AD-A099 205 OPERATING SYSTEMS INC WOODLAND HILLS CA F/G 9/2
AIS DATA BASE GENERATION.(U)
APR 81 D L DWIGGINS, G M SILVA F30602-80-C-0047
UNCLASSIFIED OSI-81-002 RADC-TR-81-43 NL

1 OF 1
AD A
099205

END
DATE
FILMED
6-81
DTIC

(12)

10

LEVEL II

RADC-TR-81-43
Final Technical Report
April 1981

AIS DATA BASE GENERATION

Operating Systems, Inc.

Donald L. Dwiggins
Georgette M. Silva

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-81-43 has been reviewed and is approved for publication.

APPROVED:

ZBIGNIEW L. PANKOWICZ
Project Engineer

APPROVED:

JOHN N. ENTZMINGER
Technical Director
Intelligence and Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRDT) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) **REPORT DOCUMENTATION PAGE**	**READ INSTRUCTIONS BEFORE COMPLETING FORM**
1. REPORT NUMBER 18 RADC-TR-81-43	2. GOVT ACCESSION NO. AD-A099205
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 AIS DATA BASE GENERATION	(9) 5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 15 Dec 79 – 15 Jan 80
(14)	6. PERFORMING ORG. REPORT NUMBER OSI-81-002
7. AUTHOR(s) 10 Donald L. Dwiggins Georgette M. Silva	(15) 8. CONTRACT OR GRANT NUMBER(s) F30602-80-C-0047
9. PERFORMING ORGANIZATION NAME AND ADDRESS Operating Systems, Inc. 21031 Ventura Boulevard Woodland Hills CA 91364	(16) 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 45941634 17 16
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRDT) Griffiss AFB NY 13441	(11) 12. REPORT DATE April 1981
	13. NUMBER OF PAGES 50
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same 12 24	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

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)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Zbigniew L. Pankowicz (IRDT)

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

Artificial Intelligence	Natural Language Understanding
Computational Linguistics	Message Processing Technology
Cognitive Psychology	Automated Data Base Generation
Knowledge Representation	PROLOG Language

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

This report documents results of a 12-month R&D effort consisting in development of MATRES III, a language understanding system for automated generation of AIS data base elements. Section 1.0 discusses the need for a high-volume message processing technology in the I&W environment, and summarizes its current state of the art exemplified in MATRES II. Section 2.0 describes in detail the methodological approach adopted in MATRES III for analysis/description of event reports, and

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

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391114

provides background information on conceptually relevant efforts. Section 3.0 provides an overview of message processing from raw text input to final output of event record in MATRES III. Section 4.0 presents MATRES III model data base system, and Section 5.0 provides guidelines for continuing development of message processing technology. Examples of message segments and event records synthesized from these segments are presented in Appendices A and B, respectively.

PREFACE

This document is the Final Technical Report on work performed by Operating Systems, Inc. under RADC Contract No. F30602-80-C-0047, entitled *AIS Data Base Generation*.

All examples of messages and message fragments (sentences) in the body of this document are based upon facts and events reported in the daily press -- specifically, the Los Angeles Times (July 1979) -- Aviation Week (February 1973 to July 1979), American and Soviet Military Trends since the Cuban Missile Crisis by John M. Collins, published by The Center for Strategic and International Studies, Georgetown University, Washington, D.C., 1978, and Jane's All the World's Aircraft, 1976-77.

The messages used as examples resemble actual messages only in their grammatical structure. Most objects, attributes, and time/location parameters used are related to the events connected with the launch, mission, deorbit, breakup, and impact of Skylab.

ABSTRACT

This final report presents the results of work performed under RADC contract No. F30602-80-C-0047. The effort described involved the further development of OSI's automated data base generation technology and its implementation in a computational environment suitable for further exploration and enhancement of the algorithms.

The report begins with a brief discussion of the intelligence problem which OSI's message processing technology intends to solve, and goes on to summarize the development of the technology thus far. Section 2 offers a summary of OSI's methodological approach to the analysis and description of event reports. This methodology, initially developed on the basis of messages dealing with air activities, has been extended to cover reports of events involving missile and satellite launchings and related events. Section 2.1 offers an overview of related efforts in the area of natural language understanding and complex information processing. Section 2.2 describes OSI's orientation to the problem. Section 2.3 describes our particular formalism for knowledge representation, which forms the core of our methodology. Section 3 covers the message processing and event record generation algorithms used in the current system, MATRES III. It gives an overview of the processing from raw text input to final output of the event record. Section 4 presents a model data base system of event records and describes an experimental program for retrieving these event records. Section 5 looks to the future of the message processing technology, and considers further development in two directions: building a production system embodying the algorithms developed thus far, and further enhancement of the processing power of the algorithms. Examples of the kind of message segments that the system can handle are presented in Appendix A, and some sample event records synthesized from these segments are given in Appendix B. The detailed design, structures, and algorithms of the MATRES III system are described in a companion report [Dwiggins and Silva 1981].

CONTENTS

1.0 INTRODUCTION AND SUMMARY

1.1 Introduction

This final report presents the results of work performed under RADC contract No. F30602-80-C-0047. The work performed under this contract involved the further development of OSI's automated data base generation technology and its implementation in a computational environment suitable for further exploration and enhancement of the algorithms. The following sections briefly discuss the intelligence problem which OSI's message processing technology intends to solve (1.1.1.), and summarize the development of the technology thus far (1.1.2.)

1.1.1 Problem Statement. The task of an intelligence analyst is to predict the future on the basis of information describing what has happened in the past and what events are currently taking place.

At the global level, the questions the analyst asks himself are: "What is happening?" "What does it mean in terms of my knowledge about similar events in the past?", "What is going to happen next"? He is concerned with certain states of affairs, and events signifying changes in these *states of affairs.*

When working with a single message, the analyst seeks answers to at last the following questions:

1. What is its information content?
2. How reliable is the source?
3. How "credible" is the data?

His evaluations of incoming information are based on his cognitive models of certain kinds of situations, the personalities, entities, and processes involved, and the potentialities and constraints associated with changes in an existing state of affairs.

Given the volume of information he must sift, and the complexity of the cognitive models involved, the difficulties of the analyst's task are obvious. Aids to support his analytical processes clearly must involve means for distilling the content of incoming information into a form which is compact, usable, and compatible with his view of the world.

Information on the world situation comes to the I&W analyst mainly in the form of intelligence messages, which are electrically received in an I&W center 24 hours a day. The messages come from many different originators, and are largely in the form of narrative text. The volume of message traffic is extremely high, and in a crisis situation, increases dramatically. Even under normal operating conditions it is very difficult for an analyst to isolate items of information from message text and to assimilate and correlate these items into a pattern of events of indications significance. In a crisis situation, the analyst is completely saturated with data, and the performance of his task demands superhuman capabilities for handling the enormous amount of information which is contained in the message traffic.

A computer, on the other hand, can process large amounts of information. Thus, the notion of offloading some of the processing functions onto the machine seems to provide a logical solution to the information problem.

In order for the computer to process the contents of a message as information, however, that information must be extracted from the text of the message. Currently, that information must be extracted manually for input to an analysis system, which is a tedious and error-prone task, particularly when performed by an analyst with other important demands on his time.

1.1.2 Toward a Solution. For the past several years RADC has been sponsoring an exploratory and developmental program related to the design and development of a general methodology for the efficient and effective exploitation of the content of electrically transmitted intelligence messages. The long term goal of this work is to develop a system which would assist the analyst in creating and maintaining formatted data bases derived from natural language text, and thus offload some of the processing functions from the analyst to the computer. Such a system should provide the analyst with information which is needed for the attainment of his particular goal, i.e., information which is relevant to his task, is of high epistemic standing, and therefore useful to solving his problems.

As mentioned above, the work described in this report is concerned with the analysis of textual reports of events and the synthesis of relevant information elements in a format suitable for automated input to a data base and/or analysis system.

Specifically, the program addresses the problem of automating the analysis of the narrative text portions of intelligence messages describing events, with the aim of transforming them into succinct, formatted, computer processable content representations.

The automated generation of information elements from narrative message text requires that the computer in some sense "understand" natural language text. Within the context of the work described here, we say that a computer system understands an input text insofar as it can construct an adequate representation of the information content of that text. Specifically, we require that the output of the computer understanding process, when applied to some message text, furnish the analyst with at least those information elements that he would himself have extracted from that particular text.

The aim is to model the cognitive activities of the human analyst as he reads and understands message text, distilling its contents into information items of interest to him, and building a conceptual model of the information conveyed by the message.

In order to model this human cognitive activity, the computer must be equipped with representations of both linguistic and extra-linguistic knowledge, and a means of manipulating such representations for the analysis of text and synthesis of information elements. The elements must then be presented in a clear and useful format suitable for the task at hand.

1.2 Summary

The MATRES II system which resulted from OSI's earlier efforts in this area was developed in a hardware/software environment offering only limited capabilities for the kind of development programming needed in such a task. In the course of developing that system, we became familiar with the concept of logic programming [Kowalski 1979] and the Prolog language as an embodiment of that concept [Warren et al 1977, Pereira et al 1978]. We used the Prolog formalism as the basis for implementing our event record generation algorithms.

Since then, a Prolog system became available to us, and we learned of a form of grammar (called "definite clause grammar") which is both natural to Prolog and quite powerful enough to express sophisticated English language grammars [Pereira and Warren 1980]. Using these tools, we have developed and extended the earlier algorithms into a system that works in a unified environment, which is capable of processing amounts of data large enough to do serious development of the algorithms, and which is capable of supporting significant development without the limitations and difficulties associated with the earlier development environment. We also took advantage of Prolog's dual status as programming language and data base system to build a *simple model* of the kind of storage and retrieval capability that can make use of the event records generated by our technology.

There are four major sections to this report. Section 2 offers a summary of OSI's methodological approach to the analysis and description of event reports. This methodology, initially developed on the basis of messages dealing with air activities, has been extended to cover reports of events involving missile and satellite launchings and related events. Section 2.1 offers an overview of related efforts in the area of natural language understanding and complex information processing. Section 2.2 describes OSI's orientation to the problem. Section 2.3 describes our particular formalism for knowledge representation, which forms the core of our methodology.

Section 3 covers the message processing and event record generation algorithms used in the current system, MATRES III. It gives an overview of the processing from raw text input to final output of the event record.

Section 4 presents the model data base system that demonstrates a kind of retrieval that could be appropriate to an event analysis system as described above.

Section 5 looks to the future of the message processing technology, and considers further development in two directions: building a production system embodying the algorithms developed thus far, and further enhancement of the processing power of the algorithms.

Examples of the kind of sentences that the system can handle are presented in Appendix A, and some sample event records from some of these sentences are given in Appendix B.

The detailed design, structures, and algorithms of the MATRES III system are described in a companion report [Dwiggins and Silva 1981].

2.0 MESSAGE TEXT ANALYSIS CONCEPTS

2.1 Introduction

This section describes OSI's automated data base generation technology. This technology has grown out of several RADC-sponsored exploratory research efforts directed toward the development of a general methodology for both interactive and fully automated exploitation of the content of electrically transmitted intelligence messages in order to create and maintain formatted indications and warnings (I&W) data bases. The motivation for this work arises from the acute need to assist the I&W analyst in dealing with message traffic, particularly in an information overload situation.

Daily message traffic consists of messages in which the information can be formatted, (in the form of tables), semi-formatted (a mixture of tables and narrative text), or unformatted, in which case all the information reported is in the form of narrative text. A design concept for the automated analysis of formatted messages and techniques for their conversion into structured data base elements for use in I&W ADP systems was developed by Montgomery and is described in Silva and Montgomery [1977], vol. II. While the difficulties involved in the synthesis of data base elements from formatted messages should not be underestimated, the procedures required are well within the state-of-the-art. The interpretation of unformatted messages, on the other hand, requires the utilization of very advanced and sophisticated analytical techniques aimed at a "computer understanding" of natural language text, for which there are no off-the-shelf, generally applicable working algorithms.

Most active researchers in the area of natural language understanding by computer come from the fields of Computational Linguistics and Artificial Intelligence (AI). In the last decade an impressive array of experimental language understanding systems have been built and applied to various tasks. These systems take fragments of English text as input, and provide a representation of the meaning content of that text as output.

There exist several comprehensive reviews of natural language understanding programs and systems. For example, Wilks [1974] surveys and compares the major projects on the understanding of natural language as they existed at that time. Included in the discussion are Winograd's SHRDLU, Charniak's work on the resolution of pronoun ambiguities in children's stories, Colby's PARRY, Simmons' work on semantic networks, Schank's MARGIE, and Wilks' own work, as reflected in an English-to-French translation system. Another, more recent survey, is that by Kender [1977], who provides us with an excellent *Annotated Bibliography of Natural Language and Speech Understanding Systems*. Kender's bibliography summarizes some 80 papers in the field, and also includes several overviews and criticisms, usually in the form of comparative studies.

Another area which contributes to and benefits from research in natural language understanding is that of "expert" systems. These are computer programs designed to solve specific technical problems, perform medical diagnoses, provide education, training or guidance, and act as "assistants" to humans engaged in some task. Many of them have natural language understanding components which allow users to converse with the system in a natural way during the performance of some task (c.f. [Woods 1977], [Bobrow, D. G. et al. 1977]; and [Brown et al. 1975]).

A list of systems and programs which allow natural language communication with a computer specifically for the purpose of data base query was recently compiled by Waitz [1977], and published in a special issue of the *SIGART Newsletter*, devoted to Natural Language Interfaces.

Research efforts specifically concerned with the automated creation of structured databases from narrative text, however, seem to be few and restricted to applications in the field of medicine. The best-known effort in this area is the ongoing research of the Linguistic String Project at the New York University under the leadership of Naomi Sager. The Linguistic String Project is concerned with the development of techniques for converting medical records written in English into tabular structures called "information formats" from which it will be possible to gather statistics and answer questions ([Hirschman et al. 1976], [Grishman et al. 1978], and [Sager 1978]). In this method, input sentences are subjected to an elaborate syntactic analysis, followed by the application of "formatting transformations" which fill in the rows of the tabular "information formats" mentioned above. The latter constitute the building blocks of the data base.

Another interesting approach to automated database generation is the one taken by the Heuristic Programming Project at Stanford University. Bonnet [1979] describes a computer program designed to understand medical summaries describing the status of patients. In his system, "computer understanding" is guided by abstract data structures called "schemas", which model the way physicians present medical problems in their summaries.

The next section defines OSI's orientation to the problem of automated data base generation from narrative text, and examines the concept of a "computer understanding", which underlies OSI's technology. Section 2.3 examines OSI's approach to the representation of knowledge in the light of the intended application, and discusses two data structures, the *template* and the *event record*, which play an important role in the representation of knowledge about events and the transformation of narrative message text into conceptually structured data elements. OSI's methodological approach is implemented in a computer system, called MATRES, which serves as a testbed for the continuing development of the procedures required for automated generation of structured data elements from the narrative text of intelligence messages. A detailed functional description of MATRES is offered in Section 3. Briefly, MATRES takes sentences of messages as input, analyzes and interprets each such sentence, identifies information elements relevant to a given task, and organizes those elements into compact, computer-processable content representations suitable for input to and update of I&W data bases. The current version of MATRES is implemented in the programming language Prolog, and runs on a PDP-11/70 under the Unix operating system.

2.2 *OSI's Orientation*

Operating Systems' approach to automated database generation is aimed toward the support of information processing functions in the intelligence community. While firmly committed to the utilization of the most advanced techniques, OSI is highly selective in its approach to the solution of given problems. OSI's policy has always been to search for the most sensible and productive ways to aid and support the analyst's mission relative to his particular environment and the type of data with which he works. Applicable information processing techniques, therefore, are evaluated in the light of the needs and requirements of the particular intelligence activities they are to serve.

The OSI approach integrates considerable practical experience gained from many years of involvement with the design and development of sophisticated information-processing tools for the intelligence community with theoretical and practical advances in several disciplines, including linguistics, computational linguistics, artificial intelligence, and advanced database technology.

One of the key concepts underlying OSI's automated data base generation technology is that of "computer understanding" of message text. For the purposes of this work, we say that a computer system "understands" a message text insofar as it can construct a representation of the meaning content of that text which satisfies the information requirements of the prospective class of users. Specifically, we require that the output of the understanding process furnish the analyst with at least those information items that he would himself have extracted from that text during the performance of a particular task.

To achieve a "computer understanding" of message text, an attempt must be made to model the human processes of assimilation, i.e., the computer must in some sense be made to emulate the mental processes of an analyst scanning a text and distilling from it those items of information which are compatible with his current requirements, and therefore relevant and useful to the attainment of his goal.

Understanding language is a very complex intellectual process, and the knowledge of the subprocesses involved is at best fragmentary. Nevertheless, recent theoretical and methodological advances in Cognitive Science -- a new science which combines theoretical and empirical methodologies from computational linguistics, artificial intelligence, philosophy, cognitive psychology, computer science and education -- have led to a number of significant insights which together shed enough light upon the nature of these processes that meaningful investigations can be undertaken. The next few paragraphs give a brief sketch of the theory of comprehension adopted for the purposes of this work.

This theory is based upon the assumption that, when people interpret text, they do so, not on the basis of the text alone, but by assimilating new information into a framework of already existing knowledge which has been derived from personal experience with the real world, processed, structured, and stored in memory, ready to be accessed when required. ([Norman and Rumelhart 1975], [Miller and Johnson-Laird 1976], [Miller 1978]). In other words, this theory claims that the information language conveys, i.e., its semantic structure, is interpreted directly in terms of the human mind's organization of knowledge derived from experience, a level of mental representation which is often referred to as "conceptual structure". This is assumed to be a level of representation at which knowledge acquired from all sources, including language, vision, nonverbal auditory perception, touch, etc., are fused and represented in a uniform manner.

As Rumelhart and Ortony [1976] remind us, this view of language comprehension is not entirely new. Similar ideas are discussed in Bartlett [1932], and much earlier, by Kant in his *Critique of Pure Reason* (1787). Recent discussion of related issues can be found in [Goffman 1974], [Fillmore 1975], [Bobrow and Norman 1975], [Bransford and McCarrell 1974], and [Schank 1975].

In the intelligence environment, the analyst seeks answers to such questions as "What's happening"? "Who is doing what to whom"? "Is the current situation changing"? "Who are the key figures bringing about the change"? "What are the consequences of such changes"? "What do we do about it"? "What are our courses of action?"

An analyst versed in a given subject matter assimilates and evaluates the information communicated by the message, draws conclusions, and produces intelligence using many kinds of knowledge, including his innate knowledge of English grammar, his expert knowledge of the objects, events, and relations characteristic of his subject domain, and his knowledge of the laws and probabilities governing the entities in his domain.

Modeling the processes of assimilation in a computer is not a trivial task. It is obvious that in order to achieve such a goal, the machine must must incorporate many skills. It must be provided with full "language understanding" capabilities modeled upon those of the human reader, i.e., the system must be able to perform a linguistic analysis of text, and subsequently interpret the results within the conceptual framework of a particular subject domain.

The next section examines issues of knowledge representation.

2.3 Knowledge Representation

One of the most important design questions facing the implementer of knowledge-based systems concerns how to structure the knowledge base of facts and rules so that appropriate items can be efficiently stored, accessed and modified. Current approaches to knowledge representation, include formal logic notations, production rules, various network structures, "structured object representations" and various combinations of these methods. "Structured object representations", which in the AI literature are variously referred to as "semantic networks", "frames", "scripts" and "schemata", tend to package information in "chunks". This is done by aggregating several related facts into larger structures that are identified with important objects (entities) in the subject domain of the system.

Brachman and Smith [1980] provide a comprehensive survey of current knowledge representation research in the February, 1980 issue of the *SIGART Newsletter*. Other, more detailed, treatments of current approaches can be found in [Bobrow and Collins 1975], [Rumelhart and Ortony 1976]; [Schank 1975]; [Walker 1978]; and [Bobrow and Winograd 1977]. For a description of OSI's work in this area, see [Kuhns 1974]; [Kuhns et al. 1975]; [Silva and [Montgomery 1978], and [Silva et al. 1979a, 1979b].

We now turn to a discussion of OSI's approach to the representation of knowledge and discuss its appropriateness in the light of the intended application.

OSI's message analysis system is designed to process intelligence reports describing real-world actions, events and situations, their associated entities, and the relations between them. Facts of this kind may be reported with various degrees of uncertainty, and are often incomplete. Several messages may seemingly report on the same facts, but there may be conflicts. If a data base, which no longer consists of the original message traffic but of structured content representations derived from the latter, is to be useful to the analyst, all the original information, including indications of uncertainty, unreliability and incompleteness, must be faithfully preserved.

Another important consideration when deciding upon a representation has to do with the storage and subsequent retrieval of information. In the previous subsection we presented a theory of comprehension which suggests that, when a person thinks of a particular entity, a number of other closely related entities immediately come into focus forming a conceptual whole, or a "chunk" of information. Or, to put it differently, whenever a new concept is introduced into a communication, that concept introduces a number of related concepts into the local context or foreground ([Chafe 1972]; [Miller 1978]). If our approach is to be consonant with the human way of processing information, it would seem that, when information is needed about a particular event, situation or object, all of the relevant facts about it should be retrieved as one coherent package.

The structured object representations mentioned previously have a number of properties which render them particularly appropriate for the representation of event-oriented data. As a consequence, they were adopted as the basis for the development of *Templates* and *Event Records*, two related types of data structures which play a particularly important role in OSI's approach to the representation and derivation of meaning from message text.

Templates and Event Records

Taking the *event* as the primary unit of analysis, OSI has developed the concept of a ***template*** as an organizing principle for the abstract characterization of event classes, their associated objects, and the processes which act upon them.

The structure of templates is based upon the principle of "information chunking" mentioned above. The main function of a template is to group event-related "chunks" of knowledge into a single data structure in a coherent and efficient way. In this sense, templates resemble the artificial intelligence notions of *prototypes*, *frames*, *scripts* and *schemata.*

Templates have two major components: a descriptive component, and a procedural component. The descriptive components of templates provide a framework for the representation of knowledge about a given subject domain approximating that which a human has. Each individual template contains invariant knowledge about an event class (or a class of objects associated with an event). Each such class of entities is described from the point of view of the prospective analyst-user community in terms of a set of parameters which an analyst normally perceives as being associated with that class of entities at the level of detail required for the performance of his particular task. Thus, a template describing the class of "launch" events includes parameters such as the objects associated with the launch, the launch site, the launch system and the time of the launch.

The procedural component of templates consists of mapping rules which play an active role in the processes which transform narrative text into event records. Thus, the template for a particular event class embodies the necessary procedural knowledge to construct a representation of an individual event of that class from a sentence describing such an event. An important part of this information consists of what linguists call the "selectional restrictions" or "selectional preferences" in a particular domain.

To illustrate this notion, consider Table I, which illustrates the kinds of information incorporated in the "launch" template, as used for the automated interpretation of

Table 1. Informal Description of the LAUNCH Concept in the Missile and Satellite Domain.

Descriptive Elements			Procedural Elements
Descriptor	Filler Specification	OBL or OPT	Procedures for filling slots
Agent	If expressed, then logical subject of sentence. Head noun with feature NATION	OPT	If conditions hold, fill Agent slot with subject noun phrase
Object	If no Agent, then Object in logical subject position; otherwise in object position. Allowable features: MISSILE and SATELLITE	OBL	According to which conditions hold, construct Object template from either subject nounphrase or object nounphrase
Launchsys	Either subject nounphrase with headnoun with feature BOOSTER, or PP with prep BY and headnoun with feature BOOSTER	OPT	Test headnoun of subject for feature BOOSTER Search VMODS list for specified prepositional phrase
Launchsite	PP with headnoun (+LOC)		Search VMODS list for specified constituent
Inclination	PP with headnoun (+INCL)		Search VMODS list for specified constituent
Destination	PP with Preps TO or INTO and headnoun with feature LOC		Search VMODS list for specified constituent
Time	1. ADV (TYME REF) or 2. PP with TYME prep and headnoun with feature TYME	OPT	Search VMODS list for specified constituent
Date	PP with DATE-node	OPT	Search VMODS list for specified constituent

message text describing missile or satellite launch events.

The entities associated with the central concept, i.e., the agent who launched the object, the object itself, the orbit, the launch system, the launch site, etc., are represented by means of template "descriptors". The latter roughly correspond to the semantic "cases" of linguistic deep case theory ([Fillmore 1968, 1971]; [Chafe 1970]). Cases, in the linguistic sense, are the relations which connect nominal concepts to verbs. The "case" relations used in the work described here are derived from a representative sample of message traffic by means of an in-depth "intellectual" analysis. They name the relations which hold between the various components of a complex entity in a given task domain in terms familiar to the analyst. The cases therefore differ somewhat from those proposed by Fillmore or Chafe, in that they are more specific to the knowledge domain of interest. A case analysis, as reflected in a template, therefore, is compatible with the intellectual analysis an analyst would have arrived at.

Table 2 illustrates the notion of an *event record*, which is the data structure used for the description of specific events. The information stored in Table 2 represents the event described in sentence (a) below.

Sentence (a):

ON 14 MAY 1973, THE SKYLAB ORBITAL WORKSHOP
WAS FIRED FROM THE KENNEDY SPACE CENTER BY A
SATURN-6 TYPE LAUNCHER.

Table 2. Event Record representing the information contained in Sentence (a)

```
Event: launch
  Agent=agentunk
  Object:satellite
    Equipment=skylab orbital workshop
    Quantity= one
  Launchsys=by a saturn-6 type launcher
  Launchsite=from the kennedy space center
  Time=
  Date=on 14 may 1973
```

Event records are related to templates as the description of an individual is related to the description of its class. In other words, each event record describes one *individual*, i.e., a unique member of a class of individuals in the world being modeled.

Thus, while the *launch* template characterizes a class of events in a general and abstract way, the event record representing the information in sentence (a) describes a *specific* launch event involving a particular spacecraft (e.g., Skylab), launched from a particular launch site (the Kennedy Space Center), by a particular launch system (a Saturn-6 type launch vehicle), on a particular date (14 May 1973). In the terminology of logic one would say that the relation between a template and its corresponding event records is roughly the same as that which holds between an *intensional* description of a concept and an element of its *extension*. Thus, the set of event records describing events of the same class, i.e., event records related to a particular template, constitutes the *extension* of the concept described by the template.

Event records are derived from an input text by procedures attached to individual descriptor slots. These procedures match the input against the appropriate template in an attempt to "fill in" text-derived values for the descriptors of the currently activated template. Consequently, event records, like templates, are event-centered structures in which the information conveyed by the input can be viewed from the perspective of time, location, type of activity, object(s) involved, etc. They provide content representations for individual sentences describing atomic events, and form the building blocks of message content representations which, in turn underlie data base element synthesis.

Relations between the various entities associated with an event, i.e., the set of relations connecting descriptors to the main concept within a template, are referred to as *intra-template relations*. Intra-template relations can be used to alert the analyst to missing information. To see this, consider the event record for sentence (a). For the event record to be complete, a value must be found for all the descriptor slots of its corresponding template. Notice, however, that the message fragment represented by sentence (a) is incomplete in terms of the "launch" template specification outlined in Table 1. Conspicuously absent are values for the Agent and Time descriptors. Since sentence (a) contained no explicit mention of who launched the spacecraft, but the sentence was passive, the Agent slot was given the default value of "agent-unk" (agent unknown) by the system. The Time slot, on the other hand, was left empty.

The concept of Time, because of its complexity in the intelligence environment, also requires a separate template for its abstract description. Time is not always explicitly stated in a message. It and must often be computed from the tense of the verbs occurring in the message, and such complex natural language expressions as "AT 10 MINUTE INTERVALS FOR A PERIOD OF 6 HOURS". Moreover, time operators such as "currently", require reference to information in the message header or the message text for their resolution. The absence of any expected element in an event record can be used to inform the analyst that some description of an object or event is incomplete.

While the *intra-template* relations described above define the internal composition of concepts, *inter-template* relations specify the relationships between concepts. Inter-concept relations are also represented by templates. For example, the temporal relation of precedence (often expressed in language as a prepositional phrase with the preposition "before") is defined as a template which links two event types: event type A normally takes place before event type B, where each event type is itself represented by a template.

Note that, as opposed to intra-template relations, inter-template relations are constrained by the physical laws and normal expectations of the world being modeled. Examples of inter-template relations are causal relations, temporal succession relations, and other implicational relations between events. A spacecraft which is successfully launched will normally attain orbit, remain in orbit for a certain number of revolutions, and subsequently deorbit, reenter the earth's atmosphere, break up, burn up or crash to earth, in which case it can be recovered. Inter-template relations, as intra-template relations, can be used in a predictive capacity.

To sum up, templates are models of objects, events, and situations habitually encountered in the domain of application. They provide the means of coding the analyst's knowledge of his task domain in terms of logical structures which are easily transformed into representations susceptible to automatic processing. They are active memory structures which embody hypotheses about facts, processes, operations, procedures and computations required to characterize the links between form and content. Procedurally, they can best be described as the fundamental knowledge structures which mediate the correlations between syntactic structures and their corresponding information content.

Templates can be helpful in discovering inconsistencies both in the input and in the system's model of the world. If information derived from the input cannot be fitted into one of the templates stored in memory, it may mean that there are inconsistencies. For example, finding that a missile is reported as being in orbit, is inconsistent with the expected actions a missile is designed to perform. On the other hand, the inability of the system to account for the input may also mean that the structure of the model is incomplete: either a component is missing, or that this is a possible but atypical situation for which there is no representation in the system.

Each subject domain yields its own template inventory corresponding to the events and objects and their internal and external relations -- optional or obligatory -- which have informational significance within that domain. In the Missile and Satellite domain, some of the key concepts for which templates have been constructed are: "launch", "deorbit", "reentry", "breakup", "impact", "missile", "satellite", and "DTG" (date/time group).

Event records, on the other hand, contain descriptions of individual events, objects, and their properties. They have several important properties which render them particularly useful as a support tool for the I&W analyst:

- They present information to the analyst in the form of a structure which is compatible with his view of the world and are therefore easier to read and digest than unstructured text. It is suggested that this should maximize the analyst's ability to monitor and assimilate incoming data.
- They are usable for the construction of an analyst's data base, which allows flexible retrieval of information not only by event type, but also by other associated parameters, such as object(s) involved in the event and time and location indicators.
- The information stored in these data structures lends itself readily to further processing. This processing may be statistical in nature, or may be part of the general inference making mechanisms of the message data base.

It is our contention that adopting templates and event records as representational constructs in the data acquisition component of a message data base has significant advantages over the use of production rules, and can lead to powerful, effective, and efficient systems.

As explained earlier, templates represent the system's current hypothesis about how to interpret incoming message data. The procedures associated with template slots are executed for the purpose of verifying the current hypothesis. Execution of procedures is thus strongly constrained by the currently activated template. In a production system without such structures, rules are generally executed in a mode which may cause irrelevant rules to be tried. A model-based system, therefore, is potentially faster because needless rule invocation tends to be minimized.

3.0 MESSAGE PROCESSING IN THE MATRES III SYSTEM

The principles and data structures discussed above are implemented in the MATRES Message Analysis System, which specifically addresses the problem of automating the analysis of the narrative text portions of intelligence messages describing events in the Air Activities and Missile and Satellites domains. The analytical procedures employed by MATRES automatically identify the topics and entities referred to in a text, including the meaning relations which hold among those entities. In the process, they transform verbose message text into succinct, computer-processable content representations, by performing a lexical, syntactic, semantic, and pragmatic analysis of the text. The outputs of this analysis are the "event records" described in the previous section. They serve as a basis for the construction of the analyst's data base.

The analytical procedures of MATRES utilize two major knowledge sources: (1) a model of the sublanguage that characterizes the domain of application and (2), a model of the entities and relations characteristic of this domain.

The term *sublanguage* is used to refer to the specialized usage of English in a particular domain of discourse. The sublanguages used in message traffic differ from general English in several respects. First, they are restricted, in the sense that they normally use a small and specialized vocabulary. Second, they consist only of declarative sentences (or fragments thereof); there are no questions or commands. Furthermore, such reporting languages are usually characterized by heavy use of abbreviations, and often employ grammatical constructions which deviate from those of "normal" English, (e.g., dropped articles, dropped prepositions, and idiosyncratic usage of words. As an example of idiosyncratic vocabulary usage, consider the verb "deploy". In ordinary discursive prose, this verb is normally used with an animate agent: somebody deploys something. In messages reporting on aircraft activities, aircraft, although inanimate, and therefore not "agentive" in the strict sense of the word, deploy themselves. Also, sentences are mostly passive; constructions with the agent expressed in surface structure occur infrequently. The differences are often so marked, that it turns out to be more convenient to write specialized grammars for these sublanguages, than it would be to adapt a grammar written for "normal" English. This was, as will become apparent later, to a certain extent the case in the work described here.

The basic unit of analysis adopted for this work is the "Event Report", which is described in detail below.

The two subject domains which have served as a basis for the development of MATRES are aircraft activities and missile and satellite events. Although the automated database generation methodology incorporated in MATRES was developed on the basis of the somewhat restricted subject domains just mentioned, it is presumed to be extensible to applications of greater scope than the current one.

3.1 Current Domain of Application.

The current domain of application of MATRES covers a type of text which describes observable events involving objects such as aircraft, missiles, satellites and spacecraft, whose activities normally fall into predictable patterns.

Of fundamental importance in this work is the notion of *EVENT*, which is defined as the aggregate of all the states, processes, and actions associated with an object or a set of objects from the inception to the termination of some larger activity. To illustrate this notion, consider the following EVENT REPORT, which describes the launch of Skylab and the various actions it was involved in from the time it was launched to the time it reentered the earth's atmosphere and impacted in Western Australia.

> Skylab was launched from the Kennedy Space Center at approximately 1 pm on May 14th, 1973. A Saturn-5 type launch vehicle was used to inject the spacecraft into an orbit inclined at 50 degrees to the equator. Skylab incurred serious damage at lift-off time and was later repaired by the astronauts previously scheduled to rendezvous with it. The astronauts, launched in a modified Apollo service module, attached themselves to Skylab by "umbilical" cords (connections to life support systems), and salvaged the ailing spacecraft. Subsequently, Skylab was used for experiments, and was later abandoned. Recently, it caused much concern, because it was obvious Skylab was going to deorbit and crash to earth. It did indeed reenter the earth's atmosphere, upon which it began to break up into pieces, some of which burned up on reentry. Others impacted in the Indian Ocean, and yet others landed in Western Australia and were recovered.

The text of this EVENT REPORT was artificially composed on the basis of articles which appeared in the Los Angeles Times, Newsweek, and Aviation Week, between 1973 and 1979. Normally, EVENT REPORTS comprise several intelligence messages received over a period of time, rather than a single paragraph.

Events of the kind described above usually have a complex internal structure composed of smaller interrelated units of action -- in this case the launch, the deorbit, the impact, etc. These smaller units are referred to as *atomic events*. Atomic events are the basic units of description at the event level. An atomic event cannot have another event as a component. The internal structure of atomic events reflects the participants in the event and their relations to the central action.

3.2 Prolog Representation of Templates and Event Records.

Our current domain of discourse is characterized by aircraft, missiles and satellites, their respective properties, the purposes for which they can be used, the actions that can be performed on them, the actions in which they can participate, their particular role in each event type, and the relations between such events in the context of a larger, coherent activity.

To model the objects, event classes and relationships of our domain, we defined a set of templates which embody the system's knowledge of the possible objects, events and relationships in the domain of application. At the same time, they contain the procedural knowledge necessary for mapping input sentences onto data structures called *event records*.

In our system, templates are implemented as Prolog procedures, while event records take the form of Prolog "terms". As mentioned above, the template for a particular class of atomic event embodies the necessary information to construct an event record from a sentence describing an event of that class. Thus, the template shown in Table 1, when applied to sentence (1), yields the event record shown in Table 2. Each event record names the atomic event represented, and records those

(1) The Skylab orbital workshop, a converted
S-4B third stage from a Saturn-5 launch
vehicle, successfully deorbited into
the Australian outback on 12 Jul 1979.

Table 1. DEORBIT Template

```
  construct(deorbit, s(Tense, Voice, Subj, Vbgr, Obj, Compl, Vmods),
                  [OBJ, AG, Loc, Rev, DTG]) :-
        semobject(Subj, Obj, OBJ),
        agent(Subj, Vmods, AG),
        location(Obj, Vmods, Loc),
        revolution(Vmods, Rev, Vmods2),
        construct(dtg, Vmods2, DTG).
```

Table 2. Event Record for Sentence (1)

```
Status=successfully
Event: deorbit
  Object:satellite
    Equipment=skylab orbital workshop
    Quantity= one
    Add_Attributes= a converted s-4b third stage
                 from a saturn-5 launch vehicle
  Location=into the australian outback
  Revolution=
  Time=
  Date=on 12 july 1979
```

properties which were explicitly expressed in the input sentence.

To achieve a content representation for an entire Event description, a higher-level template for the abstract representation of larger event types was defined. Such a template represents characteristic patterns of atomic events in a given domain of application. When applied to a coherent text, it constructs a meaning representation reflecting the conceptual organization of the text as a whole.

Table 3 shows a possible Prolog representation of a higher-level template, which, when implemented, will serve to derive meaning representations from a class of event descriptions resembling the *Skylab* text.

The following section illustrates the processes involved in the computer "understanding" of message text.

3.3 Principles of Message Text Analysis.

From the theoretical viewpoint, the primary goal of natural language understanding is to arrive at the *total* meaning content of a text. This is usually taken to include all

Table 3. Prolog Representation of SPACECRAFT ACTIVITY Template

```
assemble(spacecraft_activity, [list of event records],
                      [connected discourse structure] :-
          launch(.........),
          achieve_orbit(......),
          deorbit(......),
          reentry(......),
          breakup(......),
          impact(......),
          recover(.....).
```

the facts which are explicitly recorded plus those facts which can be inferred from the meaning of the words appearing in that text and their syntactic and semantic interrelations. Such a goal may lead to meaning representations which are overburdened with detail, which would be irrelevant in the context of this work.

A human analyst is selective. He does not seek to extract *all* the information a text may contain, but only that which is is needed for the performance of his task. The goal of this work, then, is to disregard irrelevant detail; message text is transformed into content structures reflecting the analyst's view of the world.

The logical unit of analysis adopted as a basis for describing intelligence information is the EVENT REPORT.

The OSI natural language processing system (MATRES) is based upon a process model of taxt understanding involving four sets of operations.

First, the sentences of a message text are parsed into a set of propositional structures. The propositions are linked by various semantic relations which may be explicitly expressed in the surface structure of the text, or inferred during the interpretation process on the basis of contextual and/or real world knowledge.

Second, the resulting set of propositions are organized into higher-level conceptual categories, namely, event representations.

The first two operations described above are implemented in the current version of OSI's message text analysis system MATRES, and will be described in detail in the next section.

A third set of operations links the resulting event representations into a coherent whole, reflecting the meaning of the message text as a whole.

Finally, when all messages constituting an EVENT REPORT are processed in this manner, a set of constraints checks the coherence of the set of messages constituting an EVENT REPORT at the global level, i.e., at the level of the EVENT REPORT.

Many of the subprocesses involved in language understanding are still largely unexplored and it is therefore not possible to construct a comprehensive model of language understanding. For example, no complete specification for the many and complex inferential processes involved in language understanding can be given at this stage, although work in progress in the fields of Artificial Intelligence and

Cognitive Science is very promising.

3.4 Analytical Processes.

3.4.1 Sentence-Level Analysis In the current implementation of OSI's message text analysis system (MATRES), the understanding process begins when a sentence of the message text is input to the system. This can be done either from a terminal, or from a disk file. As a first step in the analysis, each sentence of an input text is parsed by a syntactic processor implemented directly as a Definite Clause Grammar (DCG) [Pereira and Warren 1980].*

The current scope of the MATRES sentence grammar covers declarative sentences with simple and complex noun phrases, sentential and infinitival complements, and a large range of adverbials. The grammar currently comprises 80 DCG rules of the form

nt --> body.

where *nt* is a non-terminal symbol with one or more arguments, and *body* is a sequence of one or more *goals* separated by commas. This is illustrated by the rule below, which builds the parse tree (S) of an input sentence.

```
sentence(S) -->
/* preposed adverbials */   adverbials(PreAdv),
/* grammatical subject */   noun_phrase(G_Subj),
/* evaluative adverb (e.g., probably) */   evaladvb(Evaladvb),
   verb(Verb,Tense),
   rest_sentence(PreAdv, G_Subj, Evaladvb, Verb, Tense, S),  {! } .
```

The associated lexicon takes the form of Prolog clauses. Examples of lexical entries are:

```
adj(central).
noun(satellite).
verbform(launched,  launch,  past).
```

The output of this stage is a shallow parse tree with mostly standard grammatical categories (s, np, pp, n, v, adv, etc.) at nodes. The grammar also contains a few nonstandard definitions of syntactic categories which reflect idiosyncratic usage of language in our particular domain of application, and were introduced to expedite the parsing. Their main function is to help avoid false parsing paths. Some nonstandard phrasal categories (e.g., *tm* for "time phrase", and *mp* for "measure phrase") can be observed in Table 4b, which displays the internal representation of a sentence after syntactic analysis.

The parser is currently constrained so as to provide only one analysis per sentence. This constraint was implemented by using the Prolog "cut symbol", which restrains the sentence grammar procedure from returning any but the first parse; the grammar itself is capable of returning multiple parses.

---

* The Definite Clause Grammar formalism described in Pereira et al. is based upon Colmerauer's *Metamorphosis Grammar*, the original grammar formalism for natural language analysis using definite clauses [Colmerauer 1975]. We chose the DCG formalism as a model for this work, because of its close relation to Prolog and its expressive power, as described by Pereira and Warren.

3.4.2 From Parsed Sentences to Atomic Events The next step in the analysis of a sentence is to derive its meaning in terms of *atomic events*, where an atomic event corresponds to a simple proposition referring to an action which changes the prevailing state of affairs. To achieve this objective, the program selects and activates the appropriate template on the basis of certain characteristics stored in the lexical entry of the main verb of that sentence. Control now passes to the template, which actively seeks "fillers" for its "slots" to construct a content representation in the form of one or more event records. These "fillers" are selected by the syntactic analyzer on the basis of syntactic, semantic and pragmatic constraints on the constituents of the parse tree output.

The interpretive component of MATRES consists of a module for syntactic normalization, seven event templates and their associated procedures, two object templates, and a template for the date/time concept. If one includes the top-level controlling procedures which initiate the semantic interpretation of a parse tree, the total number of Prolog clauses used by the interpretive module is close to 100.

An Illustrative Example

The following is an example of how our programs interpret the complex sentence (4) which describes an expected *deorbit* event. The internal representations shown are explained in full detail in the program documentation [Dwiggins and Silva 1981].

(4) The two 81 degree probable second generation satellites,
which were launched from the Kennedy Space Center
earlier today, were expected to deorbit over Western Australia
just northwest of Kalgoorlie at 1330 hours this date.

Table 4a. Internal representation of sentence (4) after being read in.

```
[the,two,nb(81,2),degree,probable,third,generation,satellites,,,
which,were,launched,from,the,kennedy,space,center,earlier,today,,,
were,expected,to,deorbit,over,western,australia,
just,northwest,of,kalgoorlie, at,nb(1330,4),hours,this,date,.]
```

The list of words shown in Table 4a constitutes the input to the parser. In our system, sentences have seven immediate constituents: tense, voice, logical subject, verbgroup, logical object, complement, and a list of adverbials. Table 4b shows the syntactic structure assigned to sentence (4) by the parser.

Notice that the parser converts the passive input sentence of Table 4a into its active form and supplies the word "agentunk" (agent unknown) for the missing subject of "expect". The same procedure is applied to the relative clause, "which were launched from the Kennedy Space Center earlier today". Furthermore, clauses with "extraposed" subjects, such as the infinitival complement "to deorbit over Western Australia....", are also assigned propositional structure. Accordingly, in the analysis of sentence (4), the entire complex noun phrase describing the two satellites in question is copied to become the logical subject of the verb "deorbit". There are, however, cases of extraposition which are not easily handled by a DCG grammar. For a discussion of extensions to the DCG formalism specifically designed to handle cases of extraposition in a general way, see [Pereira 1980].

Table 4b. Internal representation of sentence (4) after syntactic analysis

```
s(
  tns(past),
  voice(psv),
  np([],[],nnode(agentunk,[]),[]),
  vg([],expect),
  np(dp([],the,nbr([],two)),[mp(nbr([],nb(81,2)),u(degree,angular)),
     probable,third,generation],nnode(satellites,[]),
       [s(
          tns(past),
          voice(psv),
          np([],[],nnode(agentunk,[]),[]),
          vg([],launch),
          np(dp([],the,nbr([],two)),[mp(nbr([],nb(81,2)),u(degree,angular)),
            probable,third,generation],nnode(satellites,[]),[]),[],
          [pp([],from,np(dp([],the,[]),[kennedy,space],nnode(center,[]),[])),
          earlier,today])]),
  s(
    tns(present),
    voice(act),
    np(dp([],the,nbr([],two)),[mp(nbr([],nb(81,2)),u(degree,angular)),
       probable,third,generation],nnode(satellites,[]),
         [s(
          tns(past),
          voice(psv),
          np([],[],nnode(agentunk,[]),[]),
          vg([],launch),
          np(dp([],the,nbr([],two)),[mp(nbr([],nb(81,2)),u(degree,angular)),
            probable,third,generation],nnode(satellites,[]),[]),[],
          [pp([],from,np(dp([],the,[]),[kennedy,space],nnode(center,[]),[])),
          earlier,today])]),
    vg([],deorbit),[],[],
    [pp([],over,np(dp([],[],[]),[western],nnode(australia,[]),[])),
    pp([just],northwest of,np(dp([],[],[]),[],nnode(kalgoorlie,[]),[])),
    tm(at,mp(nbr([],nb(1330,4)),hours)),
    np(dp([],this,[]),[],nnode(date,[]),[])]),[])
```

There are a number of points worth noting regarding the interpretation of the various syntactic components of sentence (4).

First, the higher predicate *expect* is interpreted as an event status indicator, and as such is accepted as an appropriate filler for *Status* operator modifying the deorbit template (see Table 4c). Other verbs indicating status are *fail*, and *continue*. Verbs of communication taking sentential complements are interpreted as signaling the source of the report, while "speaker-oriented" adverbs such as "successfully", "probably" and "possibly", contribute to the "certainty" of the entity with which they are associated.

Table 4c. Event Record for sentence(4)

```
Status=expect
  Event: deorbit
    Object:satellite
      Equipment=81 degree probable third generation satellites
      Quantity=two
      Rel_event=
          Event: launch
            Agent=agentunk
            Object:satellite
              Equipment=81 degree probable third
                    generation satellites
              Quantity=two
            Launchsite=from the kennedy space center
            Destination=
            Time=earlier
            Date=today
    Location=over western australia just northwest of kalgoorlie
    Revolution=
    Time=at 1330 hours
    Date=this date
```

Second, both the relative clause embedded in the subject noun phrase ("which were launched from the Kennedy Space Center") and the infinitival complement ("to deorbit over Western Australia just northeast of Kalgoorlie at 1300 hours this date") describe atomic events in the sense defined previously, and are therefore transformed into a event records. The "deorbit" event described by the infinitival complement is interpreted as the main event reported in the sentence. The "launch" event described by the relative clause, on the other hand, refers to a previously reported event, and is interpreted as a related event. It becomes the value of the *Rel_event* attribute of the *Object* template. Relative clauses which do not describe events, but represent additional properties of the head noun are interpreted as *Other Attributes.* Since such attributes were not present in the sentence under discussion, they do not appear in its event record (Table 4c).

This concludes our brief sketch of the processes involved in the analysis of independent sentences. These processes are repeated until all sentences of the text are transformed into event records. From the above, it can be seen that the number of event records constructed for a particular sentence depends on the number of atomic events described by that sentence. Thus, while simple sentences yield one event record, complex sentences yield event records in which some of the slots contain other event records.

As the sentences constituting an event description are processed in the manner described above, the resulting event records are concatenated into a list. The latter constitutes the output of the sentence-level analysis, and also serves as input to the text-level interpretive procedures.

3.4.3 Beyond the Atomic Event At this point, a higher-level template (not yet implemented) is activated. It takes as input the list of event records derived by the previously described processes, and assembles them into a connected discourse structure reflecting the meaning of the text as a whole. A first approximation to such a template was shown in Table 3.

It is at this level that some anaphoric references are resolved, and temporal, causal, and other text-level relations between atomic events are established. It is important to note, however, that not all types of anaphora can be resolved with currently known techniques.

The Prolog formalism lends itself well to the expression of certain text-level relations. For example, returning to Table 3, observe that the goals in the body of the "assemble" clause embodying the "spacecraft activity" template are ordered so as to reflect the physical laws governing the permissible succession of atomic events within the larger event class represented by that clause. Thus, the goal ordering in Table 3 reflects the constraint that a spacecraft can only achieve orbit after it has been launched, and thereafter must deorbit before it can crash to earth and impact. Moreover, by making some of the goals optional, we can express two important facts. One: not all atomic events conceptually associated with a larger event need necessarily take place, as a deorbited satellite need not necessarily impact; it can burn up upon reentry and never crash to earth. Two: not all actions performed by a spacecraft need be explicitly recorded in a text reporting on the activities of that spacecraft: a first report may mention a launching, and a subsequent one an impact. The deorbit, which, according to the physical laws governing this type of event must have taken place, is implied.

It is important to note at this point that modeling the permissible temporal sequence of atomic events is only a first step in achieving an adequate representation of discourse structure. A great deal more needs to be done if all relevant information is to be represented. Interpretive procedures required at this level, including inferential procedures for establishing coherence at the text level within our domain of investigation are currently under study.

4.0 MATRES III MODEL DATA BASE SYSTEM

In order to demonstrate the capabilities of the technology described in the previous sections for event record structuring within a more complete environment, we have added to the system a simple data storage and retrieval capability, emulating, to a limited extent, the type of data base and analysis system that could best use the type of event *record structure* that MATRES III generates. Such a system must be rather different from a standard data base system, since an event record may have repeated slots of the same type and optional slots, along with hierarchical relations among event records. In addition, event reports of the type handled by MATRES III may contain incomplete, uncertain, and/or conflicting information. Also, matching queries against components of an event record is more complicated than simple identity testing; numeric or semantic similarity must be considered if the intent of the query is to be well carried out. In this system, we present a simple model of retrieval of event records according to a set of matching criteria, each peculiar to the type of slot under consideration.

4.1 Function

The basic functioning of our model data base system is as follows: sentences are processed and event records (ERs) are displayed and stored The retrieval phase searches the stored ERs against user queries. A query consists of a sentence like those processed in the data base generation phase, but allowing particular components to be replaced by slot names bounded by '<' and '>', e.g.

<Object> deorbited on 12 july 1979.

Skylab was launched by <Launchsys>.

The missile impacted before <Time> on <date>.

The saturn-5 flew to <Location>.

These "open sentences" are parsed into Query Records (QRs), which are then matched against the stored ERs; those ERs that match are displayed in their entirety. The following sections discuss the structure of QRs and the matching process; the detailed structure of ERs is described in [Dwiggins and Silva 1981].

4.2 Open Sentences and QRs

4.2.1 Slot Names The only structural difference between normal sentences of the corpus and open sentences is the presence of "slot names" in open sentences. As indicated above, a slot name is word enclosed in angle brackets, e.g. "<Location>" or "<Date>". The following list shows the slot names currently recognized by the system, along with the syntactic contexts in which they may occur.

Agent

represents the human or institutional agent causing the action of the given satellite or missile; it may occur as the subject of an active sentence whose object is a satellite or missile, or in a "by"-phrase in a passive sentence of that type (e.g. "<Agent> launched ..." or "... was launched by <Agent>").

Date

can occur as the object of a date-related preposition.

Destination

can occur as the object of a directional preposition.

Flightsource

can occur in a "flight"-type sentence as the object of "from".

Inclination

can occur in a "launch"-type sentence as the object of "on".

Infosource

must be the subject of a verb which takes a sentential complement, e.g. "announce [that]", "expect [that]".

Launchsite

can occur in a "launch"-type sentence as the object of "from".

Launchsys

can occur in a "launch"-type sentence as the object of "by", or as the subject.

Location

can occur in most types of sentence, as the object of a directional or locative preposition or as the direct object of certain verbs such as "reenter".

Object

can take the place of a noun phrase denoting a missile or satellite in the subject or direct object of a sentence.

Revolution

takes the place of a noun phrase such as "revolution 123" or "the 45th revolution" in a prepositional phrase.

Time

can occur as the object of a prepositional phrase appropriate to time.

Slot names have two functions in querying: first, they may be used to fill obligatory syntactic roles to make a sentence parsable, as in the first example sentence in Section 1 (in this example, the user presumably wished to retrieve any ER concerning a deorbiting event on 12 july 1979, regardless of what deorbited); second, they allow the user to specify the presence of slots in ERs to be retrieved, without restricting the contents of those slots.

4.2.2 Processing Open Sentences An open sentence presented as a query will create a QR with the same structure as an ER, except that the slot filler corresponding to a slot name in the sentence will have the value "[]", which will be taken as matching anything in the corresponding place in the ER.

To extend the message processing portion of the system to process open sentences and generate query records required only slight extensions: the "morphology" section was modified to recognize slot names and create special lexical items for them; these are recognized by the grammar in the appropriate places, as indicated above; finally, a few template matching predicates had to be extended to recognize these items.

4.3 Implementation

A generated event record is stored essentially as a set of slots, with markers to cross-reference slots and event records. This scheme allows direct matching of corresponding ER and QR slots. An ER will match a QR if and only if all QR slots match the corresponding ER slots. Thus, the QR slots may be a "subset" of the ER slots, but not vice versa. This scheme was chosen as an a priori reasonable approximation to what a user might desire to retrieve via a given query.

5.0 FUTURE DIRECTIONS

MATRES III, as an embodiment of OSI's automated data base generation technology, provides a clear demonstration of the validity and operational feasibility of this technology. It is, of course, a limited system, designed within the constraints of a single process running on a timeshared minicomputer, and written in a programming language which has been designed more to give easy expression to complex algorithms than to fully exploit the power of a given computer. Nevertheless, it it fully functional on its limited domain, and provides an excellent base for further development of the message processing and data base technology.

In this section, we consider the possibilities for further development in two directions: first, to bring the current technology to the point where it could operate, in a production environment, as a front end to a message content analysis system, and could handle live message traffic at normal (and peak) traffic loads. Second, we discuss the development directions which would allow the system to operate on a wider class of input, and with more power in its automatic processing.

At a minimum, a production version of the system should also be able to extract header information that could be useful in processing message text, e.g. date/time, source, references, content identifiers. This technology is well understood, and is utilized in OSI's message handling systems; such capabilities could thus easily be added to the MATRES design whenever a production system development is required. A production version of the system must also be able to deal with such characteristics of live traffic as misspellings, unexpected information, totally unprocessable sentences (e.g. comments on political implications), etc., without losing its context. Although much of this could be automated using state of the art technology (e.g., spelling correction systems), a human partner will be required as backup to handle the intractable material in the foreseeable future, since new problems will be encountered as additional subject domains and capabilities are added to the system. Integrating a human partner in a problem-solving mode would require a sophisticated error and problem diagnostic capability, as well as a complex interaction facility between the user and the system's knowledge base.

Depending on the requirements of the content analysis system, the current approach to event record structuring and generation might require considerable modification. It is possible to envision two different approaches such modifications might take. One approach would be based on a complete analysis of every sentence in each message, while another would use the templates to drive a selective analysis of key segments of the messages, searching for template-specific types of content. With the first approach, a traditional data base system would place considerable requirements on the system to resolve pronouns and other referential constructs, as well as inter-event relationships, to represent the semantics of a message within the structural constraints of such a system. The second approach would be aimed at covering more material but with less precision, leaving gaps that the user would have to fill.

Finally, in order to handle realistic traffic loads, the system may well need to be programmed at a lower level, with efficiency as a prime objective. Indeed, in order to handle a variety of reasonably complex event types with acceptable speed, a fast processor with a large primary memory could be required, and considerable attention paid to efficient representation of algorithms and data.

On the theoretical side, we have taken a fairly conservative approach to date, using mostly well established techniques of parsing and semantic interpretation, with a trivial approach to morphology and lexical problems. Considerable improvements are possible in all the analytical components of the system. Specifically, within areas:

Morphology: Currently, we make no attempt at a real morphology, i.e. using morphological analysis to supplement dictionary lookup, and thus reduce dictionary size and simplify the job of extending the system's vocabulary. In addition, military language is rife with abbreviations and terminology whose meaning is, at least in part, discernable from their form. Even in the case of an unknown word, a good morphological analysis can often give a clue to its syntactic role through affix analysis.

Syntax: Our current algorithm is strictly left-to-right and top-down. In addition the grammar is essentially coded as part of the program. In the effort to automatically handle mistyped or genuinely new words or phrases, it would be highly desirable to be able to analyze first those pieces of syntax that come easiest, and use these to aid in parsing the remainder of the sentence. In addition, once one or a small set of possible templates has been identified as applying to the sentence, such semantic information could be a considerable aid to processing difficult areas, or at least in identifying them for the human partner.

Template Filling: As mentioned above, the template selection and filling process could be integrated with syntax analysis (and possibly even morphology) to their mutual benefit. In a restricted sublanguage such as we are dealing with, considerable sharing can be done between syntactic and semantic analysis.

Text Relations: We have currently taken only a small step toward the processing of a message of several sentences as a whole unit, rather than a sequence of isolated pieces. At least within a message, resolution of pronouns and other anaphora could be performed to some extent using extant techniques. References to earlier messages, explicit and implicit, would require cooperation between the event record generation system and the content analysis system to which it interfaces.

At another level of analysis, work should be done to systematize the parts of the system that are specific to the sublanguage domain, so that the modification of the system to add new domains, with attendant vocabulary, syntax, and templates, can be done, perhaps with some guidance, by a subject expert. This type of analysis, called "knowledge acquisition" is a fairly new field of investigation, but is beginning to receive considerable attention.

6.0 References

Bartlett, F. C. [1932]
Remembering. Cambridge University Press, Cambridge, 1932.

Bobrow, D. G. and A. Collins, (eds.). [1975]
Representation and Understanding. Academic Press, New York, 1975.

Bobrow, D. G. and D. A. Norman [1975]
Some Principles of Memory Schemata. D. G. Bobrow and A. M. Collins (eds.), *Representation and Understanding*. New York: Academic Press, 1975.

Bobrow, D. G. and T. Winograd [1977]
An Overview of KRL, a Knowledge Representation Language. Cognitive Science, Vol. 1, No 1, January 1977, pp. 3-46.

Bonnet, A. [1979]
Understanding Medical Jargon as if it were a Natural Language. Proceedings IJCAI 6, Tokyo, 1979, pps. 79-81.

Brachman, R. J. and C. Smith (Eds.). [1980]
Special Issue on Knowledge Representation. SIGART Newsletter No. 70, Feb. 1980.

Bransford, J. D., and A. McCarrell [1974]
Sketch of a Cognitive Approach to Comprehension, W. Weimar and D. Palermo (eds), *Cognition and the Symbolic Processes*, Hillsdale, New Jersey: Lawrence Erlbaum Assoc., 1974.

Brown, J. S., Burton, R. R., and A. G. Bell [1975]
Sophie: A Step Toward Creating a Reactive Learning Environment. International Journal of Man-Machine Studies, September 1975, pp. 675-696.

Chafe, W. [1970]
Meaning and the Structure of Language. University of Chicago press, 1970.

Chafe, W. [1972]
Discourse Structure and Human Knowledge. In: R. Freedle and J. Carroll (Eds.). *Language Comprehension and the Acquisition of Knowledge*, Washington: V. H. Vinton, 1972, pp. 41-70.

Colmerauer, A. [1975]
Les Grammaires de Metamorphose. Marseille-Luminy: Groupe d'Intelligence Artificielle, 1975

Dwiggins, D. and G. Silva [1981]
Program Documentation for the MATRES III System Program Documentation submitted as part of RADC Contract F30602-80-C-0047, 1981.

Fillmore, C. [1968]
The Case for Case. In: E. Bach and R. Harms (eds.). *Universals in Linguistic Theory*. New York: Holt, Rinehart and Winston, 1968.

Fillmore, C. J. [1971]
Verbs of Judging: An Exercise in Semantic Description. Fillmore and Langendoen, *Studies in Linguistic Semantics.* New York: Hold, Rinehart and Winston, 1971.

Fillmore, C. J. [1975]
An Alternative to Checklist Theories of Meaning. Cathy Cogen, Henry Thompson, Graham Thurgood, Kenneth Whistler and James Wright, (eds.), *Proceedings of the First Annual Meeting of The Berkeley Linguistics Society.* University of California at Berkeley, 1975, pp. 123-131.

Goffman, Erving. [1974]
Frame Analysis, New York: Harper and Row, 1974.

Grishman, R. and L. Hirschman [1978]
Question Answering from Natural Language Medical Data Bases. Artificial Intelligence 11, 1978, pps. 25-43.

Hirschman, L., Grishman, R., and N. Sager [1976]
From Text to Structured Information: Automatic Processing of Medical Reports. Proceedings of the 1976 National Computer Conference, Montvale. NJ: AFIPS Press, 1976, pp. 267-275.

Kender, J.R. [1977]
An Annotated Bibliography of Natural Language and Speech Understanding Systems. Pittsburgh: Department of Computer Science, Carnegie-Mellon University, 1977.

Kowalski, R. [1979]
Logic for Problem Solving. New York: North Holland, 1979.

Kuhns J. L. [1974] *Synthesis of Inference Techniques: An Interpreted Syntax for the Logical Description of Events,* Operating Systems, Inc., N74-003 May 31, 1974.

Kuhns, J. L., Montgomery, C. A., and D. K. Whelchel [1975] *ERGO -- A System for Event Record Generation and Organization,* RADC-TR-75-61, Rome Air Development Center, Griffiss Air Force Base, New York, March 1975.

Miller, G. [1978]
Semantic Relations Among Words. M. Halle, J. Bresnan, and G. Miller, (Eds.), *Linguistic Theory and Psychological Reality.* Cambridge, MIT Press, 1978.

Miller, G., and P. Johnson-Laird [1976]
Language and Perception. Cambridge, Harvard University Press, 1976.

Montague, R. [1973]
The Proper Treatment of Quantification in Logic. Approaches to Natural Language, J. Hintikka, J. Moravcsik and P. Suppes, (eds.), Dordrecht, 1973.

Norman, D. A., and D. Rumelhart [1975]
Explorations in Cognition, W. H. Freeman and Co., San Francisco, 1975.

Pereira, F. [1980]
Extraposition Grammars. Proceedings of the Logic Programming Workshop, Debrecen, Hungary, 14-16 July 1980, pps. 231-242

Pereira, F., Pereira, L. and D. Warren [1978]
User's Guide to DECsystem-10 Prolog. Department of Artificial Intelligence, University of Edinburgh, 1978.

Pereira, F., and D. Warren [1980]
Definite Clause Grammars Compared with Augmented Transition Network Grammars. to appear in *Artificial Intelligence Journal*".

Rumelhart, D. and A. Ortony [1976]
The Representation of Knowledge in Memory. Center for Human Information Processing, CHIP 55, University of California, San Diego, 1976.

Sager, N. [1978]
Natural language information formatting, the automatic conversion of texts into a structured database. Advances in Computers, vol. 17, 1978.

Schank, R. (ed.). [1975]
Conceptual Information Processing. North-Holland Publishing Company, Amsterdam, 1975.

Silva, G., and C. A. Montgomery [1977]
Automated I&W File Generation, RADC-TR-77-194, Rome, New York, June 1977.

Silva, G., and C. A. Montgomery [1978]
Knowledge Representation for Automated Understanding of Natural Language Discourse. Computers and the Humanities, Vol 11, Pergamon Press, 1978.

Silva, G., Dwiggins, D. L., and J. L. Kuhns [1979a]
A Knowledge-Based Automated Message Understanding Methodology for an Advanced Indications System. RADC-TR-79-133, Rome Air Development Center, New York, June 1979.

Silva, G., Dwiggins, D. L., and C. A. Montgomery [1979b]
Satellite and Missile Data Generation for AIS, Final Technical Report, Operating Systems, R79-037, October, 1979.

Walker, D. E. (ed.) [1978]
Understanding Spoken Language. The Computer Science Library, Artificial Intelligence Series 5, New York: North-Holland, 1978.

Waltz, D. [1977]
Natural Language Interfaces. SIGART Newsletter no. 61, February 1977, pp. 16-64.

Warren, D., Pereira, L., and F. Pereira [1977]
Prolog — the Language and its Implementation Compared with Lisp. Proc of the ACM Symposium on AI and Programming Languages, SIGPLAN/SIGART Newsletter, Rochester, NY, August 1977.

Wilks, Yorick [1974]
Natural Language Understanding Within the AI Paradigm, A Survey and some Comparisons Stanford University, Stanford Artificial Intelligence Laboratory: Memo AIM-237, Stanford, CA. December 1974.

Woods, W. A. [1977]
Lunar Rocks in Natural English: Explorations in Natural Language Question

Answering. [Zampolli 1977], pp. 521-569.

Zampolli, A. [1977]
Linguistic Structures Processing. Fundamental Studies in Computer Science 5.
Amsterdam: North-Holland, 1977.

Appendix A — Sample Sentences from MATRES III Corpus

SKYLAB BROKE UP IN ORBIT.

SKYLAB HAS BROKEN UP IN ORBIT.

BREAKUP OF SKYLAB OCCURRED OVER THE INDIAN OCEAN
NEAR WESTERN AUSTRALIA.

BREAKUP OF SKYLAB TOOK PLACE OVER WESTERN AUSTRALIA.

BREAKUP OF SKYLAB WAS OVER WESTERN AUSTRALIA.

SKYLAB, THE ORBITAL WORKSHOP WHICH WAS LAUNCHED
FROM THE KENNEDY SPACE CENTER ON 14 MAY 1973,
HAS BROKEN UP IN ORBIT.

THE ORBITAL WORKSHOP, SKYLAB, WHICH WAS LAUNCHED
FROM THE KENNEDY SPACE CENTER ON 14 MAY 1973,
BROKE UP AT 0900Z ON 12 JUL 1979.

BREAKUP OF SKYLAB, THE ORBITAL WORKSHOP WHICH WAS
LAUNCHED FROM THE KENNEDY SPACE CENTER
ON 14 MAY 1973, TOOK PLACE AT 0900Z ON 12 JUL 1979.

SKYLAB, THE 81 DEGREE PROBABLE THIRD GENERATION SATELLITE,
HAS BEEN CONFIRMED IN ORBIT.

THE APPROXIMATELY 8 SATELLITES WHICH WERE LAUNCHED FROM CAIRO ON 21 MAY 1988
HAVE BEEN CONFIRMED IN ORBIT.

THE SATELLITE WAS CONFIRMED IN ORBIT ON REVOLUTION ONE
AT 0900Z ON 31 MAY 1978.

THE SATELLITE HAS BEEN CONFIRMED IN ORBIT ON REVOLUTION ONE
AT 0900Z ON 31 MAY 1971.

THE SKYLAB ORBITAL WORKSHOP, A CONVERTED S-48 THIRD STAGE FROM A
SATURN-5 LAUNCH VEHICLE,
WAS CONFIRMED IN ORBIT AT ABOUT 1700Z ON REVOLUTION ONE.

SKYLAB, THE 81 DEGREE PROBABLE THIRD GENERATION SATELLITE
LAUNCHED FROM THE KENNEDY SPACE CENTER AT 1900Z ON 6 MAR 1971
BY A SATURN-5 TYPE LAUNCHER,
HAS BEEN CONFIRMED IN ORBIT ON REVOLUTION ONE.

SKYLAB, LAUNCHED FROM THE KENNEDY SPACE CENTER
AT 1900Z TODAY, HAS BEEN CONFIRMED IN ORBIT
ON REVOLUTION ONE.

SKYLAB, THE 81 DEGREE PROBABLE THIRD GENERATION SATELLITE, LAUNCHED FROM THE KENNEDY SPACE CENTER EARLIER TODAY AT 1300Z, HAS BEEN CONFIRMED IN EARTH ORBIT ON REVOLUTION ONE.

THE SOVIET NEWS AGENCY TASS ANNOUNCED THAT SKYLAB DEORBITED INTO THE INDIAN OCEAN ON THE 34981ST REVOLUTION.

THE SOVIET NEWS AGENCY TASS ANNOUNCED THAT THE DEORBIT OF SKYLAB PROBABLY OCCURRED OVER CANADA EARLY ON REVOLUTION 34981.

THE DEORBIT OF SKYLAB WAS OVER CANADA ON 12 JULY 1979.

DEORBIT TOOK PLACE INTO THE AUSTRALIAN OUTBACK.

DEORBIT TOOK PLACE AT 1900Z ON 12 JULY 1979.

THE DEORBIT OF SKYLAB OVER CANADA TOOK PLACE ON 12 JULY 1979.

THE DEORBIT OF SKYLAB WAS DURING THE INITIAL PORTION OF REVOLUTION 34981.

THE SPACECRAFT WAS DEORBITED DURING ITS 125TH REVOLUTION AT 1330 HOURS THIS DATE.

THE DEORBIT OF SKYLAB OCCURRED DURING THE EARLY PORTION OF REVOLUTION 34981.

THE DEORBIT OF SKYLAB WAS PROBABLY OVER CANADA ON 12 JULY 1979.

THE SKYLAB ORBITAL WORKSHOP, A CONVERTED S-4B THIRD STAGE FROM A SATURN-5 LAUNCH VEHICLE, WAS DEORBITED INTO THE INDIAN OCEAN ON 12 JULY 1979.

THE SKYLAB ORBITAL WORKSHOP, A CONVERTED S-4B THIRD STAGE FROM A SATURN-5 LAUNCH VEHICLE, SUCCESSFULLY DEORBITED INTO THE AUSTRALIAN OUTBACK ON 12 JULY 1979.

THE SPACECRAFT DEORBITED ON THE SAME DAY.

THE SATELLITE WAS DEORBITED EARLY ON REVOLUTION 125.

THE SATELLITE WAS DEORBITED INTO THE INDIAN OCEAN.

NASA DEORBITED THE SATELLITE INTO THE INDIAN OCEAN ON THE EARLY PORTION OF REVOLUTION 145.

THE SATELLITE WAS DEORBITED INTO THE INDIAN OCEAN EARLY ON REVOLUTION ONE.

DEORBIT OF SKYLAB OCCURRED OVER CANADA EARLY ON REVOLUTION 34981.

DEORBIT TOOK PLACE EARLY ON REVOLUTION 34981.

NASA DEORBITED THE SATELLITE INTO THE INDIAN OCEAN ON REVOLUTION 123.

THE SATELLITE WAS DEORBITED BY NASA INTO THE INDIAN OCEAN ON REVOLUTION 123.

THE SATELLITE WAS EXPECTED TO DEORBIT INTO THE INDIAN OCEAN.

DEORBIT WAS EXPECTED TO OCCUR IN THE INDIAN OCEAN.

DEORBIT OF THE SATELLITE WAS EXPECTED TO OCCUR IN THE INDIAN OCEAN.

DEORBIT OF THE SATELLITE FAILED TO OCCUR IN THE INDIAN OCEAN.

THE SATELLITE FAILED TO DEORBIT INTO THE INDIAN OCEAN.

THE ICBM LAUNCHED FROM THE PLACENAME MISSILE TEST RANGE
AT APPROXIMATELY 1330 HOURS ON 13 MAY
FLEW TO THE NOMINAL 100 KM IMPACT AREA NEAR THE
PLACENAME MISSILE TEST CENTER.

AN ICBM WAS SUCCESSFULLY FLOWN TO THE PLACENAME AREA.
NASA ANNOUNCED THAT THE IMPACT OF SKYLAB TOOK PLACE IN AN AREA ABOUT 500 MILES FROM PERTH IN WESTERN AUSTRALIA.

THE SKYLAB ORBITAL WORKSHOP, A CONVERTED S-4B THIRD STAGE FPOM A SATURN-5 LAUNCH VEHICLE, IMPACTED IN WESTERN AUSTRALIA JUST NORTHEAST OF KALGOORLIE ON 12 JULY 1979.

THE VEHICLE IMPACTED NEAR KALGOORLIE IN THE AUSTRALIAN OUTBACK.

IMPACT OF SKYLAB OCCURRED IN WESTERN AUSTRALIA JUST NORTHEAST OF KALGOORLIE ON 12 JULY 1979.

IMPACT OCCURRED INTO THE NORMAL RECOVERY AREA.

IMPACT OCCURRED AT 1900Z ON 12 JULY 1979.

IMPACT IN WESTERN AUSTRALIA WAS NEAR KALGOORLIE.

THE IMPACT OF SKYLAB TOOK PLACE IN THE AUSTRALIAN OUTBACK.

IMPACT OF SKYLAB WAS ABOUT 500 MILES NORTHEAST OF PERTH IN WESTERN AUSTRALIA.

IMPACT WAS ABOUT 10 NM OUTSIDE OF THE NORMAL RECOVERY AREA.

THE MISSILE IMPACTED NEAR THE NORMAL RECOVERY AREA AT 1900Z ON 13 APR 1999.

IMPACT PROBABLY OCCURRED IN THE AUSTRALIAN OUTBACK JUST NORTHEAST OF THE CENTRAL PORTION OF THE RECOVERY AREA.

IMPACT OF SKYLAB IN WESTERN AUSTRALIA WAS ON 12 JULY 1979.

IMPACT IN AN AREA JUST NORTHEAST OF KALGOORLIE IN WESTERN AUSTRALIA WAS ON 12 JULY 1979.

IMPACT WAS PROBABLY IN AN AREA ABOUT 500 KMS FROM PERTH IN WESTERN AUSTRALIA ON 12 JULY 1979.

IMPACT PROBABLY OCCURRED IN AN AREA ABOUT 100 KMS FROM KALGOORLIE.

IMPACT OF SKYLAB WAS PROBABLY IN AN AREA ABOUT 500 KMS NORTHEAST OF PERTH.

IT IMPACTED IN WESTERN AUSTRALIA NEAR KALGOORLIE AT 1420Z.

A PROBABLE NAVAL ICBM WAS LAUNCHED FROM THE INDIAN OCEAN MISSILE COMPLEX TO THE INDIAN OCEAN EXTENDED RANGE IMPACT AREA AT 1900Z, ON 23 MAY 1971.

THE SKYLAB ORBITAL WORKSHOP, A CONVERTED S-4B THIRD STAGE FROM A SATURN-5 LAUNCH VEHICLE, WAS LAUNCHED FROM THE KENNEDY SPACE CENTER AT ABOUT 1330 HOURS ON 14 MAY 1973.

THE SKYLAB ORBITAL WORKSHOP, A CONVERTED S-4B THIRD STAGE FROM A SATURN-5 LAUNCH VEHICLE, WAS FIRED FROM THE KENNEDY SPACE CENTER AT APPROXIMATELY 1330 HOURS ON 14 MAY 1973.

AT APPROXIMATELY 1330 HOURS ON 14 MAY 1973, THE SKYLAB ORBITAL WORKSHOP WAS FIRED FROM THE KENNEDY SPACE CENTER BY A SATURN-5 TYPE LAUNCHER.

AT 1320Z, A DEFENSIVE MISSILE, POSSIBLY A SAM, WAS LAUNCHED FROM THE MISSILE TEST CENTER.

AN UNIDENTIFIED MISSILE, PERHAPS A DRONE, WAS FIRED FROM THE TEST CENTER AT 1320Z ON 15 MAY 1971.

THE SKYLAB ORBITAL WORKSHOP WAS LAUNCHED FROM THE KENNEDY SPACE CENTER AT APPROXIMATELY 1330 HOURS ON 14 MAY 1973 BY A SATURN-5 TYPE LAUNCHER.

THE SATELLITE WAS LAUNCHED ON A 67 DEGREE ORBITAL INCLINATION.

THE LAUNCH OF SKYLAB TOOK PLACE ON 14 MAY 1973.

THE SOVIET NEWS AGENCY, TASS, ANNOUNCED THAT THE LAUNCH OF SKYLAB TOOK PLACE ON 14 MAY 1973.

THE SOVIET NEWS AGENCY, TASS, ANNOUNCED THE LAUNCHING OF A MANNED SATELLITE.

IT WAS LAUNCHED AT 1320Z TODAY FROM THE KENNEDY SPACE CENTER.

SKYLAB REENTERED THE EARTH'S ATMOSPHERE OVER CANADA ON 12 JUL 1979.

REENTRY OCCURRED IN THE KALGOORLIE REGION AT ABOUT 1900 HOURS ON 12 JUL 1979.

REENTRY OF SKYLAB TOOK PLACE OVER CANADA ON 21 JUL 1979.

THE MISSILE REENTERED OVER WESTERN AUSTRALIA IN THE GENERAL VICINITY OF KALGOORLIE ABOUT 10 MINUTES AFTER LIFT-OFF.

THE SPACECRAFT REENTERED THE EARTH'S ATMOSPHERE AT 1330Z IN THE GENERAL VICINITY OF 9999N9999E.

AN UNIDENTIFIED MISSILE, LAUNCHED FROM THE PLACENAME TEST RANGE AT 1900 HOURS TODAY, REENTERED IN THE NORMAL RECOVERY AREA.

IMPACT OCCURRED AT THE SAME TIME.

IMPACT OCCURRED EARLIER TODAY.

IMPACT OCCURRED AT ABOUT 1230Z THIS DATE.

IMPACT OCCURRED AT ABOUT 1230Z ON 20 MAY 1971.

IMPACT OCCURRED AT ABOUT 1230 HOURS ON 20 MAY 1971.

IMPACT OCCURRED ABOUT 19 MINUTES LATER
AT A POINT NEAR 1530N2520E SOME 50 MILES DOWNRANGE.

IMPACT TOOK PLACE IN THE PLACENAME
ABOUT 10 MINUTES LATER.

IMPACT WAS ABOUT 50 NAUTICAL MILES OUTSIDE OF THE
NORTHWESTERN EDGE OF THE NORTHERN BROAD OCEAN AREA
CIRCULAR CLOSURE.

IMPACT OCCURRED IN CANADA IN THE GENERAL VICINITY OF
0000N0000W ABOUT 10 MINUTES AFTER LIFT-OFF.

THE MISSILE IMPACTED ON PLACENAME ABOUT 10 MINUTES AFTER LAUNCH.

THE MISSILE REENTERED IN THE NORMAL RECOVERY AREA
ABOUT 10 MINUTES AFTER LAUNCH AT 1330Z.

Appendix B — Sample Event Record Outputs from MATRES III

skylab broke up in orbit .

```
Event: breakup
  Object: satellite
    Equipment= skylab
    Quantity=
  Location= in orbit
  Time=
  Date=
```

skylab has broken up in orbit .

```
Event: breakup
  Object: satellite
    Equipment= skylab
    Quantity=
  Location= in orbit
  Time=
  Date=
```

breakup of skylab occurred over the indian ocean near western australia .

```
Event: breakup
  Object: satellite
    Equipment= skylab
    Quantity=
  Location= over the indian ocean near western australia
  Time=
  Date=
```

breakup of skylab took place over western australia .

```
Event: breakup
  Object: satellite
    Equipment= skylab
    Quantity=
  Location= over western australia
  Time=
  Date=
```

breakup of skylab was over western australia .

```
Event: breakup
   Object: satellite
      Equipment= skylab
      Quantity=
   Location= over western australia
   Time=
   Date=
```

skylab , the orbital workshop which was launched from the kennedy space center on 14 may 1973 , has broken up in orbit .

```
Event: breakup
   Object: satellite
      Equipment= skylab
      Quantity=
      Add_Attributes= the orbital workshop
         Rel_event=
            Event: launch
               Agent= agentunk
               Object: satellite
                  Equipment= orbital workshop
                  Quantity=
               Launchsite= from the kennedy space center
               Destination=
               Time=
               Date= on 14 may 1973
   Location= in orbit
   Time=
   Date=
```

the orbital workshop , skylab , which was launched from the kennedy space center on 14 may 1973 , broke up at 0900 on 12 jul 1979 .

```
Event: breakup
   Object: satellite
      Equipment= orbital workshop
      Quantity=
      Add_Attributes= skylab
         Rel_event=
            Event: launch
               Agent= agentunk
               Object: satellite
                  Equipment= skylab
                  Quantity=
               Launchsite= from the kennedy space center
               Destination=
               Time=
               Date= on 14 may 1973
```

```
  Location=
  Time= at 0900 hours
  Date= on 12 jul 1979
```

breakup of skylab , the orbital workshop which was launched from the kennedy space center on 14 may 1973 , took place at 0900 on 12 jul 1979 .

```
Event: breakup
  Object: satellite
    Equipment= skylab
    Quantity=
    Add_Attributes= the orbital workshop
      Rel_event=
        Event: launch
          Agent= agentunk
          Object: satellite
            Equipment= orbital workshop
            Quantity=
          Launchsite= from the kennedy space center
          Destination=
          Time=
          Date= on 14 may 1973
  Location=
  Time= at 0900 hours
  Date= on 12 jul 1979
```

skylab , the 81 degree probable third generation satellite , has been confirmed in orbit .

```
Event: confirm
  Object: satellite
    Equipment= skylab
    Quantity=
    Add_Attributes= the 81 degree probable third generation satellite
  Location= in orbit
  Revolution=
  Time=
  Date=
```

the approximately 8 satellites which were launched from cairo on 21 may 1988 have been confirmed in orbit .

```
Event: confirm
  Object: satellite
    Equipment= satellites
    Quantity= approximately 8
    Rel_event=
```

```
      Event: launch
        Agent= agentunk
        Object: satellite
          Equipment= satellites
          Quantity= approximately 8
        Launchsite= from cairo
        Destination=
        Time=
        Date= on 21 may 1988
  Location= in orbit
  Revolution=
  Time=
  Date=


the satellite was confirmed in orbit on revolution one at 0900 on 31 may 1978.

Event: confirm
  Object: satellite
    Equipment= satellite
    Quantity=
  Location= in orbit
  Revolution= on revolution one
  Time= at 0900 hours
  Date= on 31 may 1978


the satellite has been confirmed in orbit on revolution one at 0900 on 31 may
 1971 .

Event: confirm
  Object: satellite
    Equipment= satellite
    Quantity=
  Location= in orbit
  Revolution= on revolution one
  Time= at 0900 hours
  Date= on 31 may 1971


the skylab orbital workshop , a converted s-4b third stage from a saturn-5
launch vehicle , was confirmed in orbit at about 1700 on revolution one .

Event: confirm
  Object: satellite
    Equipment= skylab orbital workshop
    Quantity=
    Add_Attributes= a converted s-4b third stage from a saturn-5 launch
vehicle
  Location= in orbit
```

```
  Revolution= on revolution one
  Time= at about 1700 hours
  Date=
```

skylab , the 81 degree probable third generation satellite launched from the kennedy space center at 1900 on 6 mar 1971 by a saturn-5 type launcher , has been confirmed in orbit on revolution one .

```
Event: confirm
  Object: satellite
    Equipment= skylab
    Quantity=
    Add_Attributes= the 81 degree probable third generation satellite
    Rel_event=
      Event: launch
        Agent= agentunk
        Object: satellite
          Equipment= skylab
          Quantity=
        Launchsys= by a saturn-5 type launcher
        Launchsite= from the kennedy space center
        Destination=
        Time= at 1900 hours
        Date= on 6 mar 1971
  Location= in orbit
  Revolution= on revolution one
  Time=
  Date=
```

skylab , launched from the kennedy space center at 1900 today , has been confirmed in orbit on revolution one .

```
Event: confirm
  Object: satellite
    Equipment= skylab
    Quantity=
    Rel_event=
      Event: launch
        Agent= agentunk
        Object: satellite
          Equipment= skylab
          Quantity=
        Launchsite= from the kennedy space center
        Destination=
        Time= at 1900 hours
        Date= today
  Location= in orbit
  Revolution= on revolution one
```

```
  Time=
  Date=
```

skylab , the 81 degree probable third generation satellite , launched from the kennedy space center earlier today at 1300 , has been confirmed in earth orbit on revolution one .

```
Event: confirm
  Object: satellite
    Equipment= skylab
    Quantity=
    Add_Attributes= the 81 degree probable third generation satellite
      Rel_event=
        Event: launch
          Agent= agentunk
          Object: satellite
            Equipment= 81 degree probable third generation satellite
            Quantity=
          Launchsite= from the kennedy space center
          Destination=
          Time= earlier at 1300 hours
          Date= today
  Location= in earth orbit
  Revolution= on revolution one
  Time=
  Date=
```

the soviet news agency tass announced that skylab deorbited into the indian ocean on the 34981 revolution .

```
Infosource= the soviet news agency tass announce
Event: deorbit
  Object: satellite
    Equipment= skylab
    Quantity=
  Location= into the indian ocean
  Revolution= on the 34981 revolution
  Time=
  Date=
```

the soviet news agency tass announced that the deorbit of skylab probably occurred over canada early on revolution 34981 .

```
Infosource= the soviet news agency tass announce
Status= probably
Event: deorbit
  Object: satellite
```

```
    Equipment= skylab
    Quantity=
  Location= over canada
  Revolution= early on revolution 34981
  Time=
  Date=


the deorbit of skylab was over canada on 12 july 1979 .

Event: deorbit
  Object: satellite
    Equipment= skylab
    Quantity=
  Location= over canada
  Revolution=
  Time=
  Date= on 12 july 1979


deorbit took place into the australian outback .

Event: deorbit
  Location= into the australian outback
  Revolution=
  Time=
  Date=


deorbit took place at 1900 on 12 july 1979 .

Event: deorbit
  Location=
  Revolution=
  Time= at 1900 hours
  Date= on 12 july 1979


the deorbit of skylab over canada took place on 12 july 1979 .

Event: deorbit
  Object: satellite
    Equipment= skylab
    Quantity=
  Location= over canada
  Revolution=
  Time=
  Date= on 12 july 1979
```

the deorbit of skylab was during the initial portion of revolution 34981 .

Event: deorbit
  Object: satellite
    Equipment= skylab
    Quantity=
  Location=
  Revolution= during the initial portion of revolution 34981

the spacecraft was deorbited during its 125 revolution at 1330 hours this date .

Event: deorbit
  Object: satellite
    Equipment= spacecraft
    Quantity=
  Agent= agentunk
  Location=
  Revolution= during its 125 revolution
  Time= at 1330 hours
  Date= this date

the deorbit of skylab occurred during the early portion of revolution 34981 .

Event: deorbit
  Object: satellite
    Equipment= skylab
    Quantity=
  Location=
  Revolution= during the early portion of revolution 34981

the deorbit of skylab was probably over canada on 12 july 1979 .

Event: deorbit
  Object: satellite
    Equipment= skylab
    Quantity=
  Location= probably over canada
  Revolution=
  Time=
  Date= on 12 july 1979

the skylab orbital workshop , a converted s-4b third stage from a saturn-5 launch vehicle , was deorbited into the indian ocean on 12 july 1979 .

Event: deorbit

```
  Object: satellite
    Equipment= skylab orbital workshop
    Quantity=
    Add_Attributes= a converted s-4b third stage from a saturn-5 launch vehicle
  Agent= agentunk
  Location= into the indian ocean
  Revolution=
  Time=
  Date= on 12 july 1979
```

the skylab orbital workshop , a converted s-4b third stage from a saturn-5 launch vehicle , successfully deorbited into the australian outback on 12 july 1979

```
Status= successfully
Event: deorbit
  Object: satellite
    Equipment= skylab orbital workshop
    Quantity=
    Add_Attributes= a converted s-4b third stage from a saturn-5 launch vehicle
  Location= into the australian outback
  Revolution=
  Time=
  Date= on 12 july 1979
```

the spacecraft deorbited on the same day .

```
Event: deorbit
  Object: satellite
    Equipment= spacecraft
    Quantity=
  Location=
  Revolution=
  Time=
  Date= on the same day
```

the satellite was deorbited early on revolution 125 .

```
Event: deorbit
  Object: satellite
    Equipment= satellite
    Quantity=
  Agent= agentunk
  Location=
  Revolution= early on revolution 125
```

the satellite was deorbited into the indian ocean .

```
Event: deorbit
  Object: satellite
    Equipment= satellite
    Quantity=
  Agent= agentunk
  Location= into the indian ocean
  Revolution=
  Time=
  Date=
```

nasa deorbited the satellite into the indian ocean on the early portion of revolution 145

```
Event: deorbit
  Object: satellite
    Equipment= satellite
    Quantity=
  Agent= nasa
  Location= into the indian ocean
  Revolution= on the early portion of revolution 145
  Time=
  Date=
```

the satellite was deorbited into the indian ocean early on revolution one .

```
Event: deorbit
  Object: satellite
    Equipment= satellite
    Quantity=
  Agent= agentunk
  Location= into the indian ocean
  Revolution= early on revolution one
  Time=
  Date=
```

deorbit of skylab occurred over canada early on revolution 34981 .

```
Event: deorbit
  Object: satellite
    Equipment= skylab
    Quantity=
  Location= over canada
  Revolution= early on revolution 34981
  Time=
  Date=
```

deorbit took place early on revolution 34981 .

```
Event: deorbit
  Location=
  Revolution= early on revolution 34981
```

nasa deorbited the satellite into the indian ocean on revolution 123 .

```
Event: deorbit
  Object: satellite
    Equipment= satellite
    Quantity=
  Agent= nasa
  Location= into the indian ocean
  Revolution= on revolution 123
  Time=
  Date=
```

the satellite was deorbited by nasa into the indian ocean on revolution 123 .

```
Event: deorbit
  Object: satellite
    Equipment= satellite
    Quantity=
  Agent= nasa
```

the satellite was expected to deorbit into the indian ocean .

```
Status= expect
Event: deorbit
  Object: satellite
    Equipment= satellite
    Quantity=
  Location= into the indian ocean
  Revolution=
  Time=
  Date=
```

deorbit was expected to occur in the indian ocean .

```
Status= expect
Event: deorbit
  Location= in the indian ocean
  Revolution=
  Time=
  Date=
```

deorbit of the satellite was expected to occur in the indian ocean .

```
Status= expect
Event: deorbit
  Object: satellite
    Equipment= satellite
    Quantity=
  Location= in the indian ocean
  Revolution=
  Time=
  Date=
```

deorbit of the satellite failed to occur in the indian ocean .

```
Status= fail
Event: deorbit
  Object. satellite
    Equipment= satellite
    Quantity=
  Location= in the indian ocean
  Revolution=
  Time=
  Date=
```

the satellite failed to deorbit into the indian ocean .

```
Status= fail
Event. deorbit
  Object: satellite
    Equipment= satellite
    Quantity=
  Location= into the indian ocean
  Revolution=
  Time=
  Date=
```

MISSION

of

Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.