NOSC TD 632

NOSC TD 632

Technical Document 632

REPRESENTATION OF TACTICAL KNOWLEDGE SHARED BY EXPERT SYSTEMS

R. A. Dillard

1 October 1983
Final Report

Prepared for
Naval Electronic Systems Command

NOSC

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152

NAVAL OCEAN SYSTEMS CENTER SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

J.M. PATTON, CAPT, USN
Commander

R.M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

Work was performed under program element 61153N, project XR01408, task area XR0140801 (NOSC 823-CC25), by a member of the C2 Decision Support Technology Branch (NOSC Code 8233) for Naval Electronic Systems Command, Code 613. This document covers work of FY83.

Note: The software for ROSIE was licensed from the Rand Corporation. ROSIE is a registered trademark of the Rand Corporation.

Released by
G.R. Allgaier, Head
Advanced Command Center
Technology Division

Under authority of
J.H. Maynard, Head
Command Control-Electronic
Warfare Systems & 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 632 (TD 632)	2. GOVT ACCESSION NO. AD-A130875	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) REPRESENTATION OF TACTICAL KNOWLEDGE SHARED BY EXPERT SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED Final Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R A. 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 61153N, XR01408, XR0140801 (NOSC 823-CC25)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command Code 613 Washington, DC		12. REPORT DATE 1 October 1983 13. NUMBER OF PAGES 32
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

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

18. SUPPLEMENTARY NOTES

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

Artificial intelligence
Data fusion
Expert systems
Rule-based systems
Query systems

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

This report describes new progress in the design of data fusion techniques. Emphasis in the latest work is on the sharing of knowledge by cooperating subsystems of a C3 system and the representation of complex concepts. Methods of interfacing and integrating fusion processes with planning, question-answering and historical reconstruction/analysis processes are discussed. Other issues addressed are the subdivision of memory for different functions and user-assisted fusion.

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

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BLANK PAGE

S/N 0102- LF- 014- 6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

INTRODUCTION

This report describes new progress in the design of data fusion techniques. Previous work (reference 1) resulted in the experimental modeling of an automated system. Artificial intelligence techniques included in the model were natural language processi (NLP), rule-based updating of the system database, and rulebased inferencing for tactical data fusion. Emphasis in the latest work is on the sharing of knowledge by cooperating subsystems of a C3 system and the representation of complex concepts. Other issues addressed are the subdivision of memory for different functions and user-assisted fusion.

SYSTEM OVERVIEW

Figure 1 shows the component rulesets (procedural knowledge) and memories (descriptive knowledge) of a rule-based system that can perform automated and user-assisted fusion and support extensive querying capabilities. (The architecture is shown primarily as a ROSIE implementation; other systems would have moderate variations.) Examples of event updating, data retiring, and data fusion appear in references 1 and 2.

Constraining fusion rules to operate only on a relatively small part of the database as shown in figure 1 is being investigated as a method of increasing efficiency in rule evaluation. Most data will not be directly pertinent to the conditions of any fusion rules but will be of potential interest to the user. A query-only memory would include details of plans and activities (e.g., responsibilities, crews, circuits, buoy patterns) and free-form comments. Examples of free-form comments on a report or event are "identity of Barsuk and Kobchik may be reversed" and "wreckage pieces had US markings." These comments often would derive from the residue of an NLP subsystem.

Several widely different approaches may be taken to restrict fusion rule evaluation to the pertinent data, and which one to follow is an important consideration when designing a database system. Our modeling of separate memories in experiments employs the multiple database capability of ROSIE (reference 3). Rulesets are organized according to the memory they use and feed, so activation of a memory is needed less frequently.

The history file contains the data retired from the data fusion system when no longer pertinent to current situation assessment. (The fusion system user, however, should have access to the history file.) In addition to old information, those data include recently fulfilled or rejected plans and predictions. References 1 and 2 discuss and illustrate the retiring process. The history file can be used later in reconstructing and analyzing the flow of events of naval exercises and operations. Most data will not originally be in the syntax of the rule-based system, and some in addition will be in natural language. Also, geographical and other spatial information may be stored in another medium, such as video or optical disks, or in an untranslatable representation. If so, external evaluation of rule conditions involving certain spatial concepts or fuzzy set mappings could be the simpler approach. In figure 1, the external boxes interfaced with "---" represent the processes needed to convert various kinds of data into a usable form. The rule-based system will have the ability to convert formatted data into its own syntax, which is useful when new sources of data are tapped, but, in general, syntax conversion will be more efficiently performed before input.

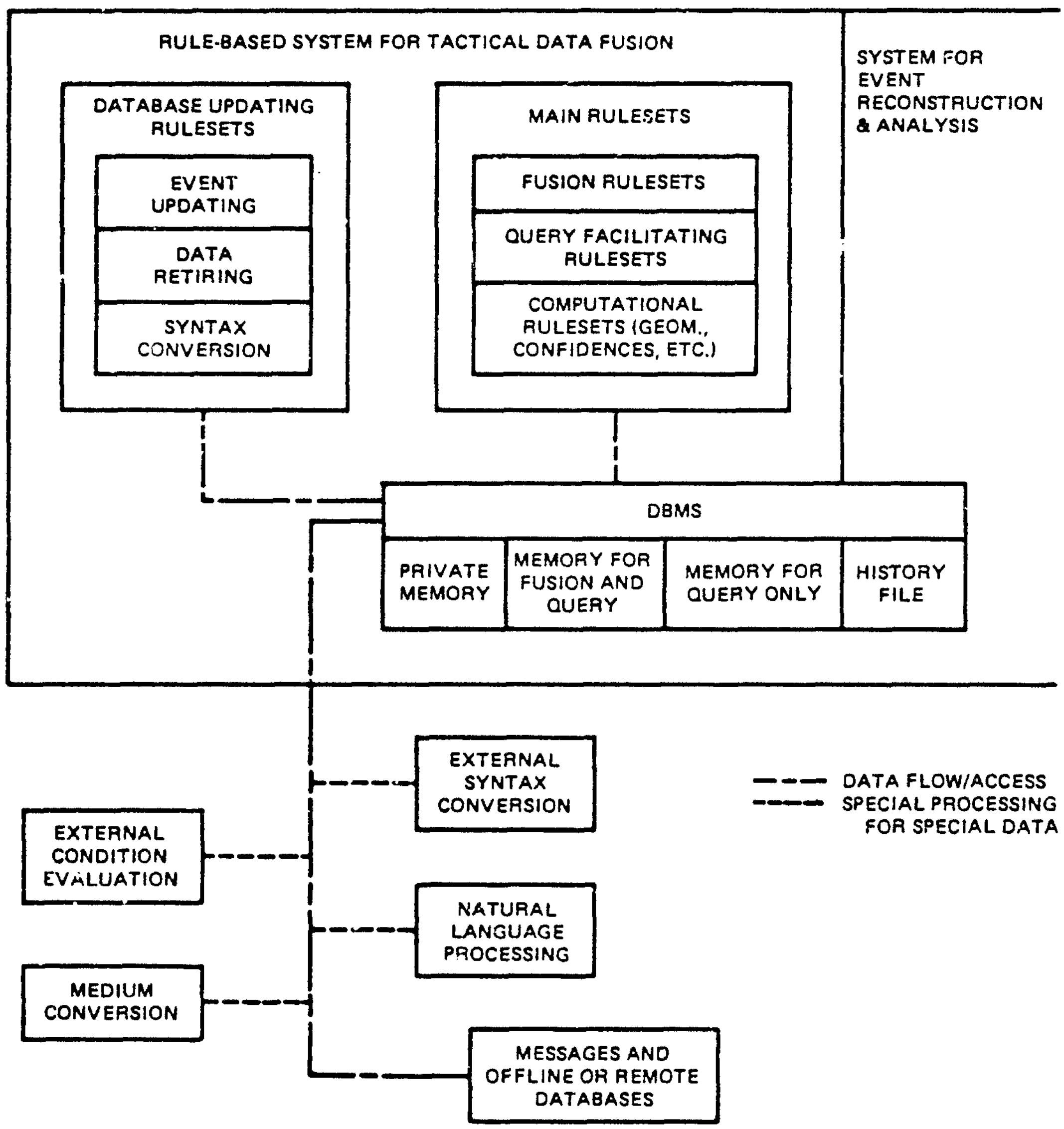

Figure 1. Internal and external interactions of a rule-based data fusion system.

Figure 1 suggests independence of NLP (the conversion of narrative input data into system syntax), inferencing, and query functions. Much of the same knowledge is needed by these processes, as indicated in figure 2, so we are investigating methods of sharing it. Several approaches to sharing reasoning processes among the inferencing subsystem and the NLP subsystem were outlined in reference 2, and much additional work in that area is needed. The sharing of reasoning processes by the query subsystem with other subsystems is also a possibility. Mechanisms for querying the database are currently built into most rule-based systems, but the queries must be stated in highly constrained forms. A fully natural language query capability would require considerable linguistic knowledge. One approach would be to use an "off the shelf" transportable database interface (see references 4 and 5 for a description of the concept) and program it for the tactical applications. Since the linguistic knowledge needed for handling queries differs somewhat from that for converting input data, the failure to share linguistic knowledge with the NLP subsystem might not be unduly wasteful. However, new vocabulary learned by one NLP subsystem would be usefully shared with other NLP subsystems, since the message creator, the "expert" contributing fusion rules, and the query system user will use similar vocabulary.

REPRESENTATION OF DESCRIPTIVE INFORMATION

Several kinds of expert systems will be cooperatively functioning in tactical reasoning activities and will require much of the same knowledge. One system not shown in figure 1 is a mission planning system, and a mission planning system under development (reference 6) uses FRL (frame representation language (reference 7)). While the procedural information of the various systems may be coded very differently, there is a large degree of commonality among the forms in which descriptive information is stored. For example, the tactical situation assessment system STAMMER2 (reference 8) uses relational triples (i.e., 2-node relational assertions), most of the descriptive data in FRL can be easily converted to relational triples (frame-slot-datum), the high-level programming language Prolog includes the representation predicate (arg1, arg2), and ROSIE's many representation forms include the equivalent of relational triples. (For most systems, the node/value may be a tuple or list or the equivalent.) Examples of representations of translatable assertions from a file of infrequently used data are the following. (The relation "IS A" or "isa" is inferred in STAMMER2 when no relation is given. Equivalently, AIO represents "an instance of.")

```
ROSIE: PANAMA is a nation.
       PANAMA CANAL is a part of PANAMA.
       GATUN LAKE is a part of PANAMA.

STAMMER2: (NATION PANAMA)
          (PART PANAMA PANAMA-CANAL)
          (PART PANAMA GATUN-LAKE)

Prolog: nation(panama).
        part(panama, panama-canal).
        part(panama, gatun-lake).
```

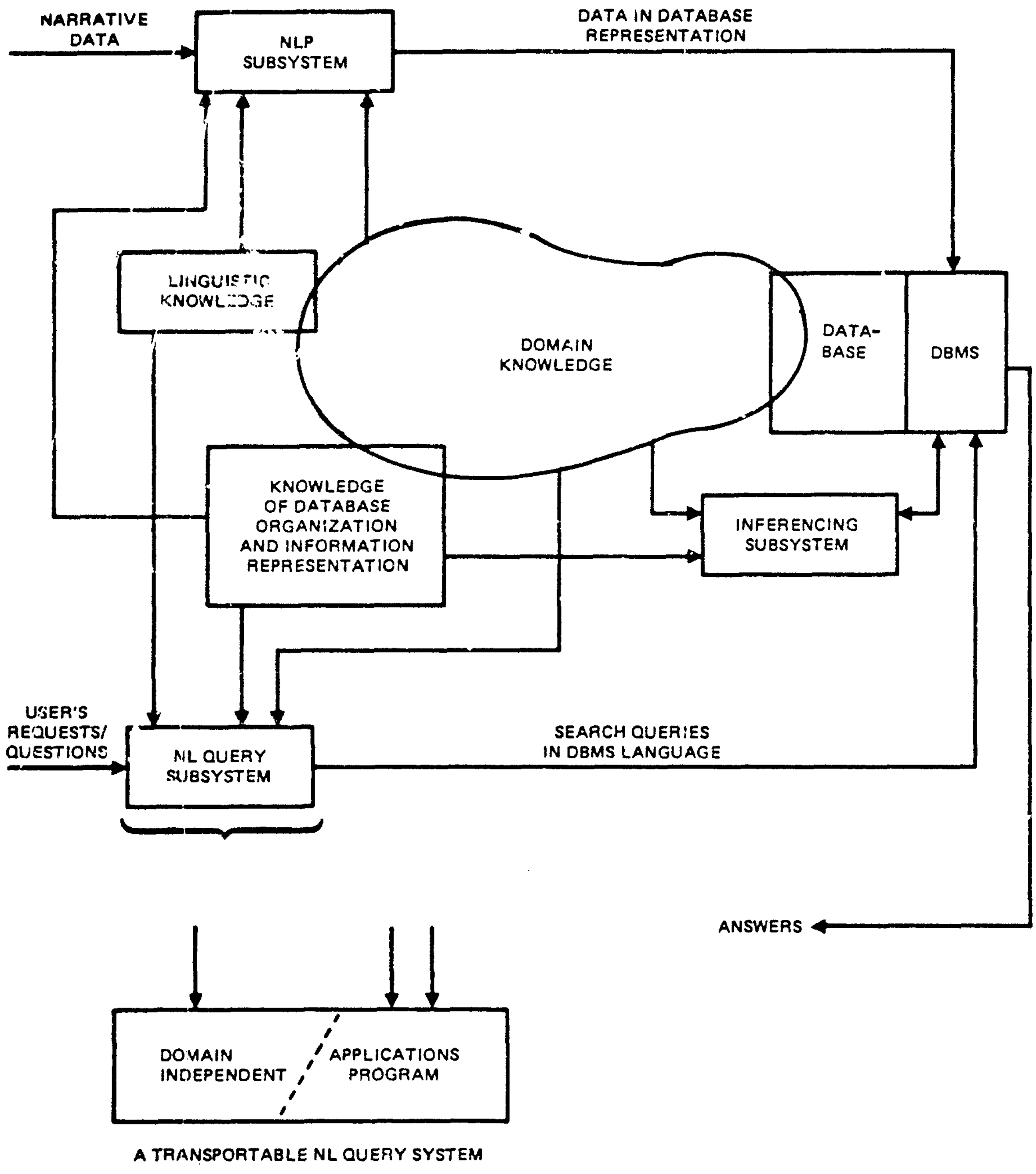

Figure 2. Knowledge shared among subsystems.

```
FRL: (PANAMA (AIO ($VALUE (NATION)))
       (HAS-PART ($VALUE (PANAMA-CANAL) (GATUN-LAKE))))
       (NATION (INSTANCES ($VALUE (PANAMA))))
```

We next consider representing in relational triples the descriptive data shared by the systems. The following facts and educated conjectures should influence the selection of the method of representation of descriptive data.

- The NLP system, planning system, inference system, query system, history system, etc., will all use much of the same data.
- The users generally will not have had experience with expert systems, but WILL be able to quickly learn to read and write in relational triples.
- Information stored as relational triples is easily retrieved.
- More complicated structures (e.g., ROSIE assertions with verbs, adjectives and prepositions) can be used to represent procedural knowledge whether or not descriptive data are in relational triples.

These reasons support the argument for storing shared data as relational triples (or their equivalent) or in a relational database that permits multiple values and permits a value to be a string (e.g., "This is a string.") or a list/tuple (e.g., a Lisp or Prolog list or a ROSIE tuple). (The data used by fusion inference rules do not require strings and could probably be expressed without lists or tuples if necessary.) To test this approach, we will use as examples the events involved in an antisubmarine warfare mission and an anti surface warfare mission.

In reference 2, the concepts of an "actual" event and a "virtual" event were introduced. The term "event" is used in a general sense to refer to an activity, situation, or static state. An "actual" event is one that is reported either to have occurred or possibly to have occurred, and a "virtual" event is one that has not occurred (at the time of the report) and that may be a planned or ordered activity or a prediction or expectation of an event. The concept of a "supporting" event (such as the launching of aircraft in support of an attack) was also introduced in reference 2. Since then we have found it useful to include the concept of an "ongoing" event. In the following discussions, the prefix $ will indicate an actual event, the prefix V$ will indicate a virtual event, and the prefix O$ will indicate an ongoing event.

Much of the message traffic concerns plans. Samples appear in Appendix A under category 1 (failure or absence), category 2 (cause of cessation, failure or plan change), category 5 (conditional plans), and category 6 (other mixtures). Consider, for example, the narrative "SCHEDULED 1200 LAUNCH TO LOCATE AND ATTACK DELTA." Figure 3 gives a representation of the planned events, shown there as virtual events in a "node-arc" depiction. Several "isa" statements involving the events are not shown, and other data from the message and from earlier messages would add other relations to the representation. Later message narrative might never state that a search was conducted, but a number of events such as buoy dropping and sonar dipping would clearly imply a search. By making these individual searching activities subsidiary events of the more comprehensive event labeled $SURVEIL/SEARCH, we will later be able to

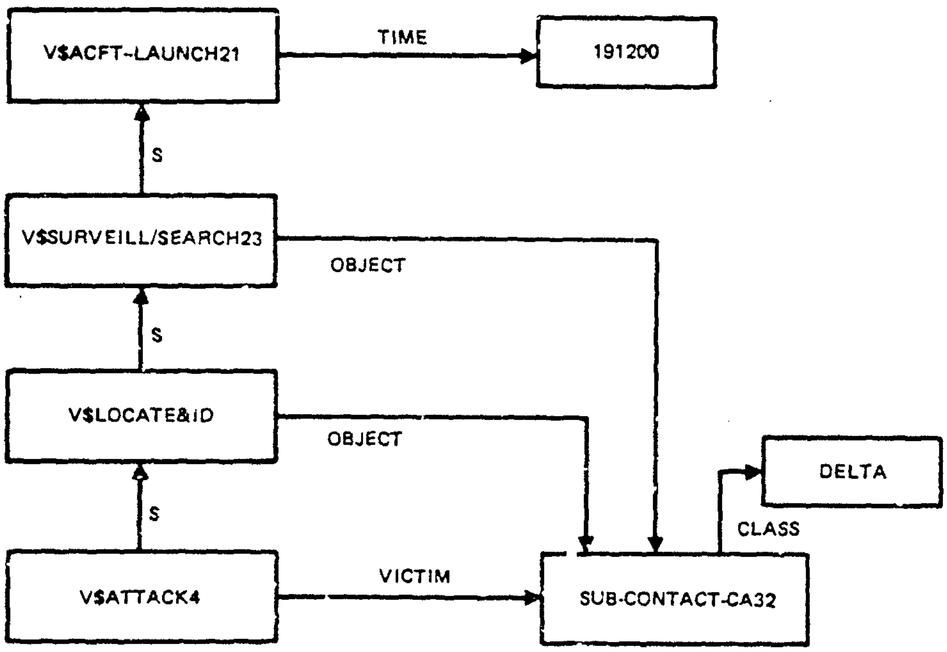

Figure 3. Representation of "SCHEDULED 1200 LAUNCH TO LOCATE AND ATTACK DELTA." The "s" label on arcs represents support.

determine that these events fulfilled the virtual event V$SURVEIL/SEARCH. An option is to exclude the V$LOCATE&ID event in the plan representation, since (as discussed later) it could be inferred if needed. Another option is to create the virtual event V$MISSION which has the other virtual events as subsidiary events. This might be esthetically pleasing and could sometimes be useful, but would add a level of complexity.

Most subsystems (NLP, fusion, planning, etc.) will have a method of numbering successive instantiations of tracks, events, contacts, etc. For example, the tracks created in ROSIE will be TRACK #1, TRACK #2, ... Most other systems written in Lisp represent instantiations with a single Lisp "atom," e.g., SIGHTING1, SIGHTING2, ... We are assuming in our sample implementations in ROSIE that as the system receives data from different sources it renumbers events, reports, etc. One exception to our numbering scheme in ROSIE will be the numbering of contacts. In this case the use of a prefix or suffix or other indication of the source can be used to avoid duplicate names.

Figure 4 gives an example concerning the inability to monitor buoys. Although the NLP system probably would give the helo as the actor, the construction of the representation would include inferring that the patrol is the actor, and the helo would be linked in the manner shown. Figure 4 also illustrates the use of a Lisp string to store comments available to the user but not necessarily to inferencing subsystems. Query and inheritance mechanisms/ rulesets (discussed later) would make it easy to "see" that the patrol is in a helicopter.

Figure 5 shows the key events and relationships in an antisubmarine warfare mission. Examples of the kinds of relationships appearing in the query database for this mission are shown in a ROSIE typescript in Appendix B. Figure 6 gives the representation for an antisurface warfare mission.

A narrative report of a planned or actual aircraft launch usually lists several aircraft, e.g., "SIX A-7, ONE A-6, ONE EA-6B." Different aircraft launched in the same series have different roles and cycle times. The roles may be ASW, surveillance, strike, plane guard (ready for rescue operations), screen, airborne early warning, etc. The report of a planned launch of several aircraft is better represented by a virtual event whose "launched" is a "plat-group" (with its composition specified so far as known) than by an individual virtual event for each plane. Narrative reports of an ASW or strike mission, however, often involve a single aircraft. Note in figures 5 and 6 that the S-3A and E-2 each is represented by a specific aircraft which is a "part" of a launched plat-group.

Other examples of platform groups can be seen in figure 6. While the original plan called for a task group to locate and strike a group of specific platforms, individual matches were not then feasible. Note that a conversion from the narrative (in the caption of figure 6) to the representation requires creating plat-group5 by removing the ship Barsuk from plat-group2.

The asserted actor of an event usually will be a platform or a platform group, but when a better defined actor (e.g., the patrol of an ASW mission) is known, using it as the actor generally will give a more accurate representation.

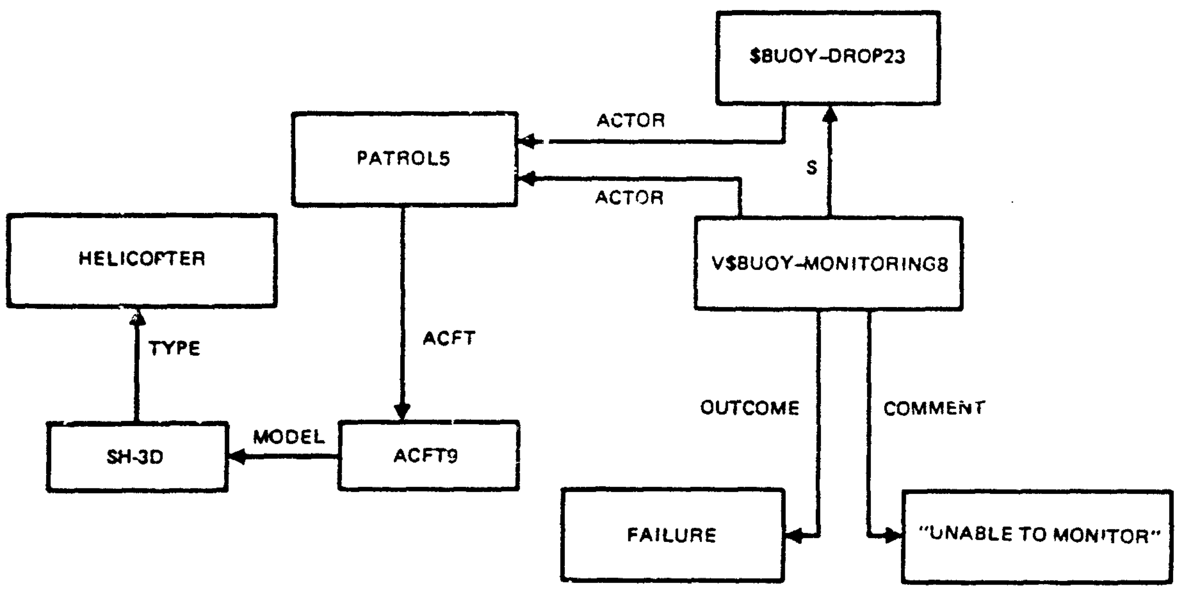

Figure 4. Representation of "helo dropped buoys but was unable to monitor" and some of the background data. The relation "comment" would be accessible to query and history subsystems but not necessarily to inferencing subsystems.

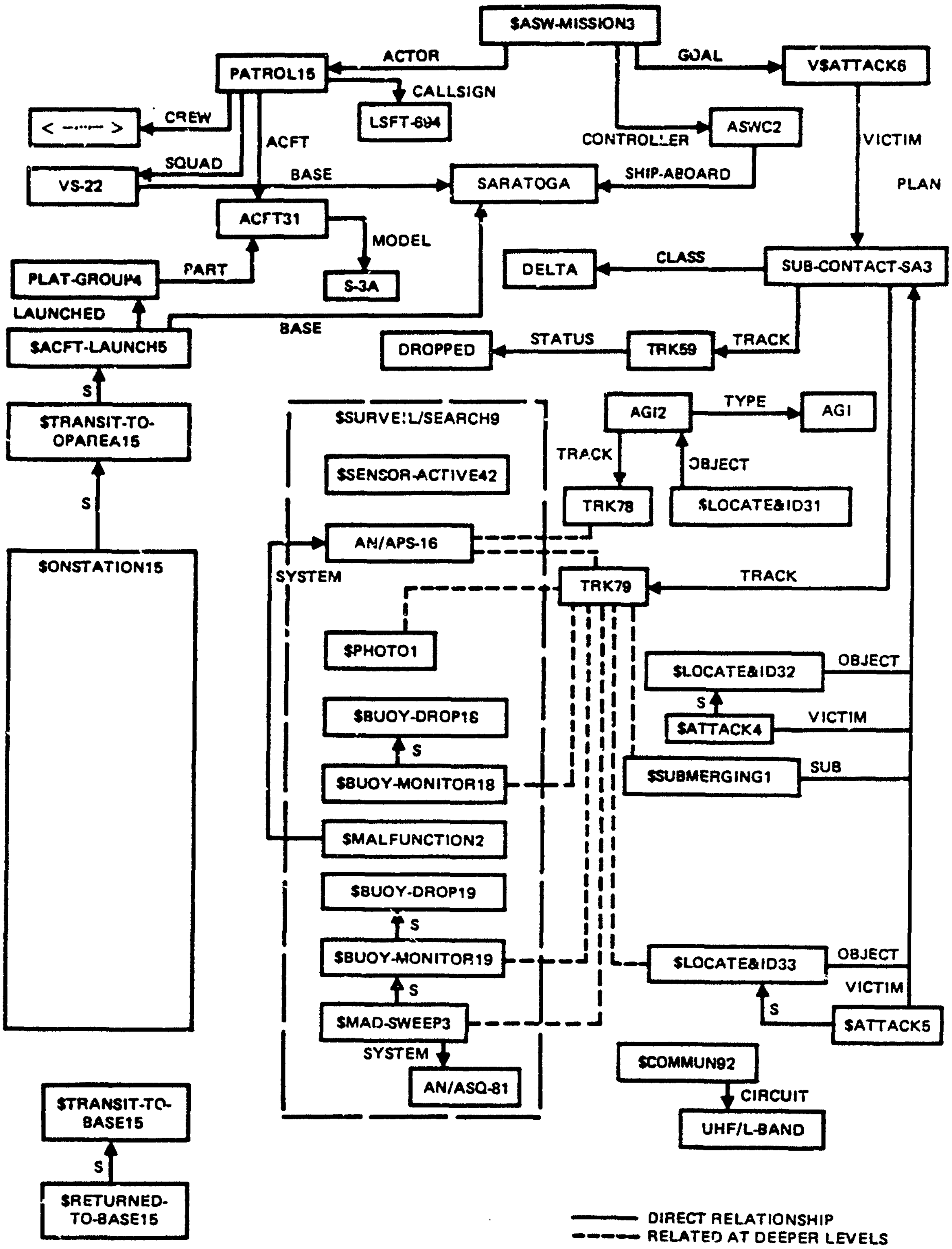

Figure 5. Interrelation of events and other concepts for an antisubmarine-warfare mission.

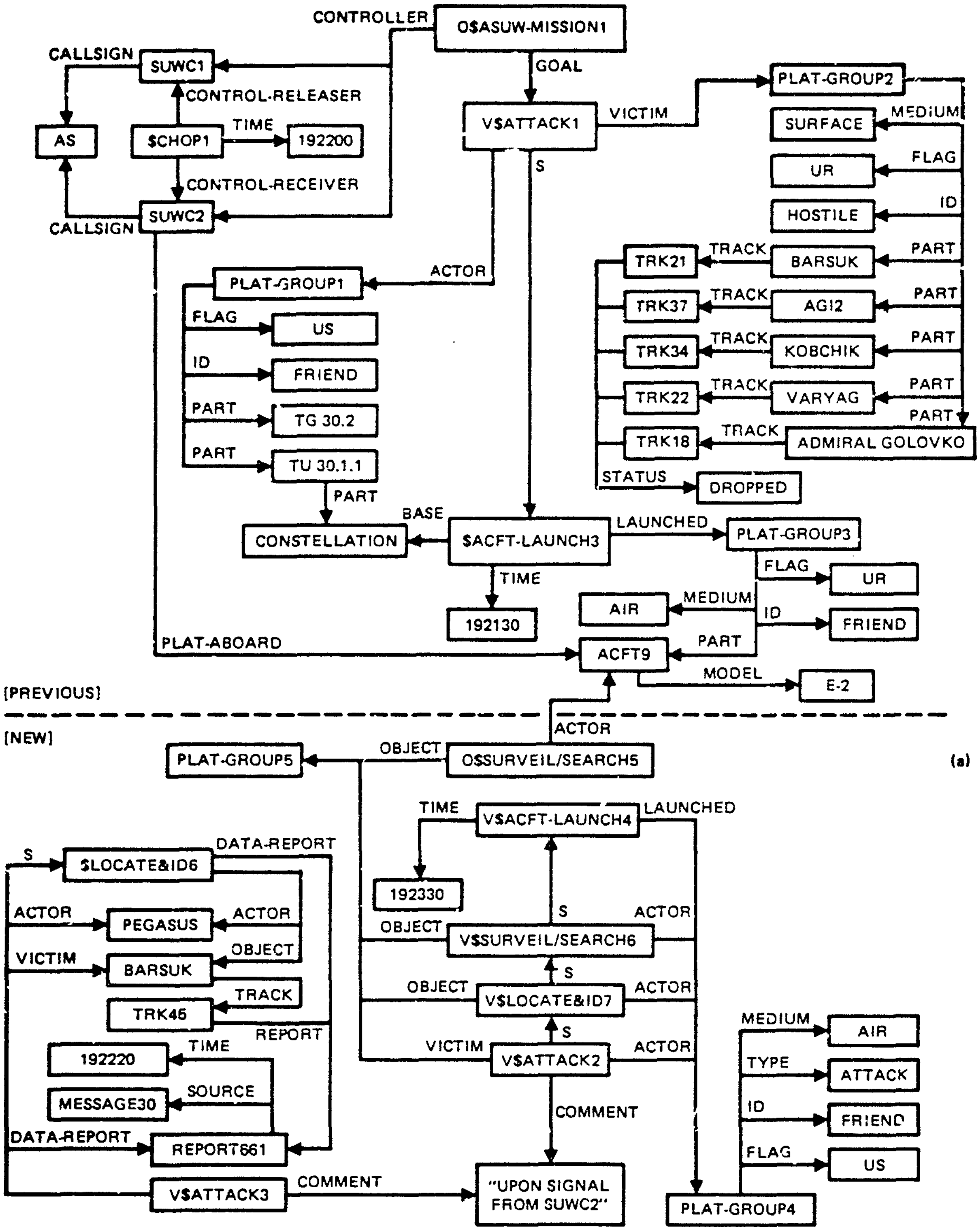

Figure 6. The "new" information shown is taken from a message giving position data on Barsuk and the narrative: "Intention is to have Pegasus maintain position commencing 2330T, to take out Barsuk on signal from AS when hostilities commence. Attack acft from Constellation 2330T launch will be used to strike other hostile surface units, if located, upon commencement hostilities. Currently conducting search with E-2." (a) Key events and relations. (b) Details of the three platform groups: TG 30.2, TU 30.1.1, and plat-group5.

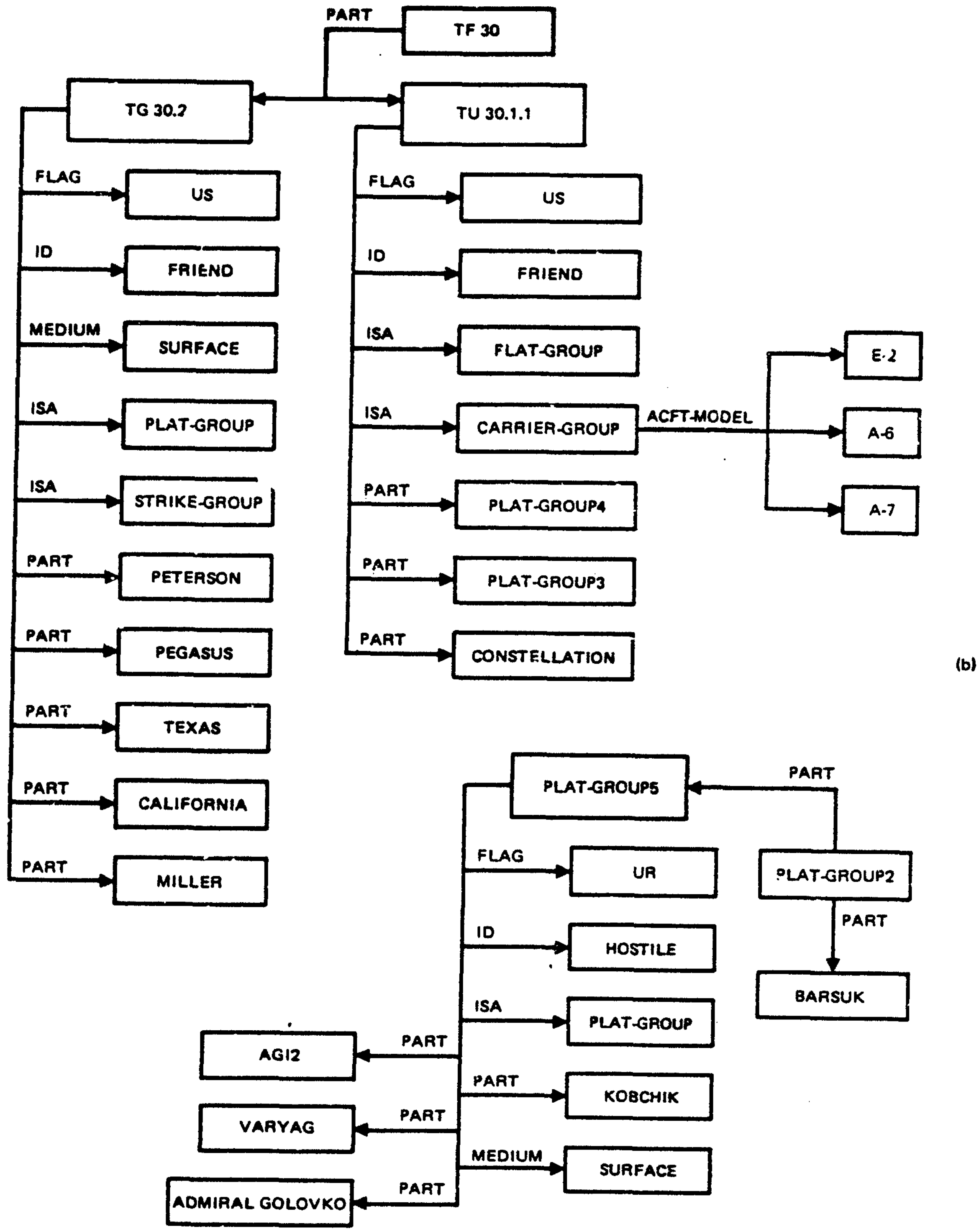

Figure 6B. Details of the three platform groups: TG 30.2, TU 30.1.1, and plat-group5.

The representation of an ASW mission shown in figure 5 disregards any earlier reports providing virtual event records which predict the actual events, with the exception of V$ATTACK6. V$ATTACK6 can be considered as a plan of $ATTACK4 and $ATTACK5 and is fulfilled if the intended victim is out of action as a result. It is also fulfilled, for practical purposes, when the planned attacks are made but are unsuccessful and the mission ends. Similarly in figure 6(a), V$ATTACK1 will be fulfilled when every platform which is a part of plat-group2 is out of action. If we were to continue this particular scenario we would have several attacks per victim platform, and V$ATTACK1 would be fulfilled at the termination of the mission whether or not successful.

Other comments about figures 5 and 6 are listed below.

- The following kinds of assertions are not shown but will be in the database. (The syntax will vary from system to system, and the relation names are arbitrary.)

  - $[event-name] is an event.
  - V$[event-name] is a virtual-event.
  - O$[event-name] is an ongoing-event.
  - V$ATTACK6 is a V$ATTACK.
  - patrol15 is a patrol.
  - $LOCATE&ID32 is an event of $ASW-MISSION3. (Some events, e.g., $SURVEIL/SEARCH, in turn have subsidiary events.)
  - $BUOY-DROP18 is an event of $SURVEIL/SEARCH9.

- Support relationships such as the following usually can be inferred from the event descriptions in the database but are better inferred earlier and entered into the database. (Support is represented by "s" in the figures.)

  - $ACFT-LAUNCH is a support of TRANSIT-TO-OPAREA.
  - $TRANSIT-TO-OPAREA is a support of $ONSTATION.
  - $TRANSIT-TO-BASE is a support of RETURNED-TO-BASE.
  - $SURVEIL/SEARCH is a support of $LOCATE&ID.
  - $LOCATE&ID is a support of $ATTACK.

- The occurrence of some events not in the original data can be inferred from other events.

  - $ONSTA => TRANSIT-TO-OPAREA => (for aircraft) $ACFT-LAUNCH
  - $RETURNED-TO-BASE => $TRANSIT-TO-BASE
  - $ATTACK => $LOCATE&ID

The information shown in figures 5 and 6 (and many of the implied relationships described above) are stored in database as an unordered set of relational triples. Examples of the generic form of these for the ASW mission are shown in table 1. This representation is generic in the sense that the basic elements can be understood, via simple (in principle) translation, by any subsystem or cooperating system.

Table 1. Generic representation of $ASW-MISSION3.

NodeA/OBJECT	RELATION/ATTRIBUTE	NodeB/VALUE
$ASW-MISSION3	ISA	$ASW-MISSION
$ASW-MISSION3	ISA	EVENT
$ASW-MISSION3	ACTOR	PATROL15
$ASW-MISSION3	GOAL	V$ATTACK6
VATTACK6	VICTIM	SUB-CONTACT-SA3
SUB-CONTACT-SA3	CLASS	DELTA
SUB-CONTACT-SA3	TRACK	TRK59

Some additional information about the relations is often needed by participating systems or their user, such as multiple values and automatic inheritance. Additional knowledge often needed concerns the value node; e.g., is it numeric? an integer? a tuple or list?, does it have a finite set or restricted range of values?, etc.

Inheritance mechanisms vary among systems, and translation for the triples concerned will be more difficult. The triples

(CLASS VARYAG KYNDA) and (TYPE VARYAG DESTROYER)

need to be expressed as

(ISA VARYAG KYNDA) and (ISA KYNDA DESTROYER)

if the automatic inheritance mechanisms of some systems are to be employed. Fortunately, almost all relations involving inheritance need be entered into a system only at its initialization, and the problem will rarely occur with tactical event descriptions. Appendix C contains examples of rulesets used in ROSIE to enable inheritance of certain features and attributes. Similar rules can be implemented in STAMMER2, although the current version enters the generated attribute into the database while the current version of ROSIE does not. Inheritance mechanisms can be shared by fusion and question-answering processes.

USER INTERACTIONS

User-assisted fusion is needed, for example, when the conditions of a good tactical inference rule are not all amenable to automatic evaluation. If all but an untractable condition are evaluated true (assuming forward-chaining), the system can ask the user if that condition is met, e.g., if a certain formation or pattern of movements (obvious to the viewer of an NTDS or radar screen) has occurred. Also, a ruleset can be activated enabling a graphical review of the pertinent information. Appendix D shows a sample implementation of ROSIE rulesets enabling interaction with the user. User-assisted fusion probably would have to be an option to each user. As geometry computing capabilities increase, the feature would be needed less.

Automatic computation and presentation of fused data could theoretically be varied from high-threat warnings alone to every new inference plus every shift of confidence values for every set of hypotheses affected by each new report. For the latter situation, the significant conclusions could be identified and only these presented, but there remains the problem that real-time processing may not be achievable for some years. For example, experiments (references 9 and 10) with computing confidences (using Dempster-Shafer methods) for platform identification and simple contact association problems have shown that the computation of confidences will consume a very high proportion of the total processing time. The approach taken in those experiments is to perform confidence computations only when the user requests them, and this would usually be for a single contact after several pieces of evidence concerning the contact are available.

While efficiency is one important aspect of the problem of what to automatically present and what to compute only upon demand, user preference is obviously another. Every user should be informed of urgent problems, e.g., the enemy targeting of a missile on ownship, but some might wish to tell the system what kinds of information to present and then interact seldomly while others might wish to examine the fused data primarily on a query basis. It becomes difficult at this point to distinguish clearly between question-answering processes and fusion processes, and it appears that a desirable capability is that of shifting the different kinds of responsibility from one to the other at the pleasure of the user. A procedure ruleset such as (in ROSIE syntax) "Summarize threats to platform" can respond either to a user request or a routine automated call.

Normally a natural language (NL) interface query system would translate questions into structured queries in the DBMS language to obtain the right answers. As suggested above, however, much of the procedural knowledge needed to retrieve information from the database can be used for more than one function, e.g., the same process might be used in answering a question, in providing a routine situation assessment, in formulating a plan, and in evaluating an inference rule condition. In addition to translating a user's natural language question into a database query, an NL interface could translate more complex requests on selected subjects into the high-level language of the expert system. In ROSIE, for example, procedural knowledge would be implemented as rulesets and the question might be translated into a query using the syntax of the pertinent ruleset headers. Typical available rulesets might include the generator ruleset "to generate opposing_platform with equipX within rangeX of plat" or the predicate ruleset "to decide new-contact is unlike earlier-plat." (Rulesets, in turn, can call on others.) The question, "Which friendly ships are within missile range of which hostile ships?" could be translated into the ROSIE retrieval request (actually, its parsed equivalent):

```
For each platform (friendly)
    whose ID is friend
    and whose medium is surface,
```

```
for each missile-type (msl),
for each opposing platform (enemy) with msl within
    (the max-range of msl) of friendly,
display friendly
and display enemy.
```

Note that the call on the "opposing platform" ruleset activates geometry computations using ship positions, so this process is not data retrieval alone. We have not concluded that translation of queries into an expert system's high-level language is a practical approach, but it is one to be considered.

Tailoring the system to a particular user could be achieved in a number of ways, one being to use flags which permit certain procedural rulesets to be activated under certain conditions. The flag values for a particular user would be set during an initial interrogation and could be changed by that user any time.

SUMMARY

The emphasis in this task has been on the fusion of tactical data, but fusion processes must integrally overlap with planning, query, and historical reconstruction/analyses processes. Many different kinds of knowledge engineering approaches are being applied to the various facets of C3 problems. The data structures of the applicable expert systems vary greatly, and, in general, "talk" among these systems has not occurred. The two primary concerns addressed in this report are the following.

- The cooperative subsystems of a total C3 system must share descriptive data. Can plans, events and other complex concepts be represented in a "generic" form understandable by all of these subsystems?
  - Tactical planning systems must base plans on assessments of tactical situation.
  - Data fusion processes must have knowledge of the ownforce plans being carried out in order to assess the situation.
  - Reconstruction and post-analysis systems must have knowledge of plans, events, and the fusion system conclusions.
  - Query systems — and each above subsystem might have its own — must be able to answer questions about the plans and fused data.
- Much of the same procedural knowledge is needed by the different subsystems. How can they efficiently share it?
  - Question-answering and data fusion often involve the same reasoning and computing processes.
  - One user may wish certain information to be automatically presented while another may wish it fused and presented only upon request.

The approach recommended for sharing descriptive data is to use two-node relational assertions, allowing the value node to have multiple values. Values should be permitted to be strings or lists/tuples, primarily for use by the query system. Examples of antisubmarine warfare events and antisurface warfare events were implemented to test this approach. Representations made use of three kinds of events: "virtual" (planned/predicted), "ongoing," and "actual." The representation of events having multiple actors and/or multiple objects or victims made use of "platform groups."

Procedural responsibilities can be shifted between fusion functions and question-answering functions by representing the sharable processes modularly, generally as distinct rulesets. These rulesets/modules could be called both by the fusion functions the user wishes automatically performed and by a query system which translates questions into the high-level language of the expert system.

Other issues addressed in this report were the sharing of linguistic knowledge, user-assisted fusion, and the subdivision of memory for efficient rule evaluation.

REFERENCES

1. Dillard RA.
Experimental Modeling of a Tactical Data Fusion System.
Technical Document 548, (Subject to Export Control Laws)
Naval Ocean Systems Center, October 7, 1982.

2. Dillard RA.
Integration of Narrative Processing, Data Fusion, and Database Updating Techniques in an Automated System.
Technical Document 480, Naval Ocean Systems Center, October 29, 1981.

3. Waterman DA, Anderson RH, Hayes-Roth F, Klahr P, Martins G, Rosenschein SJ.
Design of a Rule-Oriented System for Implementing Expertise.
Report N-1158-1-ARPA, The Rand Corporation, Santa Monica, CA, May 1979.

4. Templeton M and Burger J.
Problems in Natural-Launguage Interface to DBMS with Examples from EUFID.
In Proceedings of the Conference on Applied Natural Language Processing. February 1983.

5. Grosz BJ.
TEAM: A Transportable Natural-Language Interface System
In Proceedings of the Conference on Applied Natural Language Processing. February 1983.

6. Bechtel RJ.
Mission Planning Support System.
Semiannual Report, Naval Ocean Systems Center, September 1982.

7. Roberts RB and Goldstein IP.
The FRL Manual.
Memo 409, MIT, September 1977.

8. McCall DC, Morris PH, Kibler DF, and Bechtel RJ.
STAMMER2: A Production System for Tactical Situation Assessment.
Technical Document 298, Volumes 1 and 2, Naval Ocean Systems Center, October 1979.

9. Dillard RA.
Computing Probability Masses in Rule-Based Systems.
Technical Document 545, Naval Ocean Systems Center, September 1982.

10. Dillard RA.
Computing Confidences in Tactical Rule-Based Systems Using Dempster-Shafer Theory.
Technical Document 649, Naval Ocean Systems Center, 3 October 1983.

APPENDIX A: EXAMPLES OF NARRATIVE CONCEPTS

1. Failure or Absence

PEGASUS UNABLE TO ACQUIRE BARSUK.

E-2 UNABLE TO LINK ...

UNABLE TO CONTACT AAWC ON VIRGINIA

UNABLE TO ESTABLISH COMM WITH SFC UNITS ON ANY CKTS.

HELO DROPPED BUOYS BUT WAS UNABLE TO MONITOR.

NIL WEAPONS FIRED.

NO UNUSUAL AIR ACTIVITY REPORTED

MY PRESENCE UNKNOWN TO ENEMY.

SUSTAINED NO DAMAGE FROM KYNDA.

THE REMAINING SWIMMER PAIRS SUCCESSFULLY PLACED ... WITHOUT BEING DETECTED.

DIFAR BUOYS WERE DEPLOYED BUT WERE UNABLE TO FIX SUB.

REGRET ADDITIONAL VP ASSETS UNAVAILABLE TO ASSIST IN PROSECUTION.

CREW ALSO CONDUCTED RADAR, ESM AND FLIR SEARCH WITHOUT SUCCESS

2. Cause of Cessation, Failure, or Plan Change

LOST CONTACT AS SUB WENT BELOW LAYER.

HEAVY SMOKE PRECLUDES FURTHER INTERPRETATION.

CONTACT WAS LOST BEFORE TARGET COULD BE LOCALIZED.

SHIPS SONAR OOC AS RESULT OF LIMPIT MINE EXPLOSIONS.

... BUT DUE TO INTERMITTENT COVERED VOICE WAS UNABLE TO RAISE CONTROLLERS.

... BUT WAS INTERCEPTED BY ORANGE AIRCRAFT AND FORCED TO DISCONTINUE SEARCH.

BECAUSE OF TIME DELAY AND DISTANCE WHICH WILL BE NECESSARY TO TRANSIT TO PHASE IV START PTS, UNITS WILL BE UNAVAILABLE TO REFUEL, REARM, AND ARRIVE AT START PTS ON TIME. IT MAY BE NECESSARY TO CANCEL REARM.

FANNING AND MILLER PURSUED KOBCHIK UNTIL WITHIN GUN RNG.

LATE ONSTA DUE ACFT MALFUNCTIONS

TASS UNITS NOT TRAILING ARRAYS DUE HIGH SPEED OPS AND SURFACE ENGAGEMENTS

UNABLE TO PROSECUTE FURTHER DUE TO NEW TASKING BY COMDESRON 23.

REMAINED ONSTA FOR EXTPA 1 15 HRS DUE TO LATE ARRIVAL OF RELIEF.

... HOWEVER SHORT ONSTA TIME REMAINING PRECLUDED FURTHER TARGET PROSECUTION.

HEAVY ELECTRONIC EMISSIONS NOTED EMANATING FROM AGI ... TOOK AGI UNDER FIRE ... ALL ELECTRONIC EMISSIONS FROM AGI CEASED

MILLER TOOK UNIDENT CONTACT WHICH FAILED TO RESPOND TO CHALLENGE UNDER FIRE. SHIP MAY HAVE BEEN WAINWRIGHT.

TRACKED SUB FOR 1 HR 30 MIN THEN LOST CONTACT.

SEARCH EFFORT WILL CONTINUE ... UNTIL ALL ORANGE UNITS LOCATED.

3. Time

3a. Order

VISUAL ID ON ADM GOLOVKO, ATTACKED SUCCESSFULLY ON ADM GOLOVKO WITH GUNS. BROKE ENGAGEMENT. RETURNED TO SCREEN. ADM GOLOVKO CLEARING ...

GREEN FLARES SIGHTED SHORTLY AFTER VISUAL ON PERISCOPE

P 10 COMMENCED ATTACK RUN BY FIRING TWO TORPEDOES FOLLOWED BY TWO MISSILES.

ESTABLISHED COMMS WITH USS BRONSTEIN PRIOR TO ARRIVAL ONSTA 2300Z

PROCEEDED TO LAY AN INVERTED ACTIVE CHEVRON BARRIER AND IMMEDIATELY GAINED CONTACT

3b. Interval

HELD CONTACT FOR 7 MIN.

HELD FIRM CONTACT FOR 25 MIN. BRIEF CONTACT HELD ON POSS SUB.

PERISCOPE REMAINED VISUAL FOR APPROX 15 MIN AT SNORKEL DEPTH

PROVIDED SURFACE PLOT INFO AND COMM RELAY FOR TASS UNITS THROUGHOUT PERIOD

OBTAINED MODE IV CHECK AND LINK CHECK ENROUTE BASE

THIS INTERCEPT OCCURRED WITHIN THE TIME FRAME DURING WHICH A MISSING RA-5C WOULD HAVE BEEN IN THE VICINITY OF PURPLE.

3c. Ongoing

CONTINUING EFFORT TO LOCATE/IDENTIFY ...

MILLER HAS HAD CONTINUOUS CONTACT WITH SKAMPI ...

CURRENTLY CONDUCTING SEARCH WITH E-2.

AS OF 202300T, ONE E-2 ACFT UNDER AW CONTROL IS PROVIDING SURFACE SURVEILLANCE DATA AS FEASIBLE.

PEGASUS CURRENTLY ENROUTE TO ATTACK ZHEVNY WITH GUN.

E-2 ACFT PRESENTLY ASSIGNED SU... ...OVIDING UPDATE ON ALL TRACKS.

ACFT RETURNING TO CONSTELLATION FOR 2030 RECOVERY.

CONTINUING TO SEARCH AREA.

SMALL MISSILE CRAFT CONTINUE HARRASSMENT OF MERCHANT SHIPS.

CONTINUED SEARCH FOR AGI PELENG AT LAST KNOWN POSIT.

... AND CREW CONTINUED TO TRACK.

FORCE HAS SUSTAINED LIMITED DAMAGE, BUT IS CONTINUING ENROUTE PT WINFIELD.

AT 1716Z RESUMED PLANE GUARD.

COMMENCING 2330T CONTINUOUSLY MONITOR STATION RELATIVE TO BARSUK FROM WHICH YOU CAN LAUNCH IMMEDIATE MISSILE ATTACK.

INTEND CONTINUE EFFORT TO LOCATE REMAINING KYNDA

INTENTIONS: CONTINUE TRANSIT LOCATING AND ENGAGING ORANGE UNITS.

<u>3d. Other</u>

...TO UPDATE POSITION ON ORANGE UNITS AT LEAST EVERY SIX HOURS.

SUBSEQUENTLY FORCE SUBJECTED TO INTERMITTENT ATTACK BY AIR, SURFACE AND SUBSURFACE UNITS THROUGHOUT THE DAY.

INTEND LAUNCH ADDITIONAL ACFT AT 221130T TO ... . NO ACFT AVAIL PRIOR THAT TIME.

4. Logical and Fuzzy Quantifiers

10 ASROC REMAINING.

FIRED 2 OR 3 BIRDS EACH TARGET.

ALL ORANGE SURFACE COMBATANTS STRUCK ONCE SINCE 221115T; BARSUK STRUCK TWICE.

ALL ORANGE SURFACE UNITS HAVE NOW BEEN STRUCK AT LEAST ONCE AND SEVERAL REPEATEDLY WITH CONSTELLATION ACFT.

CONTACT APPEARS TO BE ON PATROL SURVEILLANCE STATION

ONLY ESM INTERCEPT RECEIVED ON PELENG SINCE 171624Z1.

IDENTITY OF BARSUK AND KOBCHIK MAY BE REVERSED.

HELO REPORTED FF TYPE WITH HULL NUMBERS 32 WITH LAST NUMBER INDISTINGUISHABLE.

RELIABLE INTELLIGENCE POSITIONS TWO ORANGE SUBMARINES TO WEST OF BLUE ANCHORAGES.

IN A POSSIBLE RELATED REPORT, A RELIABLE SOURCE ON PURPLE DISCOVERED WHAT HE BELIEVES TO BE A RECENT AIRCRAFT WRECKAGE SCATTERED OVER A LARGE AREA OF THE WESTERN PORTION OF THE ISLAND NATION, ONE OF THE LARGER PIECES FOUND HAD BLUE MARKINGS.

5. Conditional Plans

WILL PICK UP AMPLIFIER VIA MY HELO IF AVAIL.

ATTACK ACFT FROM CONSTELLATION 2330T LAUNCH WILL BE USED TO STRIKE OTHER ORANGE SURFACE UNITS, IF LOCATED, UPON COMMENCEMENT HOSTILITIES.

INTEND LAUNCH ALERT ACFT FOR STRIKE AGAINST ADM GOLOVKO WHEN LOCATED.

PHOTOGRAPH AND ID IF SURFACING OCCURS.

IF DESIRE REFUEL SURFACE COMBATANTS RCMD HAVE MLSF GROUP CLOSE ASTERN OF CV GROUP AND BREAK OFF COMBATANTS TO REFUEL INDIVIDUALLY.

6. Other mixtures

ATTACK WILL BE LAUNCHED ON SIGNAL FROM AS UPON COMMENCEMENT HOSTILITIES SOMETIME AFTER 2359T.

TASS SHIPS ARE WIDELY DISTRIBUTED AT THIS TIME ...

ALERT A-6, A-7 TO AUGMENT AS NECESSARY FOR MAJOR STRIKES AGAINST ORANGE UNITS PRESENTING IMMEDIATE THREAT TO FORCE.

ANTICIPATE APPROX 60 ACFT ARE AVAIL FOR IMPENDING OPERATIONS.

BELIEVE I HAVE RUN ACROSS ALL ORANGE FORCES.

WILL STRIKE KOBCHIK WITH AIRBORNE ACFT WHEN HOSTILITIES COMMENCE.

PELENG REMAINING IN GENERAL VICINITY NIMITZ BUT NOT IN CONTINUOUS CLOSE SURVEILLANCE/SHADOWING ROLE. NO UNUSUAL ACTIVITY.

ACFT DROPPED DIFAR BUT ADP HAD FAILED. UNSUCCESSFULLY ATTEMPTED TO VECTOR HELO TO SINKER POSITION, WHICH WAS 5NM 225T FROM CV

CONTINUED TO DIP AND CONDUCT MAD SWEEPS WITHOUT CONTACT. RETURNED TO PLANE GUARD AT 0130Z.

BRONSTEIN LATER GAINED TASS CONTACT AND USING DIFAR BEARING INFO GENERATED BY SLVG 703, DEVELOPED AN AOP WITH CUS/SPD.

LOST ACTIVE SONAR CONTACT PRIOR TO OBTAINING FIRING SOLUTION. NO JOY AT ATTEMPT TO REGAIN CONTACT ANY SENSOR.

Acronyms

AAWC -- anti-air warfare commander
ADP -- automatic data processing
AOP -- area of position/probability
AS -- call sign of commander
CKTS -- circuits
OOC -- out of commission
SFC -- surface
SLVG 703 -- a patrol's call sign
SUWC -- surface warfare coordinator
VP -- US patrol squadron

Soviet Ships

ADMIRAL GOLOVKO
BARSUK
KOBCHIK
KYNDA (a class name)
PEGASUS
PELENG
SKAMPI (a fictitious sub)

APPENDIX B. EXAMPLES OF DATA IN QUERY-ONLY MEMORY

```
                    TM
        [ Rosie    Version 1.3  11-Jul-83  07:29:13 ]
<3> load cc-querybase.
<4> display the crew of patrol #15.
<4, TC-LCDR BURNE, PC-LT FULLER, DBO-LT MORGAN>
<5> describe pattern #8.
[ PATTERN #8 ]
    PATTERN #8 is a pattern.
    PATTERN #8 is a pattern of $BUOY-DROP #8.
    BARRIER is a type cf PATTERN #8.
    60FT is a depth of PATTERN #8.
    3HR is a duration of PATTERN #8.
    225T is an orientation of PATTERN #8.
    11 is a max-row-length of PATTERN #8.
    4NM is a buoy-distance of PATTERN #8.
    11 is a buoy-count of PATTERN #8.
    1 is a row-count of PATTERN #8.

<6> display every expendable of $ASW-mission #3.
<25, SSQ-41>
<1 MK-46>
<2, MK-82>
<7> for each event (e),
    if there is a comment (c) on e,
    display e and display c.
$BUOY-DROP #8
"ahead of Brumby"
$BUOY-MONITOR #8
"held weak contact 7 min"
$MALFUNCTION #2
"ATR tube"
<8> for each report (r),
     if there is a comment (c) on r,
     display r and display c.
REPORT #810
"maneuvering, changing depth"
<9> logout.
```

APPENDIX C. RULESETS FOR INHERITANCE

```
[: CC-INHERIT  Created  7-Jul-83 12:37:06, edit by DILLARD :]

To generate medium of platform:

[1] If (there is a type (x) of the platform
or there is a model (x) of the platform)
and there is a medium (m) of x,
produce m.

End.


To generate type of platform:

[1] If (there is a class (x) of the platform
or there is a model (x) of the platform)
and there is a type (t) of x,
produce t.

End.


To generate ID of platform:

[1] If (there is a class (x) of the platform
or there is a model (x) of the platform)
and there is an id (f) of x,
produce f.

[2] If there is a type (t) of the platform
and t = AGI,
produce hostile.

End.


To generate flag of platform:

[1] If (there is a class (x) of the platform
or there is a model (x) of the platform)
and there is a flag (f) of x,
produce f.

[2] If there is a type (t) of the platform
and t = AGI,
produce UR.

End.
```

To generate nation of platform:

Private phlag.

[1] If there is a flag (f) of the platform,
let the phlag be f,
otherwise,
return.

[2] If the phlag = US,
produce US.

[3] If the phlag = UR,
produce USSR.

[4] If the phlag = UK,
produce United Kingdom.

[5] If the phlag = JA,
produce Japan.

[6] If the phlag = IR,
produce Iran.

[7] If the phlag = CU,
produce Cuba.

[8] If the phlag = CH,
produce China.

[9] If the phlag = CA,
produce Canada.

[10] Produce the phlag.

End.

To generate propulsion of sub:

[1] If there is a class (c) of the sub
and there is a propulsion (p) of c,
produce p.

End.

```
To decide plat is a ship:

Private med.

[1] If the plat is a platform
and there is a medium (m) of the plat,
let the med be m,
otherwise,
conclude false.

[2] If the med = surface
or the med = subsurface
or the med = amphibious,
conclude true.

[3] If the med = air
or the med = land-surface,
conclude false.

End.


To generate ship:

[1] For each platform
which is a ship,
produce that platform.

End.


To decide plat is an aircraft:

Private med.

[1] If the plat is a platform
and there is a medium (m) of the plat,
let the med be m,
otherwise,
conclude false.

[2] If the med = air,
conclude true.

[3] If the med = surface
or the med = subsurface
or the med = amphibious
or the med = land-surface,
conclude false.

End.
```

```
To generate aircraft:

[1] For each platform
which is an aircraft,
produce that platform.

End.


To decide system is an equipment of plat:

[1] If (there is a class (x) of the plat
or there is a model (x) of the plat)
and the system is an equipment of x,
conclude true.

[2] Conclude false.

End.


To generate max-speed of plat:

Private plat-type.

[1] If (there is a class (x) of the plat
or there is a model (x) of the plat)
and there is a max-speed (ms) of x,
produce ms.

[2] If there is a medium (m) of the plat
and m = air,
if there is a type (t) of the plat
and t = helicopter,
produce 150,
otherwise,
produce 1200.

[3] If there is no type of the platform,
produce 40,
otherwise let the plat-type be the type of the plat.

[4] If the plat-type = fast-attack/patrol-craft,
produce 45.

[5] if (the plat-type = carrier
or the plat-type = cruiser
or the plat-type = destroyer),
produce 35.

[6] Produce 30.

End.
```

```
To generate time of event:

[1] If there is a report (r)
 such that (r is a data-report of the event
   and there is a time (t) of r),
produce t.

End.


To decide thing is a component of group:

[1] If the thing is a part of the group,
conclude true.

[2] For each part (p) of the group,
if the thing is a part of p
or (there is a part (pp) of p
such that the thing is a part of pp),
conclude true.

[3] Conclude false.

End.


To generate component of group:

[1] For each part (p) of the group,
produce p
and for each part (pp) of p,
produce pp
and for each part (ppp) of pp,
produce ppp.

End.


To generate actor of subevent:

[1] If there is a type (t) of the subevent
 and (t = malfunction
   or t = aircraft-launch
   or t = change-op-control),
return.

[2] If there is an event (e) such that
   the subevent is an event of e,
produce the actor of e.

End.
```

APPENDIX D. RULESETS FOR USER-ASSISTED FUSION

```
[: CC-USER-INPUT  Created  1-Jun-83 13:08:28, edit by DILLARD :]

To generate user-input on subject for argument:

Private answer.

[1] Send {return,
"Please type y, n, x (for Don't know) or ? (for Give more data).",
return}.

[2] Read for 240 seconds {1 line (bind response)}
and (if response = "?
",
go tell_more about (the subject) for the argument
and send {return,
"Please type y, n, or x.",
return}
and read for 240 seconds {1 line (bind response)})
and let the answer be the name {response}.

[3] If the answer ~= y
and the answer ~= Y
and the answer ~= n
and the answer ~= N
and the answer ~= x
and the answer ~= X,
send {return,
"Didn't catch that answer.  Type y, n, x, or ?.", return}
and read for 240 seconds {1 line (bind response)}
and (if response = "?
",
go tell_more about (the subject) for the argument
and send {return,
"Please type y, n, or x.",
return}
and read for 240 seconds {1 line (bind response)})
and produce the name {response}.

[4] Produce the answer.

End.


To tell_more about subject for argument:

[1] If the subject = drift-ice-avoidance
or the subject = neutral-coast-hugging,
describe_track of the argument.

End.
```

```
To decide report does not_show drift-ice-avoidance:

Private answer.

[1] If there is a course-change-cause (C3) of (the report)
such that c3 = ice-avoidance,
conclude false.

[2] If there is a latitude (lat) of the report
and lat < 60
and lat > -60,
conclude true.

[3] If the user-flag = interaction,
send (return,
"Is it likely that ", the report, " shows a change in course ",
"because of drift-ice?", return)
and let the answer be the user-input on drift-ice-avoidance
   for the report,
otherwise,
conclude true.

[4] If the answer = N
or the answer = X,
conclude true.

[5] If the answer = Y,
assert ice-avoidance is a course-change-cause of the report
and conclude false.

[6] conclude true.

End.


To describe_track of xreport:

Private xtime.

[1] Let the xtime be the time of the xreport.

[2] If there is a track (tr)
such that the xreport is a report in tr,
send (return, tr)
and (if there is a platform (plat) of tr,
send ("    Platform:  ", plat, return))
```

```
and for each report (r) in tr,
if the time of r <= the xtime,
send {r, "  Time:  ", the time of r}
and (if there is a latitude (lat) of r,
send {"   Lat:  ", lat, "     Lon:  ", the longitude of r})
and (if there is a course (c) of r,
send {"   Course:  ", c})
and send {return}.

End.


To decide report does show neutral-coast-hugging:

Private answer.

[1] If there is a characteristic (c) of (the report)
such that c = neutral-coast-hugging,
conclude true.

[2] If region is far from land,
conclude false.

[3] If the user-flag = interaction,
send {return,
"Does the contact appear to be hugging the coast line of a
neutral country?",
return}
and let the answer be the user-input
   on neutral-coast-hugging for the report,
otherwise,
conclude false.

[4] If the answer = N
or the answer = X,
conclude false.

[5] If the answer = Y,
assert neutral-coast-hugging is a characteristic of the report
and conclude true.

[6] Conclude false.

End.
```