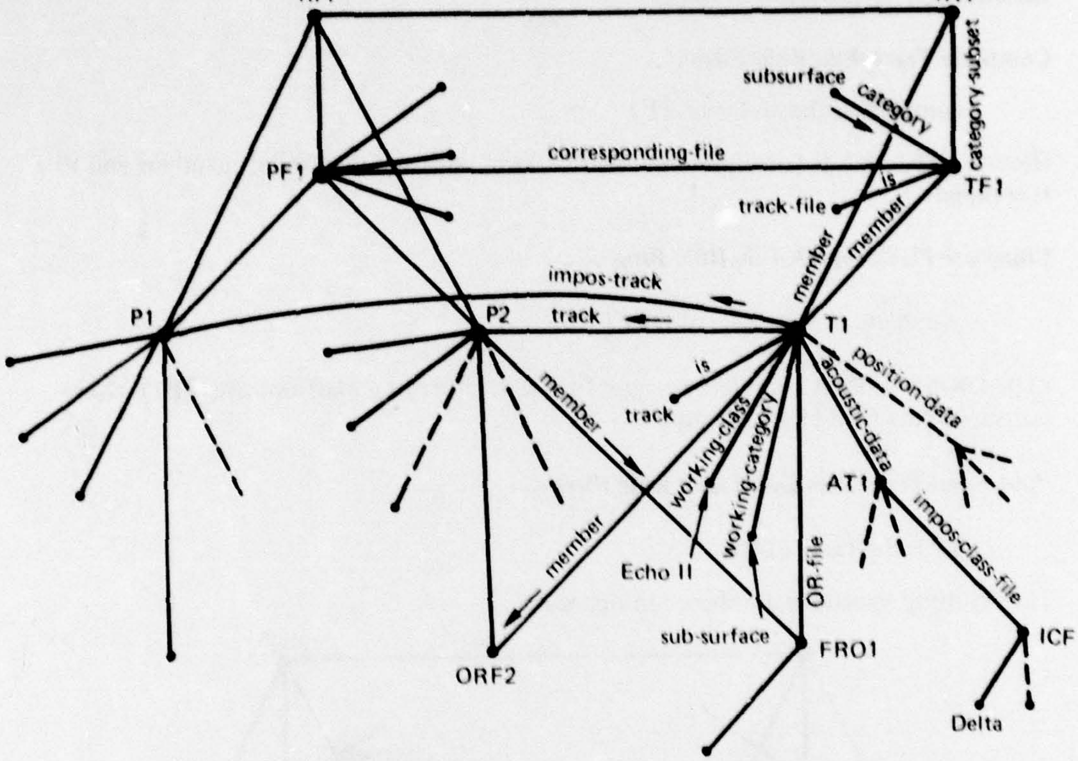



Figure 4. Pertinent assertions in data base after track T1, when no more rules can fire.

**Impos-Track by Category Rule Fires . . .**

T2 is an impos-track of (each surface platform in RPF)

**Impos-Track by Platform-Elim Rule Fires . . .**

T2 is an impos-track of P2

(Because T1 is the track of P2.)

**OR-File Member Rule Fires . . .**

T2 is a member of ORF1

P1 is a member of FRO2

**Complete Track-OR-File Rule Fires . . .**

complete is the status of FRO2

**And-Then-There-Was-One Platform Rule Fires . . .**

T2 is the track of P1

Alternatively to the last two rules:

**Complete Track-File Rule Fires . . .**

complete is the status of TF1

(Because TF1 and its corresponding file PF1 have the same number of members and PF1 is complete.)

**Complete Platform-OR-File Rule Fires . . .**

complete is the status of ORF1

(The OR-file of P1 is complete because P1 is a member of a platform file (PF1) whose corresponding file (TF1) is complete.)

**And-Then-There-Was-One Track Rule Fires . . .**

T2 is the track of P1

The resulting assertions are shown in figure 5.

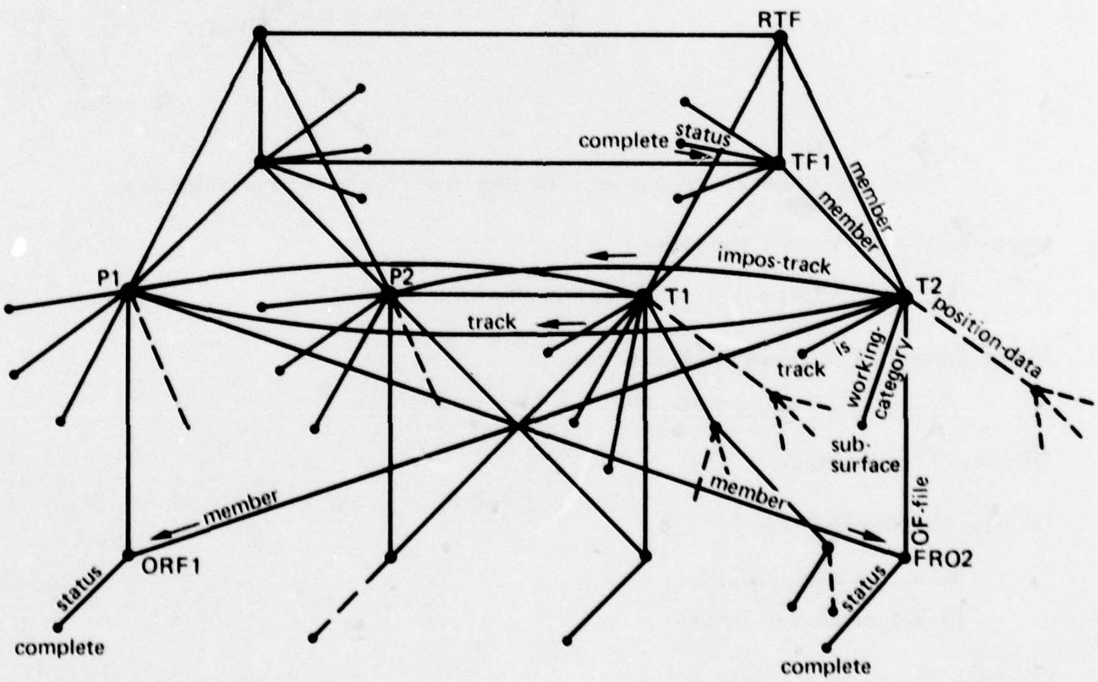



Figure 5. Pertinent assertions in data base after track T2, when no more rules can fire.

Returning to the statement of the problem, suppose that the acoustic data are obtained for the second track, T2. Then the first association (T1 is the track of P2) is about equally likely to occur via an OR-file of a platform as an OR-file of a track (with the corresponding and-then-there-was-one rule). If the second association is also made via an OR-file of a platform, the OR-file reduction rule is used by the system.

## HIGH-ALTITUDE SURVEILLANCE EXAMPLE

No radar tracks are available to own ship, because of EMCON conditions, but recent positions on all major surface ships have been obtained from a satellite radar map (fig 6). The positions of own-force ships are known, and the locations of two commercial ships are known sufficiently that they can be associated with their tracks on the map.

There are four remaining tracks (T1, T2, T3, T4), and it is concluded that these correspond to a small Soviet UNREP group (CG 155, DDG 233, AO 7, AE 12) that earlier had been reported heading for the area.

A patrol aircraft had overflown the oiler two hours earlier, and it is calculated that the oiler could not have reached the position of T1 or T2.

T1 is in the lead position, so T1 is ruled out as being either the oiler or ammunition ship.

A signal intercept is reported by the ESM system at a bearing consistent with the positions of T3 and T4. A list of ship classes having that emitter type are determined from the emitter/class file; and, of the ships in the Soviet group, only the class of the DDG 223 is on this list.

PTAPS is able to conclude in the manner described below that

T1 is the track of CG155

T2 is the track of AE12

T3 and T4 are tracks of DDG223 and AO7

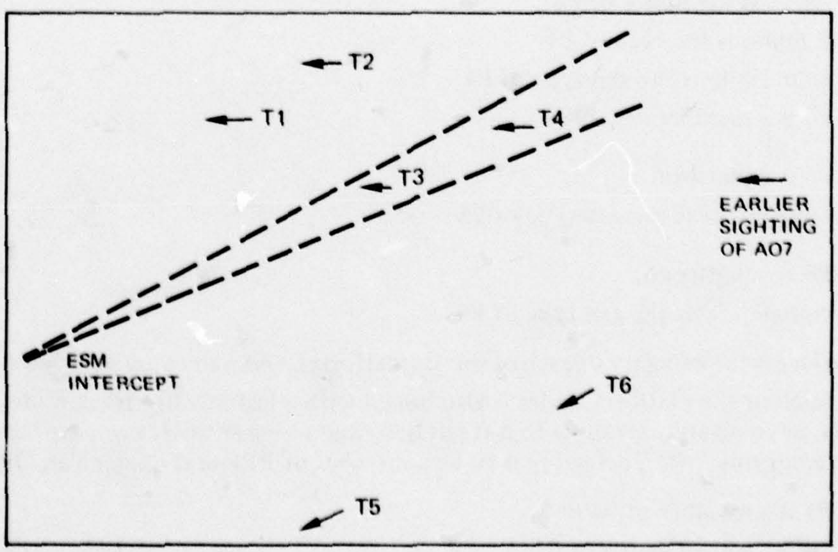



Figure 6. A satellite radar map provides position data on six platforms. Earlier position data and an ESM intercept help to identify some of the platforms.

## DATA BASE PRIOR TO RADAR MAP

The pertinent assertions in the data base at the time of receipt of position data include earlier information about the platforms:

P1 is a platform
CG155 is the hull # of P1
Kara is the class of P1
cruiser is the gen-type of P1
P1 is a member of GPF1

(GPF is a task-group-platform-file.)

P2 is a platform
DDG223 is the hull # of P2
Krivak is the class of P2
destroyer is the gen-type of P2
P2 is a member of GPF1

P3 is a platform
AO7 is the hull # of P3
Kazbek is the class of P3
mil-oiler is the gen-type of P3
P3 is a member of GPF1

P4 is a platform
AE12 is the hull # of P4
Kammo is the class of P4
ammo-ship is the gen-type of P4
P4 is a member of GPF1

P5 is a platform
commercial is the gen-type of P5

P6 is a platform
commercial is the gen-type of P6

Also, surface is the category of each of the six platforms, and names are asserted where known.

Each of the platform nodes is also linked with a last-inactive-track node. For each platform, as computations show that it has had time to enter the region, the "inactive-track RFP member rule" fires, declaring it to be a member of RPF and opening an OR-file for it.

P1 is a member of RPF
ORF1 is the OR-file of P1
P2 is a member of RPF
ORF2 is the OR-file of P2
P3 is a member of RPF

ORF3 is the OR-file of P3

P4 is a member of RPF

ORF4 is the OR-file of P4

Similarly, the commercial ships become members of the region's track file.

P5 is a member of RPF

ORF5 is the OR-file of P5

P6 is a member of RPF

ORF6 is the OR-file of P6

A number of the RPF subset rules fire. Among the pertinent conclusions are the following:

PF1 is a platform file

PF1 is a category-subset of RPF

surface is the category of PF1

P1-P6 are members of PF1

Surveillance in the surrounding areas has been thorough enough that the platform file PF1 is complete.

Complete is the status of PF1

**POSITION DATA FROM RADAR MAP**

The decision process can proceed in any of a number of ways, depending upon the order of the rule accessing and the order in which data are received. One possible decision sequence is outlined below.

After the position data from the radar map are received, each of the two merchants are quickly associated with a track, based on knowledge of its course, speed, and earlier position. Own-force positions are also quickly correlated. The tracks T1-T4 are asserted to be impos-tracks of P5 and P6, by the "impos-track by platform-elim rule." The tracks T5 and T6 are asserted to be impos-tracks of platforms P1-P4 by the "impos-track by track elimination rule." The "OR-file member rule" gives the following:

T1 is a member of ORF1

P1 is a member of FRO1

T2 is a member of ORF1

P2 is a member of FRO1

T3 is a member of ORF1

P3 is a member of FRO1

T4 is a member of ORF1

P4 is a member of FRO1

and equivalent assertions for ORF2 and FRO2, ORF3 and FRO3, ORF4 and FRO4. Also,

T5 is a member of ORF5

P5 is a member of FRO5

T6 is a member of ORF6

P6 is a member of FRO6

The category-subset TF1 of RTF whose category is surface has a status of complete by the "complete surface-track file by map rule" (activated by receipt of satellite data) or, alternatively, by the complete track-file rule. The OR-files (ORF1–ORF6) of platforms P1–P6 therefore have a status of complete from the "complete platform-OR-file rule." The OR-files (FRO1–FRO6) of tracks T1–T4 have a status of complete from the "complete track-OR-file rule." The "task-group within region rule" also fires.

## EARLIER SIGHTING OF THE OILER

The "impos-track by earlier-sighting rule" acts on data from a report of an earlier sighting of the oiler, giving the following:

T1 is an impos-track of P3

T2 is an impos-track of P3

The "OR-file reduction rule" then eliminates T1 and T2 from ORF3 and P3 from FRO1 and FRO2. (If relative platform sizes can also be determined from the satellite map, the oiler may easily be identified.)

## T1 IN LEAD POSITION

**Complete Task-Group-Subset of RTF Rule Fires . . .**

complete is the status of GTF1

**Task-Group Lead-Position Rule Fires . . .**

lead-position is a function of T1

**Impos-Track by Lead-Position Rule Fires . . .**

T1 is an impos-track of P4 [ammo ship]

The OR-file reduction rule then eliminates P4 from FRO1 and T1 from ORF4.

## ESM BEARING

The ESM system provides the type-number of an intercepted signal S1, the latitude and longitude of the sensor position, the intercept bearing, and the bearing accuracy.

**REF Member Rule Fires . . .**

S1 is a member of REF

SORF1 is the OR-file of S1

(The OR-file of an emission will later have as members all platforms that have not been ruled out as the emitter platform.)

**Impos-Emitter by Platform-Class Rule Fires . . .**

P1 is an impos-emitter of S1

P3 is an impos-emitter of S1

P4 is an impos-emitter of S1

**Impos-Emitter by Platform-Gen-Type Rule Fires . . .**

P5 is an impos-emitter of S1

P6 is an impos-emitter of S1

**Emission OR-File Member Rule Fires . . .**

P2 is a member of SORF1

(If the "emission OR-file member rule" fires before the above impos-emitter rules, then the "emission OR-file reduction rule" removes P1 and P3–P6 from SORF1.)

**Complete Emission-OR-File Rule Fires . . .**

complete is the status of SORF1

(The OR-file of signal S1 is complete because the surface subset PF1 is complete and every member of PF1 which is not an impossible emitter of S1 is a member of SORF1.)

**And-Then-There-Was-One Platform-Emitter Rule Fires . . .**

P2 is the platform-emitter of S1

**Impos-Emitter by Bearing Rule Fires . . .**

T1 is an impos-emitter of S1

T2 is an impos-emitter of S1

**Impos-Track by Platform-Emission Association Rule Fires . . .**

T1 is an impos-track of P2

T2 is an impos-track pf P2

(Because signal S1 was emitted by P2, and S1 could not have come from the direction of T1 or T2.)

The OR-file reduction rule then eliminates T1 and T2 from ORF2 and P2 from FRO1 and FRO2.

**And-Then-There-Was-One Platform Rule Fires . . .**

T1 is the track of P1

**Impos-Track by Platform-Elim Rule Fires . . .**

T2 is an impos-track of P1

T3 is an impos-track of P1

T4 is an impos-track of P1

**OR-File Reduction Rule Fires . . .**

T2, T3, and T4 are removed from ORF1

P1 is removed from FRO2, FRO3, and FRO4

**And-Then-There-Was-One Platform Rule Fires . . .**

T2 is a track of P4

**Impos-Track by Platform Rule Fires . . .**

T3 is an impos-track of P4

T4 is an impos-track of P4

**OR-File Reduction Rule Fires . . .**

T3 and T4 are removed from ORF4

P4 is removed from FRO3 and FRO4

**RESULTING DATA BASE**

The conclusions available from the system include the following:

T1 is a track of P1 [CG155]

T2 is a track of P4 [AE12]

The OR-file of P2 [DDG223] is complete and contains T3 and T4

The OR-file of P3 [AO7] is complete and contains T3 and T4

The OR-file of T3 is complete and contains P2 and P3

The OR-file of T4 is complete and contains P2 and P3.

## PRESENT CONSTRAINTS

**SIMPLIFYING ASSUMPTIONS**

The assumption was made in this verison of PTAPS that every track of a US platform (surface ship or submarine) was known to be that of a US platform. When this assumption cannot be satisfied, the rules for membership in RPF and RTF need to be modified. Any track not certain to be that of a US ship must be made a member of the track file RTF of the region it is in, and any US platform whose track could possibly be a member of RTF must be made a member of the region's platform file, RFP. Also, additional removal rules are needed. Tracks later determined to be own-force's should be removed from track files and OR-files, and the platform should be removed from platform files and OR-files. Their assertions having the relationship impos-track should also be removed. These removals are needed only to reduce the number of assertions stored in the data base.

The track files were assumed to contain only tracks of surface ships and submarines, and the problem of distinguishing ship tracks from boat tracks was disregarded. When there is doubt as to whether a surface contact is a ship or a boat (the radar cross section of a corvette can approach that of a frigate, for example), a modified version of the "complete track-file rule" is needed to determine the completeness of files containing that track. PTAPS can be extended to include boats in platform files, but new rules would be needed to deal with situations where, for example, landing craft are launched from or docked at a landing ship dock. Whether or not boats are included in platform files, it would also be useful to have assertions indicating a general range of size for each track, where possible. Overlapping subset files (eg "large/medium ships," "medium/small ships," and "small ships and boats") could be used in finding additional platform-to-ship associations via a complete track-file rule.

## DATA INCONSISTENCIES AND CONTRADICTIONS

A very important assumption already discussed under DATA BASE ORGANIZATION is that no two active tracks can be the track of a single platform. "Active" here has been interpreted to mean "at least recent," and rules as yet undefined are needed to specify when a track should be made inactive.

Incorrect data caused by deceptive measures by the enemy will lead to incorrect conclusions in PTAPS in about the same manner as in human reasoning, provided that contradictory data are discovered and resolved by the system before conclusions are drawn from them. Even without a mechanism for resolving contradictory data, deception should seldom result in misleading conclusions if, whenever a form of deception is suspected as a possibility, the type of data that can be affected by the deception is weighted with a confidence factor less than near certainty. In order to compete with human reasoning, though, the system must have the sophistication to use confidence values in the manner described next.

## EXTENDING A SYSTEM WITH PTAPS

Extending the capabilities of a production system used for tactical situation assessment by incorporating into it a PTAPS should present no major design problems for the simple version of PTAPS described in this report. One obvious step in making PTAPS compatible with other system operations is to change the terminology where necessary for consistency. A more difficult and subtle problem occurs in employing the concepts of a track and a platform uniformly throughout the system. Consistent use of the notions of an active track and an inactive track can be another problem.

As it is described in this report, the PTAPS data base contains only data having very high confidence values; confidence values are not used in the reasoning process. However, by recording conclusions based on less certain data and then reinitializing the data base by replacing selected questionable assertions with alternative assertions, the conclusions which would logically follow from different assumptions about particular tracks or platforms could be automatically determined. The confidence in PTAPS conclusions, as a whole, would be the joint confidence value of the initial data, as calculated by some appropriate weighting formula. If confidence values for individual conclusions are desired, then, in addition to having the capability of reinitializing the PTAPS data base, the system must store the confidence of each of the assertions in it. (Many production systems have this capability.) Most of the assertions will be PTAPS conclusions, and while the weighting formulas used in the non-PTAPS part of the system may be satisfactory for computing many of their confidences, a few special formulas would have to be developed. Computing confidences of conclusions from "and-then-there-was-one" rules, for example, would require a weighting formula that can assign a confidence to a member-count computation by using the confidences of individual membership in the particular file.

In a production system that tries to fire a rule only where there is new information pertinent to a condition of the rule, there could be difficulty with rules where the new information is a value of the function, such as a change in the member-count of an OR-file, rather than a new assertion.

## COMPUTATIONAL REQUIREMENTS

Experiments based on the two preceding scenarios were run in INTERLISP at USC-ISIC on the ARPANET. For both, the experiments began with the data base containing assertions representing a "snapshot" of the situation just before track information is received. Except at times of heavy computer usage, the individual runs proceeded at a tolerable rate. While no computer limitations other than occasional slowness were encountered in these experiments, a more comprehensive system having a complete PTAPS embedded in it would probably at least tax the capabilities of most existing computer systems.

Only the simpler of the geometric functions involved in evaluating rule conditions were programmed. The other functions can be implemented without serious difficulty, but including them in these experiments would not serve a purpose relative to the intent of the investigations and would increase execution-time aggravations. Some of the geometric functions used in the rules given are currently implemented in STAMMER (ref 1) and STAMMER2 (ref 8). In an operational system, the geometric function evaluation should be performed in a language more effecient for the purpose than LISP, and an interface of that language with LISP would then be needed.

## MULTIPLE REGIONS

The production system method described in this document deals with tracks, platforms, and emitters in a defined geographical area referred to as a "region." In the first example given, the region was a mostly enclosed area such as the Persian Gulf; in the second, it was an area of open ocean. In open-sea cases, the regions can be permanently defined or can be changed as satellite surveillance paths change or as a hostile task force progresses. In open-sea cases, furthermore, it may be practical to somewhat overlap adjacent regions.

In some situations the production system for a region should reside in a computer located within or close to the region, either shipboard or shore-based; but the system might best be a remote one if the region is in open seas and under satellite surveillance. Having several adjoining regions time-share a production system in a single high-speed computer would be advantageous in providing common storage of the rules and of the functions used for rule evaluation, network management, and geometric calculations. Each region would have a separate data base, although there might be an efficient way to give each region access to all platform nodes and the assertions about those platforms. Adjoining regions often will both need the same platform node when the location of the platform is uncertain, and a platform node will need to be transferred from the platform file RPF of one region to an adjoining region when its track is handed over. Also, platform nodes need to be shared if regions overlap.

## ADDITIONAL KINDS OF LOGIC

The rules given or outlined in this document are probably well short of those needed for solving all logical problems that might occur. An example of the need for additional reasoning capability is given below.

If the OR-file of the tracks T1, T2, and T3 are complete and are, respectively (P1 P2), (P1 P2), and (P1 P2 P3), the rules given and tested will quickly deduce that T3 is the track of P3. This deduction is possible because the data base will also have that the OR-file of P3 is (T1), and the and-then-there-was-one track rule applies. A more difficult problem occurs when the OR-files of T1, T2, T3, and T4 are, respectively, (P1 P2), (P1 P2), (P1 P2 P3 P4), and (P1 P2 P3 P4). Rules not presently formulated are needed to deduce that the OR-files of T3 and T4 can both be reduced to (P3 P4). The reasoning in general would be as follows: If the OR-files of N tracks are complete, are identical, and have N numbers, then those N tracks are impossible tracks of all other platforms. If this reasoning were implemented, an existing rule would remove P1 and P2 from the OR-files of T3 and T4. An equivalent rule or set of rules is needed for OR-files of N platforms.

## SUMMARY AND CONCLUSIONS

A method is described of using production rules to perform much of the higher order logical reasoning needed to associate specific platforms with tracks. In practical applications, the rules and data base involved in this logical process would reside as a subsystem in a very large production system, one containing many rules for tactical situation assessment. The application of this method in this manner has been termed PTAPS, for Platform-Track Association Production Subsystem. In practice, there would be no clear line of demarcation between PTAPS and the remainder of the system, since an unextended system would contain much of the data and a number of the rules needed by PTAPS (fig 1).

Many of the PTAPS rules were exercised experimentally, as described under PRESENT CONSTRAINTS – Computational Requirements. The rule set was augmented and refined through experimentation to the extent that the logical reasoning proceeded as intended, reaching correct conclusions. The experimentation included varying the order in which messages are received and the order in which the rules are sequenced. Because the two scenarios underlying the experiments were designed to test the method efficiently, they are not representative of more typical real situations where the available information will support relatively few platform-track associations.

Further research is needed to find solutions to the interface problems involved in extending with PTAPS a production system applied to tactical situation assessment. For interfacing PTAPS with STAMMER2 (ref 8), for example, it would be desirable to employ STAMMER2's mechanism for computing confidence values, as well as its explanation mechanisms of derivation tracing and retrieving memory contents. Actual implementation of a PTAPS in a production system such as STAMMER2 is recommended as an exploratory development task when these and compatibility problems of interfacing are solved and adequate computing resources become available.

Developing this method of performing logical reasoning has been just one phase of a larger effort to develop automated *data-fusion techniques.* The automation of data fusion will require the integration of many interacting subprocesses (ref 7). Probably the two most important of the applicable technologies are production systems and natural language processing. Much of the data to be fused is textual material, and the pertinent textual information must be converted into an assertional form acceptable by a production system. A problem now being addressed in a subtask of this project is the processing of natural language comments on tactical messages; methods of using the formatted part of the message to help understand the unformatted part are being investigated. When a satisfactory technique has been developed for solving some of the simpler problems of processing natural language tactical data, the technique should be interfaced with a production system in a small-scale experimental model of a *data-fusion system.* Investigation of individual data-fusion techniques should continue, and the more promising ones should be integrated into the experimental model of a data-fusion system. This experimental model can be used to find the interactions among the various processes, leading to the optimum design and integration of automated data-fusion processes.

## REFERENCES


1. NOSC TD 252, STAMMER: System for Tactical Assessment of Multisource Messages, Even Radar, by RJ Bechtel and PH Morris, of Systems Development Corporation, May 1979.

2. Final Technical Report on TECA Development, SDC Integrated Services, Inc, 29 December 1978.

3. Artificial Intelligence, by PH Winston, Addison-Wesley, 1977.

4. Stanford AI Lab Memo AIM-270, Computer Science Dept Report STAN-CS-75-524, An Overview of Production Systems, by R Davis and J King, October 1975.

5. Stanford AI Lab Memo AIM-266, Computer Science Dept Report STAN-CS-75-519, Production Rules as a Representation for a Knowledge-Based Consultation System, By R Davis, BG Buchanan, and EH Shortliffe, October 1975.

6. Computer-Based Medical Consultations: MYCIN, by EH Shortliffe, American Elsevier, 1976.

7. NOSC TR 364, New Methodologies for Automated Data Fusion Processing, by RA Dillard, September 1978.

8. STAMMER2. A Production System for Tactical Situation Assessment, by P Morris, D Kibler and RJ Bechtel, SDC Integrated Services, Inc, August 1979. (Also submitted for publication as an NOSC Technical Document.)

## APPENDIX A: BASIC PLATFORM/TRACK RULES

### REGION'S PLATFORM FILE RULES

(The platform node p along with its track t (or last-inactive-track) can be "handed over" from an adjoining region, or p can be created by the "platform-node creation by impos-tracks rule" given later.)

#### Active-Track RPF Member Rule

If *p is a platform

& unless p is a member of own force

& if *t is the track of p

& if (inside-region t)

& unless p is a member of RPF

& if *n: (next-node#)

then p is a member of RPF

& n is the OR-file of p

(The asterisk indicates that the variable is bound to the "retrieval." In some cases there will be a number of answers retrieved, and each of these possibilities is carried forward. A full explanation of rule interpretation is given in reference A1. A colon indicates that the asterisked variable is bound to the value of the function following it.)

- The function **(inside-region** t) determines if the most recent position associated with track t is inside the region. In a simple case, it would determine if a point lies within a polygon on a spherical earth. The value of the function is either true or nil.

- The function **(next-node#)** produces an identifying name or number. In the experimental system, the Interlisp function gensym[F] was used.

#### Inactive-Track RPF Member Rule

If *p is a platform

& unless P is a member of RPF

& unless p is a member of own force

& unless *t is the track of p

& if *tt is the last-inactive-track of p

& if (could-reach-region p tt)

& if *n: (next-node#)

then p is a member of RPF

& uncertain is the status of p

& n is the OR-file of p

---

A1. NOSC TD 252, STAMMER: System for Tactical Assessment of Multisource Messages, Even Radar, by RJ Bechtel and PH Morris, of Systems Development Corporation, May 1979.

• The function (**could-reach-region** p tt) uses the most recent position associated with an inactive track tt and the velocity of the platform p to determine if p could possibly have entered the region.

Platform-file subsets are built by the next eight rules. Depending upon the rule-evaluation mechanism of the production system, it could be much more efficient to permanently establish a surface platform subset file and a subsurface platform subset file rather than to remove them when empty and recreate them (by the first rule below) when needed.

**Platform-Category Node Creation Rule**

If *p is a member of RPF
& if *ctg is the category of p
& unless *PF is a category-subset of RPF & ctg is the category of PF
& if *n: (next-node#)
then n is a category-subset of RPF
& ctg is the category of n
& n is a platform-file

**Platform-Category Member Rule**

If *p is a member of RPF
& if *ctg is the category of p
& if *PF is a category-subset of RPF
& if ctg is the category of PF
& unless p is a member of PF
then p is a member of PF

**Platform-General-Type Node Creation Rule**

If *p is a member of RPF
& if *gty is the gen-type of p
& unless *PF is a gen-type-subset of RPF & gty is the gen-type of PF
& if *n: (next-node#)
then n is a gen-type-subset of RPF
& gty is the gen-type of n
& n is a platform-file

**Platform-General-Type Member Rule**

If *p is a member of RPF
& if *gty is the gen-type of p
& if *PF is a gen-type-subset of RPF

& if gty is the gen-type of PF
& unless p is a member of PF
then p is a member of PF

**Platform-Type Node Creation Rule**

If *p is a member of RPF
& if *ty is the type of p
& unless *PF is a type-subset of RPF & ty is the type of PF
& if *n: (next-node#)
then n is a type-subset of RPF
& ty is the type of n
& n is a platform-file

**Platform-Type Member Rule**

If *p is a member of RPF
& if *ty is the type of p
& if *PF is a type-subset of RPF
& if ty is the type of PF
& unless p is a member of PF
then p is a member of PF

**Platform-Class Node Creation Rule**

If *p is a member of RPF
& if *cls is the class of p
& unless *PF is a class-subset of RPF & cls is the class of PF
& if *n: (next-node#)
then n is a class-subset of RPF
& cls is the class of n
& n is a platform-file

**Platform-Class Member Rule**

If *p is a member of RPF
& if *cls is the class of p
& if *PF is a class-subset of RPF
& if cls is the class of PF
& unless p is a member of PF
then p is a member of PF

## REGION'S TRACK FILE RULES

### RTF Member Rule

If *t is a track
& unless t is a member of own-track-file
& if (inside-region t)
& unless t is a member of RTF
& if *n: (next-node#)
then t is a member of RTF
and n is the OR-file of t

### "Working" Relations

If the type of a contact has been determined, even though the track has not been associated with a specific platform in the platform file, this observation is expressed as

⟨type⟩ is the working-type of T0046

Category, general type, type, and class are also expressed as working relations, eg "working-class." If the category, general type, type, or class of a track is known because it has been associated with a particular platform, the information is attached to the track node via one of the following rules.

### Working-Category Rule

If *t is a member of RTF
& if t is the track of *p
& if *ctg is the category of p
& unless ctg is the working-category of t
then ctg is the working-category of t

### Working-General-Type Rule

If *t is a member of RTF
& if t is the track of *p
& if *gty is the gen-type of p
& unless gty is the working-gen-type of p
then gty is the working-gen-type of t

### Working-Type Rule

If *t is a member of RTF
& if t is the track of *p
& if *ty is the type of p

& unless ty is the working-type of t

then ty is the working-type of t

**Working-Class Rule**

If *t is a member of RTF

& if t is the track of *p

& if *cls is the class of p

& unless cls is the working-class of t

then cls is the working-class of t

When a new observation is made about a track t, and t is the track of p, and the information is not already attached to the platform p, it is then attached to p via a set of rules similar to those above.

Track-file subsets are built by the next eight rules.

**Track-Category Node Creation Rule**

If *t is a member of RTF

& if *ctg is the working-category of t

& unless *TF is a category-subset of RTF & ctg is the category of TF

& if *n: (next-node#)

then n is a category-subset of RTF

& ctg is the category of n

& n is a track-file

**Track-Category Member Rule**

If *t is a member of RTF

& if *ctg is the working-category of t

& if *TF is a category-subset of RTF

& if ctg is the category of TF

& unless t is a member of TF

then t is a member of TF

**Track-General-Type Node Creation Rule**

If *t is a member of RTF

& if *gty is the working-gen-type of t

& unless *TF is a gen-type-subset of RTF & gty is the gen-type of TF

& if *n: (next-node#)

then n is a gen-type-subset of RTF

& gty is the gen-type of n

& n is a track-file

**Track-General-Type Member Rule**

If *t is a member of RTF
& if *gty is the gen-working-type of t
& if *TF is a gen-type-subset of RTF
& if gty is the gen-type of TF
& unless t is a member of TF
then t is a member of TF

**Track-Type Node Creation Rule**

If *t is a member of RTF
& if *ty is the working-type of t
& unless *TF is a type-subset of RTF & ty is the type of TF
& if *n: (next-node#)
then n is a type-subset of RTF
& ty is the type of n
& n is a track-file

**Track-Type Member Rule**

If *t is a member of RTF
& if *ty is the working-type of t
& if *TF is a type-subset of RTF
& if ty is the type of TF
& unless t is a member of TF
then t is a member of TF

**Track-Class Node Creation Rule**

If *t is a member of RTF
& if *cls is the working-class of t
& unless *TF is a class-subset of RTF & cls is the class of TF
& if *n: (next-node#)
then n is a class-subset of RTF
& ty is the type of n
& n is a track-file

**Track-Class Member Rule**

If *t is a member of RTF
& if *cls is the working-class of t

& if *TF is a class-subset of RTF
& if cls is the class of TF
& unless t is a member of TF
then t is a member of TF

**Removal Rules**

Another set of rules removes platform nodes from RPF and its subsets and removes track nodes from RTF and its subsets when they are known to be outside the region. Subset nodes are removed when their last member is removed. A track t is also removed from RTF if it becomes an inactive track. The status of complete is removed from a track file if that file loses a member track because the track becomes inactive.

## IMPOSSIBLE-TRACK RULES

**Impos-Track by Track-Elim Rule**

If *t is a member of RTF
& if t is the track of *ap
& if *p is a member of RPF
& unless t is an impos-track of p
& unless (eq p ap)
then t is an impos-track of p

- The function (**eq** x y) has the value true if x and y are identical and the value nil otherwise.

**Impos-Track by Platform-Elim Rule**

If *p is a member of RPF
& if *at is the track of p
& if *t is a member of RTF
& unless t is an impos-track of p
& unless (eq t at)
then t is an impos-track of p

**Impos-Track by Category Rule**

If *t is a member of RTF
& if *p is a member of RPF
& unless t is an impos-track of p
& if *ctg is the working-category of t
& if *c is the category of p
& unless (eq c ctg)
then t is an impos-track of p

**Impos-Track by General-Type Rule**

If *t is a member of RTF
& if *p is a member of RPF
& unless t is an impos-track of p
& if *gty is the working-gen-type of t
& if *y is the gen-type of p
& unless (eq y gty)
then t is an impos-track of p

**Impos-Track by Type Rule**

If *t is a member of RTF
& if *p is a member of RPF
& unless t is an impos-track of p
& if *ty is the working-type of t
& if *y is the type of p
& unless (eq y ty)
then t is an impos-track of p

**Impos-Track by Class Rule**

If *t is a member of RTF
& if *p is a member of RPF
& unless t is an impos-track of p
& if *cls is the working-class of t
& if *c is the class of p
& unless (eq c cls)
then t is an impos-track of p

**Impos-Track by Acoustic-Data Rule**

If *t is a member of RTF
& if subsurface is the working-category of t
& if *ad is the acoustic-data of t
& if *icf is the impos-class-file of ad
& if *p is a member of RPF
& if *cls is the class of p
& if cls is a member of icf
& unless t is an impos-track of p
then t is an impos-track of p

**Impos-Track by Earlier-Sighting Rule**

If *p is a member of RPF
& if *t is a member of RTF
& unless t is an impos-track of p
& unless t is the track of p
& if *tt is the last-inactive-track of p
& if (impos-speed p tt t)
then t is an impos-track of p

- The function (**impos-speed** p tt t) uses the positions and times associated with inactive track tt and active track t to determine if the velocity required to transit is greater than the maximum velocity of platform p.

**Platform-Node Creation by Impos-Tracks Rule**

If *t is a member of RTF
& if [p is a member of RPF ⇒ p an impos-track of t]
& if *n: (next-node#)
then n is a platform
& t is the track of n

**CORRESPONDING-FILE RULES**

**Corresponding File (Category) Rule**

If *PF is a category-subset of RPF
& if *ctg is the category of PF
& if *TF is a category-subset of RTF
& if ctg is the category of TF
& unless TF is the corresponding-file of PF
then TF is the corresponding-file of PF

**Corresponding File (Gen-Type) Rule**

If *PF is a gen-type-subset of RPF
& if *gty is the gen-type of PF
& if *TF is a gen-type-subset of RTF
& if gty is the gen-type of TF
& unless TF is the corresponding-file of PF
then TF is the corresponding-file of PF

**Corresponding File (Type) Rule**

If *PF is a type-subset of RPF
& if *ty is the type of PF
& if *TF is a type-subset of RTF
& if ty is the type of TF
& unless TF is the corresponding-file of PF
then TF is the corresponding-file of PF

**Corresponding File (Class) Rule**

If *PF is a class-subset of RPF
& if *cls is the class of PF
& if *TF is a class-subset of RTF
& if ty is the class of TF
& unless TF is the corresponding-file of PF
then TF is the corresponding-file of PF

**COMPLETE FILE RULES**

In continually monitored regions, a rule such as the following could be kept activated. It is needed only when a subsurface platform file has been created and is unnecessary if category files are permanently established. A similar rule would exist for surface platforms.

**Always-Complete Subsurface-Platform File Rule**

If *PF is a category-subset of RPF
& if subsurface is the category of PF
& unless complete is the status of PF
then complete is the status of PF

**Complete Surface-Track File by Map Rule**

If *TF is a category-subset of RTF
& if surface is the category of TF
& if (mapflag)
& unless complete is the status of TF
then complete is the status of TF

- **(mapflag)** is a pseudo function which has the value true in the event of recent receipt of position data from satellite reconnaissance covering the region.

**Complete Track-File Rule**

If *PF is a platform-file
& if complete is the status of PF
& if *TF is the corresponding-file of PF
& if (member-count PF) = (member-count TF)
& unless complete is the status of TF
then complete is the status of TF

- The function (**member-count** n) counts the number of nodes m such that "m is a member of n" is an assertion.

## OR-FILE RULES

The members of the OR-file of a platform will be those tracks which have not yet been ruled out as the track of that platform. The members of the OR-file of a track will be those platforms that have not been ruled out for that track.

**OR-File Member Rule**

If *p is a member of RPF
& if *t is a member of RTF
& unless t is an impos-track of p
& if *orf is the OR-file of p
& if *fro is the OR-file of t
& unless t is a member of orf
then t is a member of orf
& p is a member of fro

**OR-File Reduction Rule**

If *p is a member of RPF
& if *orf is the OR-file of p
& if *t is a member of orf
& if t is an impos-track of p
& if *fro is the OR-file of t
then erase: t is a member of orf
& erase: p is a member of fro

**Complete Platform-OR-File Rule**

If *TF is a track-file
& if complete is the status of TF
& if TF is the corresponding-file of *PF

& if *p is a member of PF
& if *orf is the OR-file of p
& unless complete is the status of orf
then complete is the status of orf

**Complete-Track-OR-File Rule**

If *PF is a platform-file
& if complete is the status of PF
& if *TF is the corresponding-file of PF
& if *t is a member of TF
& if *fro is the OR-file of t
& unless complete is the status of fro
then complete is the status of fro

**Complete Associated-Track OR-File Rule**

If *t is a member of RTF
& if t is the track of *p
& if *fro is the OR-file of t
& unless complete is the status of fro
& if p is a member of fro
then complete is the status of fro

**Complete Associated-Platform OR-File Rule**

If *p is a member of RPF
& if *t is the track of p
& if *orf is the OR-file of p
& unless complete is the status of orf
& if t is a member of orf
then complete is the status of orf

**And-Then-There-Was-One Track Rule**

If *p is a member of RPF
& if *orf is the OR-file of p
& if complete is the status of orf
& if (member-count orf) = 1
& if *t is the member of orf
& unless t is the track of p
then t is the track of p

**And-Then-There-Was-One Platform Rule**

If *t is a member of RTF
& if *fro is the OR-file of t
& if complete is the status of fro
& if (member-count fro) = 1
& if *p is the member of fro
& unless t is the track of p
then t is the track of p

## APPENDIX B: EMISSION RULES

### REGION'S EMISSION-FILE MEMBER RULE

**REF Member Rule**

If *s is an emission
& if (emitter-inside-regions s)
& if *k is the emitter-type of s
& unless s is a member of own-emitter-file
& unless s is a member of REF
& if *n: (next-node#)
then s is a member of REF
& n is the OR-file of s

• The function **(emitter-inside-region** s) uses the position data (if available) or bearing data (otherwise) associated with the intercepted signal s to determine if the emitter is within the region. Unless a method of using confidence values has been built into PTAPS, the value is nil if there is doubt.

### IMPOSSIBLE-EMITTER RULES

**Impos-Emitter by Bearing Rule**

If *s is a member of REF
& if *t is a member of RTF
& unless t is an impos-emitter of s
& if *b is the bearing-data of s
& if *l is the position-data of t
& unless (bearing-consistent l b)
then t is an impos-emitter of s

• The function **(bearing-consistent** l b) uses the position data l of a track and the bearing data b of a signal to determine if the signal could have been emitted from that track location. The value is true if possible and nil if impossible.

**Impos-Emitter by Call-Sign Rule**

If *s is a member of REF
& if *c is the call-sign of s
& if *p is a member of RPF
& if (impos-call-sign c p)
then p is an impos-emitter of s

• The function (**impos-call-sign** c p) looks for the most identifying information (ie name, hull #, class, type, general-type, or category) about platform p and compares it with the corresponding classification of the platform having call sign c.

**Impos-Emitter by Track-Class Rule**

If *s is a member of REF
& if *t is a member of RTF
& unless t is an impos-emitter of s
& if *cl is the working-class of t
& if *k is the emitter-type of *s
& if *corf is the class-OR-file of k
& unless cl is a member of corf
then t is an impos-emitter of s

**Impos-Emitter by Track-Type Rule**

If *s is a member of REF
& if *t is a member of RTF
& unless t is an impos-emitter of s
& unless *cl is the working-class of t
& if *ty is the working-type of t
& if *k is the emitter type of s
& if *torf is the type-OR-file of k
& unless ty is a member of torf
then t is an impos-emitter of s

**Impos-Emitter by Track-Gen-Type Rule**

If *s is a member of REF
& if *t is a member of RTF
& unless t is an impos-emitter of s
& unless *cl is the working-class of t
& unless *ty is the working-type of t
& if *gty is the working-gen-type of t
& if k is the emitter-type of *s
& if *gorf is the gen-type-OR-file of k
& unless gty is a member of gorf
then t is an impos-emitter of s

**Impos-Emitter by Platform-Class Rule**

If *s is a member of REF
& if *p is a member of RPF
& unless p is an impos-emitter of s
& if *cl is the class of p
& if *k is the emitter-type of s
& if *corf is the class-OR-file of k
& unless cl is a member of corf
then p is an impos-emitter of s

**Impos-Emitter by Platform-Type Rule**

If *s is the member of REF
& if *p is a member of RPF
& unless p is an impos-emitter of s
& unless *cl is the class of p
& if *ty is the type of p
& if *k is the emitter-type of *s
& if *torf is the type-OR-file of k
& unless ty is a member of torf
then p is an impos-emitter of s

**Impos-Emitter by Platform Gen-Type Rule**

If *s is a member of REF
& if *p is a member of RPF
& unless p is an impos-emitter of s
& unless *cl is the class of p
& unless *ty is the type of p
& if *gty is the gen-type of p
& if *k is the emitter-type of s
& if *gorf is the gen-type-OR-file of k
& unless gty is a member of gorf
then p is an impos-emitter of s

**Impos-Emitter by OR-File of Track Rule**

If *s is a member of REF
& if *t is a member of RTF
& unless t is an impos-emitter of s

& if *fro is the OR-file of t

& if complete is the status of fro

& if [*p a member of fro ⇒ p an impos-emitter of s]

(ie all platforms in fro are impos-emitters of the signal)

then t is an impos-emitter of s

**Impos-Emitter by OR-File of Platform Rule**

If *s is a member of REF

& if *p is a member of RPF

& unless p is an impos-emitter of s

& if *orf is the OR-file of p

& if complete is the status of orf

& if [*t a member of orf ⇒ t an impos-emitter of s]

then p is an impos-emitter of s

**Impos-Emitter by Track-Elim Rule**

If *s is a member of REF

& if *t is the track-emitter of s

& if *tt is a member of RTF

& unless (eq t tt)

then tt is an impos-emitter of s

**Impos-Emitter by Platform-Elim Rule**

If *s is a member of REF

& if *p is the platform-emitter of s

& if *pp is a member of RPF

& unless pp is an impos-emitter of s

& unless (eq p pp)

then pp is an impos-emitter of s

## EMISSION OR-FILE RULES

**Emission OR-File Member Rule**

If *s is a member of REF

& if *p is a member of RPF

& unless p is an impos-emitter of s

& if *sorf is the OR-file of s

& unless p is a member of sorf

then p is a member of sorf

**Emission OR-File Reduction Rule**

If *s is a member of REF
& if *sorf is the OR-file of s
& if *p is a member of sorf
& if p is an impos-emitter of s
then erase: p is a member of sorf

**Complete Emission-OR-File Rule**

If *s is a member of REF
& if *PF is a category-subset of RPF
& if surface is the category of PF
& if complete is the status of PF
& if *k is the emitter-type of s
& if *gorf is the gen-type-OR-file of k
& unless sub is a member of gorf
& if *sorf is the OR-file of s
& unless complete is the status of sorf
then complete is the status of sorf

**And-Then-There-Was-One Platform-Emitter Rule**

If *s is a member of REF
& if *sorf is the OR-file of s
& if complete is the status of sorf
& if (member-count sorf) = 1
& if *p is the member of sorf
& unless p is the platform-emitter of s
then p is the platform-emitter of s

**ASSOCIATION RULES**

**Platform-Emission Association by Call-Sign Rule**

If *s is a member of REF
& if *c is the call-sign of s
& if *p is a member of RPF
& if c is the call-sign of p
[& unless call-sign-deception is a member of region-state]
& unless p is the platform-emitter of s
then p is the platform-emitter of s

**Platform-Emission Association by Track Rule**

If *s is a member of REF
& if *t is the track-emitter of s
& if t is the track of *p
& unless p is the platform-emitter of s
then p is the platform-emitter of s

**Track-Emission Association by Platform Rule**

If *s is a member of REF
& if *p is the platform-emitter of s
& if *t is the track of p
& unless t is the track-emitter of s
then t is the track-emitter of s

**Alternative Track-Emission Association by Platform Rule**

If *p is a member of RPF
& if p is the platform-emitter of *s
& if *t is the track of p
& unless t is the track-emitter of s
then t is the track-emitter of s

**Association by Emission Rule**

If *s is a member of REF
& if *p is the platform-emitter of s
& if *t is the track-emitter of s
& unless t is the track of p
then t is the track of p

**Platform-Emission Association by OR-File Rule**

If *s is a member of REF
& if *sorf is the OR-file of s
& if complete is the status of sorf
& if (member-count sorf) = 1
& if *p is the member of sorf
& unless p is the platform-emitter of s
then p is the platform-emitter of s

**Impos-Track by Platform-Emission Association Rule**

If *s is a member of REF
& if *p is the platform-emitter of s
& if *t is a member of RTF
& unless t is an impos-track of p
& if t is an impos-emitter of s
then t is an impos-track of p

## APPENDIX C: TASK-GROUP RULES

The initial data about a Soviet task group (or force) are either handed over from another region or entered from intelligence reports, although some could have been deduced from other data by the production system. Typically the data might be as follows:

G00020 is a task-group-platform-file

P00320 is a member of G00020

•

•

•

P00496 is a member of G00020

The platform nodes have descriptive data attached to them. If every platform in the task group is a member of the task-group file, this is expressed as follows:

complete is the status of G00020

The rules below are especially structured for the kind of situation in which a number of tracks are derived from high-altitude reconnaissance data. The system is able to associate some of the tracks with specific platforms not in the task group and to deduce that some of the other tracks cannot be those of the task group. The remainder become members of a special track file built by the first two rules below, which interact with the rules given earlier. The other rules below can be used to eliminate some of the possible associations.

### FILE RULES

**Task-Group-Subset of RPF Rule**

If *GPF is a task-group-platform-file

& unless GPF is a task-group-subset of RPF

& if *p is a member of GPF

& if p is a member of RPF

then GPF is a task-group-subset of RPF

(The subset is created when it is first discovered that a task-group platform is inside the region.)

**RTF Task-Group-Subset Node Creation Rule**

If *GPF is a task-group-subset of RPF

& unless *GTF is the corresponding-file of GPF

& if *n: (next-node#)

then n is a task-group-subset of RTF

& n is a task-group-track-file

& n is the corresponding-file of GPF

**Task-Group-Subset Member Rule**

If *GPF is a task-group-subset of RPF
& if *GTF is the corresponding-file of GPF
& if *p is a member of GPF
& if *orf is the OR-file of p
& if *t is a member of orf
& unless t is a member of GTF
then t is a member of GTF

(Note that members of GTF are members of RTF.)

**Task-Group-Subset Reduction Rule**

If *GTF is a task-group-subset of RTF
& if GTF is the corresponding-file of *GPF
& if *t is a member of GTF
& if [p a member of GPF ⇒ t is an impos-track of p]
then remove: t is a member of GTF

**GEOMETRY-RELATED RULES**

Depending on factors such as the size of the region and the type of task group, the system should be able to determine if it can be safely assumed that every platform is inside the region. The next rule is an example of one of several rules for doing this.

**Task-Group Within Region Rule**

If *GTF is a task-group-subset of RTF
& if *t is a member of GTF
& if (far-inside-region t)
& unless inside-region if GTF
then inside-region is GTF

- The function (**far-inside-region** t) takes the position associated with track t and compares its minimum distance from the edge of the region with a constant.

**Complete Task-Group Subset of RTF Rule**

If *GTF is a task-group-subset of RTF
& if inside-region is GTF
& if *TF is a category-subset of RTF
& if surface is the category of TF

& if complete is the status of TF

& unless complete is the status of GTF

then complete is the status of GTF

(The notion of complete is different than for other track files, since GTF can also contain tracks of platforms not in the task group.)

**Task-Group Lead-Position Rule**

If *GTF is a task-group-subset of RTF

& if complete is the status of GTF

& if *t: (lead-track GTF)

& if *fro is the OR-file of t

& if complete is the status of fro

& if GTF is the corresponding-file of *GPF

& if [p a member of fro ⇒ p a member of GPF]

& unless lead-position is a function of t

then lead-position is a function of t

- The function (**lead-track** GTF) has as a value either a member of GTF or, if none is significantly in the lead, nil. If a course has not already been asserted for GTF, one must be estimated from the courses of the members of GTF.

**Impos-Track by Lead-Position Rule**

If *GTF is a task-group-subset of RTF

& if *t is a member of GTF

& if lead-position is a function of t

& if *p is a member of RPF

& unless t is an impos-track of p

& if carrier is the gen-type of p

or mil-oiler is the gen-type of p

or ammunition is the gen-type of p

then t is an impos-track of p

The next two rules are best suited for use in a production system having a weighting mechanism for accumulative-evidence reasoning.

**Outlier Rule**

If *GTF is a task-group-subset of RTF

& if complete is the status of RTF

& if GTF is the corresponding file of *GPF

& if complete is the status of GPF
& if *m: (member-count GPF)
& if (member-count GTF) > m but < 2m
& if *c: (centroid GTF)
& if *v: (dispersion c GTF)
& if (*t *d): (max-distant-track GTF c)
& unless [*fro is the OR-file of t
    & complete is the status of fro
    & [p a member of fro ⇒ p a member of GPF] ]
& if d/v > constant
& unless t is an unlikely-member of GTF
then t is an unlikely-member of GTF

- The function (**centroid** GTF) yields a lat-lon pair which is the "center" of the positions of the members of GTF. Probably it would average their respective latitudes and longitudes.
- The function (**dispersion** c GTF) computes a measure of scatter based on the distance of the members of GTF from the centroid c.
- The function (**max-distant-track** GTF c) selects the member of GTF having the greatest distance from the centroid c and returns it and its distance.

**Unlikely-Track by Task-Group Outlier Rule**

If *GTF is a task-group-subset of RTF
& if *t is an unlikely-member of GTF
& if GTF is the corresponding-file of *GPF
& if p is a member of GPF
& unless t is an unlikely-track of p
then t is an unlikely-track of p