AD-A038 968 NAVAL ELECTRONICS LAB CENTER SAN DIEGO CALIF F/G 17/2
NAVY COMMAND CONTROL AND COMMUNICATIONS SYSTEM DESIGN PRINCIPLE--ETC(U)
AUG 76

UNCLASSIFIED NELC/TD-504-VOL-5 NL

1 OF 2
AD A038968

CONT.
END
DATE FILMED
5-77

NELC / TD 504

NELC / TD 504

Technical Document 504

AD A038968

# NAVY COMMAND CONTROL AND COMMUNICATIONS SYSTEM DESIGN PRINCIPLES AND CONCEPTS

## Volume V: Appendix D – Automated Orderwire Concepts for Navy $C^3$ Network

DDC
RECEIVED
MAY 3 1977
B

15 August 1976

Naval Warfare Effectiveness Group (Code 233)

Prepared for
NAVAL ELECTRONIC SYSTEMS COMMAND (PME 108)

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED

NAVAL ELECTRONICS LABORATORY CENTER
San Diego, California 92152

AD No.
DDC FILE COPY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|
| 1. REPORT NUMBER: NELC Technical Document 504 (TD 504) | 2. GOVT ACCESSION NO. |
| | 3. RECIPIENT'S CATALOG NUMBER: NELC/TD-504-Vol-5 |
| 4. TITLE (and Subtitle): NAVY COMMAND CONTROL AND COMMUNICATIONS SYSTEM DESIGN PRINCIPLES AND CONCEPTS, Volume V. Appendix D, Automated Orderwire Concepts for Navy $C^3$ Network | 5. TYPE OF REPORT & PERIOD COVERED |
| | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Electronics Laboratory Center, San Diego, California 92152 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62721N, F21241, SF21241402 (NELC Q239) |
| 11. CONTROLLING OFFICE NAME AND ADDRESS: NAVELEX (PME 108) | 12. REPORT DATE: 15 August 1976 |
| | 13. NUMBER OF PAGES: 94 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | 15. SECURITY CLASS. (of this report): UNCLASSIFIED |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited

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

18. SUPPLEMENTARY NOTES

Vol 1 — AD-B017 851 NTIS
2 AD-A038 901 NTIS
3
4

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

Command control
Navy $C^3$
Telecommunications

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

This is volume V of an eight-volume document on Navy $C^3$ concepts. Volume V contains appendix D and discusses concepts and issues related to automatic orderwire for $C^3$ systems.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102 LF 014 6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403 940

NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A


| Volume | | |
|---|---|---|
| I | | $C^3$ Principles and Concepts |
| | | Appendices |
| II | A. | GLOSSARY — A038 901 |
| III | B. | EVALUATING AND SELECTING THE MIX OF TRANSMISSION MEDIA FOR THE $NC^3N$–A METHODOLOGY |
| IV | C. | $NC^3N$ NODES – FUNCTIONS, SUBSYSTEMS, AND ARCHITECTURAL DESIGN CONCEPTS |
| V | D. | AUTOMATED ORDERWIRE CONCEPTS FOR $NC^3N$ A038 968 |
| VI | E. | NETWORKING PRINCIPLES AND FEATURES |
| | F. | DATA BASE MANAGEMENT CONSIDERATIONS |
| | G. | $NC^3N$ USER DATA AND INFORMATION EXCHANGE NETWORK REQUIREMENTS |
| VII (Secret) | H. | NUCLEAR ENVIRONMENT CONSTRAINTS (U) |
| | I. | THREAT ENVIRONMENT FOR $NC^3N$ (U) |
| | J. | $NC^3N$ VULNERABILITY TO JAMMING, DECEPTION, INTERCEPT (U) |
| | | ANNEX A OF APPENDIX B |
| | | ANNEX E OF APPENDIX E |
| | | ANNEXES C AND D OF APPENDIX G |
| VIII | K. | ARCHITECTURAL ALTERNATIVES — B047 851L |
| | L. | DESCRIPTIONS OF R&D INITIATIVES |


This is volume V of an eight-volume document on Navy $C^3$ concepts. Volume V contains appendix D and discusses concepts and issues related to automatic orderwire for $C^3$ systems.

CONTENTS

1.0 INTRODUCTION . . . page 1

2.0 CONCEPTS . . . 3

2.1 Body of Knowledge Concept . . . 3

2.2 Orderwire Language Concepts . . . 5

3.0 GENERAL PROPERTIES . . . 6

3.1 Properties of the Orderwire Network . . . 6

3.2 Properties of the Knowledge Base . . . 38

3.3 Properties of the Orderwire Language . . . 51

4.0 IMPLEMENTATION CONSIDERATIONS . . . 71

4.1 Condensation of Orderwire Information . . . 71

4.2 Equipments . . . 78

5.0 APPLICABLE TECHNOLOGIES . . . 82

5.1 Introduction . . . 82

5.2 Media Transmission Technologies . . . 82

5.3 Automated Narrowband HF, Satellite, UHF, Optical, Packet Radio, etc. . . . 82

5.4 Digital System Technology . . . 83

5.5 Modulation and EDAC Technology . . . 83

5.6 Formal Language Technology . . . 83

5.7 Knowledge Representation Technology . . . 83

5.8 Communication Protocol Technology . . . 83

5.9 Network Management Technology . . . 83

5.10 Machine Heuristic Function "Experts" . . . 83

5.11 Resource Management Technology . . . 84

5.12 Crisis Management Technology . . . 84

5.13 Man Machine Interaction Technology . . . 84

5.14 Tech Control Technology . . . 84

5.15 Survivability Analysis Technology . . . 84

5.16 Command Control Information Exchange Requirements Technology . . . 84

5.17 Data Communication Security Technology . . . 85

5.18 Communication Routing Technology . . . 85

5.19 Circuit Switching and Store and Forward Switching Technology . . . 85

CONTENTS (Continued)

5.20 System Building Technology . . . 85

5.21 Simulation Modeling Technology . . . 85

5.22 Implementation Planning Technology . . . 85

6.0 RECOMMENDED STUDIES . . . 85

6.1 Expanding Study . . . 85

6.2 Construction of Simulation Model . . . 86

6.3 Detailing of Primitive and Normal Languages . . . 86

6.4 Determination of a Family of Routing Disciplines . . . 86

6.5 Development of Centralization vs. Distribution Philosophy . . . 86

6.6 Developing Appropriate Semantic Knowledge Systems . . . 86

6.7 Integrating 6.2, 6.3, 6.4, and 6.5 Systems into the Simulation Model of 6.1 . . . 87

6.8 Developing Standards for Equipment Procurement . . . 87

6.9 Development of Special Equipment . . . 87

6.10 Development of Protocol . . . 87

6.11 Developing Control Operator Control Language and Protocol . . . 87

6.12 Developing System Measurement and Performance of Criterion . . . 88

6.13 Developing Utilization Value System . . . 88

6.14 Developing Command and Control "Expert" . . . 88

6.15 Development for Media Availability "Expert" . . . 88

6.16 Detailing Elemental Actions . . . 88

6.17 Detailing Equipment Scheduling Expert . . . 88

6.18 Developing Channel Evaluation Expert . . . 89

6.19 Developing Implementation Plans . . . 89

6.20 Determining Costs . . . 89

## 1.0 INTRODUCTION

Our modern Navy uses computer technology in virtually every phase of operations. From the requisitioning of a part for a diesel engine to the checkout of an aircraft's navigation electronics, automation performs fast and accurately and relieves operators for duties which urgently require their attention. But procedures to change or modify communications using the orderwire are neither modern nor effective.

In response to adverse atmospheric conditions, failure of communication security, and enemy communications countermeasures, operators must manually reroute communication channels. Normally, manual rerouting is merely slower and less accurate than automatic rerouting, but in a combat situation slower and less accurate can be catastrophic.

The automated system not only averts catastrophic nonreadiness through faster reaction, but provides additional capabilities as well. One of these is the capability of responding to ionospheric changes, whether natural or artificial. Sensors checking for changing propagation conditions allow the automated orderwire to change frequencies or routing to best adapt to the situation. More important, a secure automated orderwire is an improvement over an orderwire user employing noncovered channels. If the orderwire is operated in the uncovered mode, a clear and present danger exists from enemy passive countermeasures. Long-term enemy monitoring of the orderwire could compromise major portions of the system by making them vulnerable to active jamming at a most critical time. The automated system denies the enemy the uncovered fixed channel by providing orderwire information on all channels and in a covered fashion. In addition to this improvement provided by the automated orderwire on a day-to-day basis, there is a benefit derived in times of crisis. These periods of increased communications traffic perhaps arising from actual or imminent attack place great demands on communication personnel and facilities. Personnel in the automated orderwire system are free to carry out higher-priority tasks instead of being tied to the manual rerouting of channels. These demands on communication personnel are presently heavy; in the future they will be even more severe.

Projections of communications requirements indicate that the Navy must become more frugal with channel capacity. With international agreements restricting expansion in the frequency spectrum, no choice is left but to make better use of channels. Current practice is to select for the orderwire a channel having low traffic. Several networks now exist that have low traffic because of requirements for noninterference. Since these special channels are used only intermittently, the monopoly of this channel capacity would be a luxury that the Navy could ill afford.

Other developments in communications will make further demands on the orderwire. For example, although the development of the laser and optical line-of-sight communications will provide greater channel capacity, servicing the additional channels will increase the demands on the orderwire. The problem is aggravated by the possibilities of communicating optically beyond the line of sight.

The advent of miniaturized electronics has changed cost and space limitations for data processing and memories so that radical departures from conventional design are now possible for the orderwire. Microprocessors together with compact high capacity memories can process orderwire information at the communication equipment, giving distributed control with very high system survivability.

Along with hardware developments have come gains in system control technology, and this report outlines the development of the concept of an automated orderwire, including language, knowledge base, desirable management, system properties, and implementation considerations.

The general structure of naval communications and some current areas of concern can best be presented through an illustrative scenario. Picture a fleet of vessels steaming in formation. Because of the destructive power of today's weapon systems, these formations are no longer the tight formations of the prenuclear era. Rather, they are spread out over a great many square miles of ocean. There is a real need for ships on opposite sides of a very large formation to be able to communicate extensively and very, very quickly. For example, if the formation suddenly came under attack, every ship in the formation would need the complete tactical picture as it developed; Are there incoming missiles? Are there enemy submarines? What are the threats and where are they coming from? It is exactly this environment for which naval communications must be designed.

In its normal operating state, the communication system attempts to maintain readiness in anticipation of attack. Let us add a few more factors to our scenario to better understand this situation. First, while we have the technical capability to communicate between ships that are a hundred miles apart,* we must assume that the enemy has the technical capability to intercept and monitor or jam these communications. Second, the greater the distance across which we must communicate, the greater the power that is required to accomplish the communication, and the higher the probability of enemy interception. Third, it can also be assumed that the enemy does not have technical capability to successfully monitor the entire electromagnetic spectrum in all combinations and patterns of frequency. Lastly, atmospheric anomalies are constantly occurring, some with a life span of only a very few minutes, which could be advantageous to our communications.

Returning to our fleet, we would like a communication system that accommodates all the above factors. Consider two people sitting silently in the same room. The capability to communicate is present even though it is not being exercised. All that is necessary is for one of them to begin speaking. This is analogous to the anticipatory period for our fleet. However, unlike the two people in the room, the fleet's mere presence does not automatically create the capability to communicate. With the fleet, it is much more difficult. The sender and receiver have to agree as to when and how the communication will occur, which equipments will be involved, what frequencies will be used, etc. Because of the factors mentioned above, a tactic is to use intervening ships as relay stations between the outermost ships to minimize power requirements; and to change frequencies to make enemy interception more difficult. The idea is to present a small, rapidly moving communications target to the enemy. However, this places a tremendous burden on the communications operators who must manually establish and disestablish the communications links.

The future of global naval communications lies in a communication system which is secure and self-organizing, has automatic rerouting, and has a high degree of survivability. The system must be transparent to the user. That is, the user should be concerned only with functional operation and have little or no involvement with the technical electronic details. The communication system should present the user with the capability to accomplish his primary goal; namely, to transmit a message between two or more points or nodes with high probability of successful transmission. Further, the future naval communication system should be highly automated and able to transmit information via various media at different data rates, automatically rerouting traffic, if required.

The underlying concept of such a global communication system architecture is the orderwire. The orderwire is defined as a body of information or a set of specifications required to

---

* 1 statute mile = approx 1.6 km

establish new communication channels or routings or modes of operation, or assignments within existing channels.* Communications in the context of this document have a twofold meaning – data and orderwire. This report is concerned with orderwire. There must be at least two types of orderwire – one to service the user and the other for use by the system to update itself.

The user requirement would generally consist of the transfer of a set of specifications to set up specific communication equipment parameters and define the selected transmission medium. The user must not be required to wait for the link to be established. It should always appear to be available for his use.

The tactical requirement calls for the communications system to have a fluid structure, with links continually being established and disestablished to take optimum advantage of existing atmospheric conditions and command structure and to confuse the enemy. This, coupled with the user requirement, dictates a system requirement to be able to very rapidly establish a communication link in anticipation of use. Therefore, the orderwire must have the capability to automatically establish multiple communication links and to determine compatibility prior to transmission. This establishes the need for the second type of orderwire, that which the system uses to update itself.

Orderwire dialog implies the need for the transmission of data and acknowledgement of receipt of those data. The type of data to be transferred can range from specific equipment characteristics, tables, procedures, and dictionaries to functional network graphs. Finally, the underlying concept of the orderwire is the automatic building of structures in communications networks from stored knowledge under the guidance of a centralized control system.

## 2.0 CONCEPTS

### 2.1 BODY OF KNOWLEDGE CONCEPT

#### 2.1.1 DESCRIPTION

Let us begin our description of orderwire data with a steady-state channel conveying no information. A single pulse – a bit – arrives at a specific time. This is the shortest possible orderwire. We now have a single-entry orderwire sentence comprising one pair – bit and time of transmission. The bit may be either a one or a zero, and there must be agreement between the originator and the destination of that bit of information as to the action desired in either case. In very simple systems, such as turning a light on and off, the switching can represent the action to be taken. But, in general, when dealing with a system as complex as orderwire for Navy communications, the bit of information must be supported by a body of knowledge, a certain part of which is found at both the generating node and the destination node. For example, the bit may represent turning a switch on or off, but the body of knowledge must describe which switch is to be turned on or off. If there is more than one switch in the node, it will take more than one bit of information to select the appropriate switch. An understanding between the two nodes must exist as to which switch is to be turned on or off. That agreement must either have been established by a previous orderwire sentence or have been the original system convention. Further, the time when the pulse should arrive must also have been previously established by orderwire sentences. These initial states or agreements must be capable of being modified by subsequent orderwire commands.

*This definition is different from that given by Military Standard 188C, which defines orderwire as a circuit.

Since any orderwire is composed of collections of single-bit orderwires, a dependency exists between previous orderwire sentences and the current one. Each node has control of its own equipment. It also has knowledge regarding its equipment specifications, capabilities, and the specific settings for all its own equipment to accomplish communications. Each node may require similar information for some or all other nodes in the system.

### 2.1.2 STRUCTURE

Knowledge as communicated in Navy systems is generally structured as bits bound into characters, words, phrases, clauses, statements, paragraphs, etc. The minimum amount of knowledge that can be conveyed is a statement.[1]

A string of characters has no value unless there is an implied attachment between the string of characters called words and an existential or intentional existence to which the words refer.[2]

The statements in turn must be linked to the generation of connections between the words. There is an immediate connection in that the words are all grouped together in the statement, but this rather trivial connection is not adequate. The operations of reasoning and command cannot be applied to this sequence of connected words. A set of conventions must exist that translates those statements into a knowledge representation that has aspects and rules whereby the operations of reasoning and command can be applied.

The most appropriate representation of knowledge is an ordered graph each node and arc of which is namable and some arcs of which are related to objects outside the graph.* The system will interpret this graph to generate activity associates with the names (eg, turn on modem type K number 5 power).

---

[1] Sometimes a part has an implied meaning of existence. In these cases an implied statement exists.

[2] For example, "ships" is an intentional reference where ships stands for the collection of objects in a frame of knowledge such as USS NIMITZ and USS OKLAHOMA CITY. The USS NIMITZ on the other hand lies outside this frame of reference. In another knowledge frame of reference the USS NIMITZ is an intentional object and propulsion system 567S4, navigation system 47859, etc, are the existential reference.

*Generally this connection will be made only through names; ie, the graph will not have pointers to these exterior objects, but will have pointers to names.

## 2.2 ORDERWIRE LANGUAGE CONCEPTS

### 2.2.1 ORDERWIRE LANGUAGE REQUIREMENTS

The orderwire was defined to be a body of information or a set of specifications (extracted from the body of knowledge described earlier) required to establish new communications channels and routing. The body of knowledge, however, is not static. Changing conditions necessitate enlarging or extending the body of knowledge. To enlarge the existing body of knowledge can be thought of as updating a file of specifications. The extension of knowledge may be thought of as enlargement of the total communication system; ie, increasing the number of nodes in the system. This situation can occur very frequently. For example, a new ship can join an existing task force (a permanent node) or an aircraft can come under temporary command of a ground controller.

The enlargement or extension of a body of knowledge presupposes an existing body of knowledge with a language capable of describing it.

### 2.2.2 ORDERWIRE LANGUAGE CHARACTERISTICS

The orderwire language requirements do not require a continually changing language. However, they do require that changes in the language occur to address particular channel requirements. Provisions for these changes must be well formulated in the design of the system. They are finite in number and generated by the human designers and maintainers. For this reason, it is conjectured that the orderwire requirements will need totally nested sets of languages. That is, from any form of the language used, a new language can be formed that is derived from the old language and provides a more condensed notation.

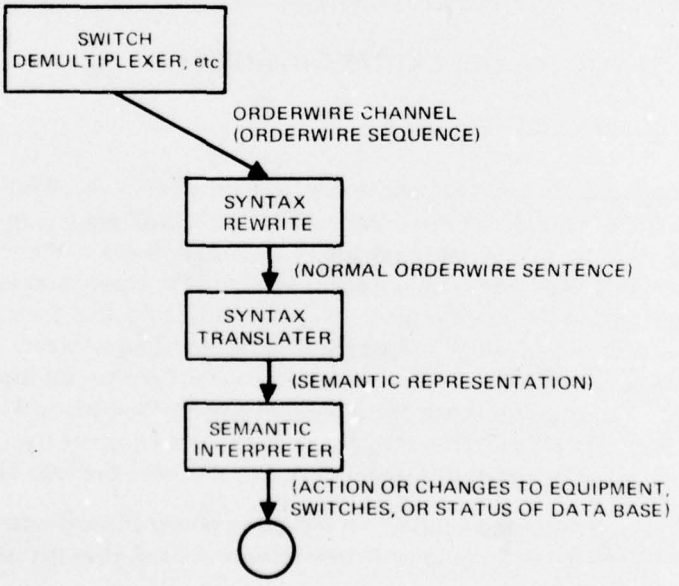



Figure 2-1. Orderwire sequence flow.

There must be a language attached to any active orderwire that permits the orderwire to be interpreted. From the design point of view, it is desired that the original interpretation of the orderwire will be universal for all orderwire channels and that the basic orderwire interpretation will be used when a primitive expanded description is desired. When an orderwire sentence conforms to this primitive interpretation, we will say that the orderwire is written in the primitive orderwire language.

The primitive orderwire language must have several essential characteristics. It must be able to describe and name actions or parts of actions. It must be amenable to partitioning: ie, many short sentences lead to one large action. This latter requirement arises because short sentences can be received better than long sentences. In this context, ordering of the short sentences should not be of great importance.

The normal orderwire language is one that has been constructed from the primitive orderwire language but still contains the syntax for the primitive orderwire language. Hence, a statement written in the primitive orderwire language will still be interpreted correctly. For the most part, the normal orderwire language will be tailored to the particular orderwire channel and may or may not share parts with other orderwire channels. The primitive orderwire language is common to all normal orderwire channels.

A normal language may be modified to create a special-purpose language. When a special language has been created, a mechanism must be built into the special language application to permit a return to the normal language. This is known as a language escape. The escape mechanism allows for the return of the previously used language, requiring that a memory of the past normal language be maintained. When the escape mechanism is used a number of times, it will return the orderwire language to the primitive language.

## 3.0 GENERAL PROPERTIES

Having reviewed the general concept of an automated orderwire for network control, let us now undertake a more detailed examination of its general properties.

### 3.1 PROPERTIES OF THE ORDERWIRE NETWORK*

#### 3.1.1 EQUIPMENT CONTROL

Each node has its own equipment complement to manage. While it is expected that a standardized suite of equipment will eventually be used, each node may possess some functionally unique abilities which imply the use of different classes of equipment. Besides the logistics aspects, this gives rise to the problem of where the equipment knowledge should reside. As evidenced in the preceding section, it is mandatory that the system control have available a tremendous amount of information about nodal equipment. The suite of equipments presently involved consists of antenna controllers, cryptos, multiplexers, modems, computers, etc. The computers and the attendant storage facilities will contain data bases and procedural specifications. However, for each of these equipments which exercises a function which has to be controlled, the orderwire must have the requisite capability so that:

a. Every state of the equipment, including power turnoff, can be immediately selected for that equipment by a remote control through one or more ports.

*See also **SIGNET**, section 2.0, appendix E.

b. The state of the equipment can be ascertained by interrogation without modifying the functioning states of the equipment.

c. Each change of state of the equipment shall be available for monitoring if desired. An interrupt shall be activated by any change of the equipment state.

d. Certain important state changes shall cause a special* interrupt to be activated.

e. Any interrupt for the control functions can be inhibited at the option of the system operator or control executive.

f. Each interrupt can be reset individually or in groups.

g. Two separate control line ports should be available in which the latest control from either port will always be honored.

h. Controls from either of the control ports can turn off the other. The state in which all control ports are turned off is prohibited.

i. Every significant state of the system should be controllable by manual operation of the equipment switches.

j. All controls including manual are effective at initial power up.

k. The remote control can inhibit or reinstate manual controls except for the power-off switch.

l. All manipulation of manual controls causes capture of the manual switch settings and an interrupt to be activated. This should be true whether manual controls are inhibited or not.

m. All instruments on the equipment shall show the true state of the equipment.

n. Each equipment shall have a number of status indicators that reflect the control status in addition to the equipment status indicators. Seven alphanumeric digits would probably be adequate.

o. In addition to the equipment control switches, each equipment shall also have a set of manual switches that do not control the function of the equipment, but are capable of being read remotely. This is for the purpose of allowing a dialog between a central controller and the local station concerning system control actions.

p. Except for the initial power-on switch and the set of switches described in item o, the manual controls need not be physically mounted on the equipment. If they are not mounted on the equipment, they should be capable of displaying the settings of the switches described in item o and be able to override these settings.

q. Any sequencing that is normally a function of equipment self-control can be inhibited or stepped at any step by remote control. (Exceptions are made with regard to cascade steps that have no internally well defined states.) This is to allow the stepping through of a system to determine system flow and faults.

---

*An equipment may have many substeps which are significant if the equipment is being checked. These changes are trapped by the requirement expressed in paragraph c. But when the equipment is in actual use, it is not desirable that each substep cause an interrupt to tell the control that the state has changed. However, a number of steps are significant in its operation and these changes should be monitorable by the control. A second interrupt type is required for these changes of state. This requirement is specified in paragraph d. Examples of this type of change are: the equipment was turned on; the equipment is ready to receive control; the equipment has received a frequency change command; the equipment has made the frequency change; the equipment has received a transmit command; the equipment has stored transmitting; etc.

The foregoing list provides a good indication of the level of automation which will be required at the equipments themselves. While the list appears complex, there are only three types of elemental actions involved in equipment control.

The first elemental action is that action which will cause power to be applied to the equipment in such a manner that it becomes, in terms of the network, an operating unit. There are actually two parts to the power-on phase, only one of which is considered an elemental action. The two parts may be considered to be "master on" and "controlled on." Only the latter is an elemental action in the automated orderwire system. The "master on" action is a manual operation which applies power to the system and sets certain condition lines so that the automated system can remotely control a "turn on" which allows for the system to be powered up to the point at which it is useful.

The reverse action would be slightly different. The automated system would be capable of remotely effecting either a "control off" or a "master off." It should be pointed out that this and the following elemental actions imply that the system must have the capability of routing the command from the control element involved in the orderwire to the specific equipment in order to send it a signal that turns the equipment on or off. The following actions must also have this routing capability.

### 3.1.2 LINK GENERATION

The establishment of a link between two nodes may be initiated either by one of the two nodes involved or by a central node that has been assigned network responsibilities. Without attention to how it was determined that a link should be established, the mechanics involved in its establishment will be examined.

First, the two nodes have to acknowledge each other and agree to establish a link. Assume that a hf radio link is to be established. After the agreement to establish a link has been completed, transmitting node must ask for certain information. What is the status of reception conditions? Which frequencies are operative? Which are best to use? The receiving node will respond. A not unusual occurence is that the receiving node is having frequency selection problems and as a result is limited in selections. The receiving node indicates which frequencies will be monitored. The two nodes get the proper receivers, transmitters, multicouplers etc, interconnected. Some testing of the links is necessary. The transmitting node then sends a signal. The receiving node analyzes the signal to determine whether the level of quality is satisfactory for the type of transmission to be performed. The two nodes may send an innocuous signal designed for error analysis. The two nodes have to communicate with each other and acknowledge receipt of the communication about the setup and testing of a number of parameters, such as the kind of error detection and correction to be employed, the keysetting to be used, and at what point in time the first bit of a time lock synchronization code is to arrive. After all this has been accomplished and tested and the fact has been verified that a working link exists, the working link can be made available to the user.

All the above must occur for every link established in a manual system. All the above must also occur for every link established in an automated system. The automation of a communication network does not eliminate the need for exchange of knowledge – it simply changes the residence of that knowledge and the manner of access. The automated system, having made a determination (perhaps by a stimulus injected by a communications operator that a given link should be established), must perform all the same interrogations, testing, and "handshaking" that occur in the manual system. This forces a certain kind of structure

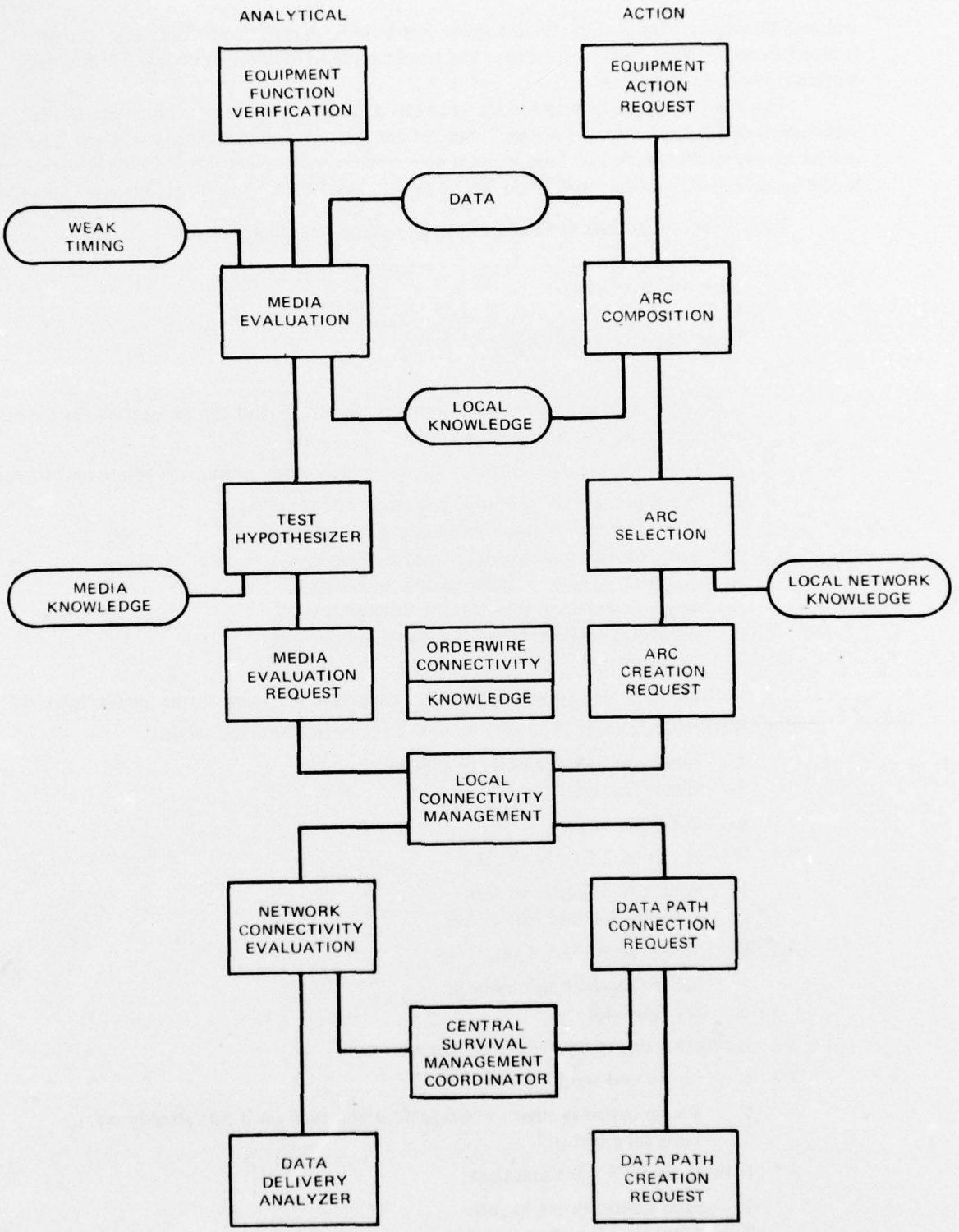



Figure 3-1. Link generation procedure.

and intelligence to that part of the automated orderwire that performs network control. It also forces a certain kind of structure and intelligence to become embedded in the network hardware equipments.

The analysis of arc generation on hf below gives an example of a production rule description of the activities of creating a new hf arc for the communication system. The rules in each top-level section are an independent rule system with activations explicitly mentioned. In the next level all rules are active simultaneously. The "then" points are executed in order.

1.0 A request for arc generation arrives with parameters:

Source node
Sink or destination
HF
What capacity or reliability

2.0 Arc generation process for hf

2.1 If current node does not equal source node, then send the request to source node and terminate this agent at this node.

2.2 If current node equals source node, clear frequency register, if filled, and then:

1. Send request to sink node for cooperative dialog
2. Send an item of clear frequency to sink and mark
3. Request transmission test from sink node
4. Request ACK on test frequency application
5. Request NAK on rejection of frequency
6. Activate 3.0 (Establish Exchange), cancel 2.0
7. Set time out

2.3 If clear frequency register has been exhausted, then consult hf media "expert."*

2.4 If hf media expert produces no new frequencies on a call, then:

1. Notify requesting node of failure
2. Cancel request

3.0 Establish Exchange

3.1 If time out of 2.2 of NAK, then:

1. Send cancel order to sink
2. Activate 2.0 and cancel 3.0

3.2 If ACK and receiver not on, then:

1. Set up receiver and turn on
2. Set time out

3.3 If receiver on, then set up test and turn on.

3.4 If test is on and signal present, then:

1. Set up capacity measure subsystem and turn on if not already on.
2. Turn time out off

3.5 If time out of 3.2 occurs, then:

1. Send cancel order to sink
2. Activate 2.0 and cancel 3.0

---

*Expert as used here refers to an automatic callable program.

3.6 If capacity results are determined, then activate 4.0 and cancel 3.0.

4.0 Set up arc.

4.1 If capacity < request capacity, then activate 2.0 and 5.0 (utilization evaluation) and cancel 4.0

4.2 If capacity > requested capacity, then call Modem EDAC expert if not already called.

4.3 If Modem EDAC expert has determined a scheme, then:

1. Send that scheme to transmitter
2. Set up EDAC on arc
3. Activate 7.0 and cancel 4.0

5.0 Utilization and memory

This adds the knowledge to the connectivity knowledge module that a potential arc exists at this frequency, and sends similar information to transmitting node. If equipment and request evaluation determines utility, then a connecting arc may be set up on this arc connector even though it does not meet the present requirements for which an arc was generated.

6.0 Complete connection of arc

7.0 Cover Link

7.1 If Modem EDAC is operating and no key, then activate AKDS expert.

7.2 If key is available and not applied, then:

1. Time and insert key
2. Send command to time and insert key at transmitter

7.3 If arc covered, then:

1. Activate arc utilization system with request compliance
2. Kill 7.0

### 3.1.3 NETWORK CONTROL

Multinode networks are generated in accordance with a set of specifications. The impetus behind network growth can be twofold. First, it can be generated by user demand which can be described in terms of transmission media requirement, time and distance requirements, and traffic loads. Second, growth can be spurred on by the need to establish goal tests and evaluation procedures.

Network control consists of those activities of the system that handle the development of a user network, be it a simple network, such as one node to another, or a multiuser network with many users. In either case, there are a number of activities that must be accomplished in order for an automated network control to be realizable.

a. Building User Networks. The network is composed of a number of channels in a system. Each channel must have a path in which to lie. The first activity of network control is to build the paths into which the channel can be placed. The building of paths is essentially the building of a sequence of arcs or links in order to establish an activity between two points. Once the need for a network is established, it is generally considered to be a network

for a set of users. That set of users, the protocol, and the structure of the communications between the users are considered to be the definition of the network. Once that definition is known, the system can then configure itself in order to build a structure that is compatible with that definition of the network. This is accomplished essentially by first building the paths between the user nodes of the system and then building the necessary protocol that allows the user to speak in the manner that is desired by the user network.

b. Building Protocol for User Networks. Protocol is the set of rules and regulations that governs the actions of the various pieces of the network with respect to each other and the data system. For example, it may determine which users are interrogated at which point in time and what result that interrogation should have as far as further action of the network is concerned. If there is a network controller, protocol defines what functions the network controller has, what scanning structures exist for the purpose of identification, which user is next, etc. In order to construct a useful network, the protocol must be built into the system and be more or less invisible to the user. An example of such a system would be the NTDS Link 11 type of system.

A tremendous number of actions and activities are required of the network control. The network essentially is a collection of all the using activity and connectivity of the network. The control of the network generally involves the collecting of data and some evaluation structure in order to maintain the network in a peak performance condition. In order to accomplish this, a certain number of elemental activities are required. Succeeding items list these activities.

c. Request Arc Generation. This is a step in the structure of building the user paths. The request for arc generation essentially states that there is not a path and therefore a new path must be generated. The building mechanism is different from the actual steps for link generation and therefore the elemental action we are talking about here is the request for the arc necessary to make a path.

d. Request Channels. The network itself is composed of channels and the channels are embedded in paths. This activity essentially causes the channels to be identified within the paths that have been built.

e. Maintain Clocking Structure for the Network. A network generally consists of a condensation of activities that normally would be required for a collection of instances of communication connectivity. And in the process this condensation generally requires that a clocking structure be incorporated into the system of networking so as to allow for this condensation. The clocking structure must be maintained by this orderwire system.

f. Add Nodes and Arcs to the Network. Once the network is established, there are requirements from time to time to add new nodes to the network, and in the process new arcs may also be required.

g. Remove Nodes and Arcs from the Network. Once nodes and arcs are in a network, again on the basis of positive dissolutions, it may be desired that a command be given to remove a node and arc from the network so that the system can maintain itself in a timely manner.

h. Monitor Current Network Performance. At any instant, some type of structure of maintenance of the status of the system must be considered. That is, the system is really composed of a log of activity. Some type of mechanism should exist so that in any instant we might be able to determine what each part of the system was doing. If this is not possible because of the complexity, at least from the local level the status should be maintained in some form.

i. Test Network Reliability. Once a network has been put together, the system must have an elementary activity which tests the reliability of the network, and determines whether the network is operating properly, and indicates corrective action if defects are detected.

j. Measure Network Performance. Some mechanism must be in existence for measuring the performance of the system. This may be a set of collective structures, each one measuring a different thing, or just simply the collection of all good transmissions. Or it may be one overall unifying parameter which indicates the level of performance of the system. Whatever the case might be, a mechanism for measuring this performance must be provided. Two alternatives are the collection of a lot of pieces of data together with a formula for generating a single value or a vector where consolidating into a single value is mandatory.

k. Request Arc Standby. When building a user network, the entire network is not being utilized at all times. Therefore, some mechanism must exist to allow the system or network control to identify a communication (set of communication arcs) for its use. Whether that utilization should be exclusive is a matter of system utilization. Many times, as long as the arc can be recalled at a moment's notice, it is not necessary that the arc be utilized solely for the purposes of the network. There is quite a bit of activity that goes on that would conceivably say that some activity, such as orderwire transmission, could be transmitted when the arc was not being used as part of the network.

l. Maintain Network. Once a network has been in operation, a requirement exists to ensure that the network maintains its existence until it is dissolved. This maintenance operation is not trivial in an adverse environment. The problems are such that even in a natural environment there is deterioration of some arcs because of changes in distance between the nodes or interference in the media. Therefore, a mechanism for maintaining the network must be incorporated in the network control. This mechanism would essentially have to go through procedures of testing, rerouting, and reestablishing various arcs of the network structure.

   For the maintenance of system control one has to maintain the orderwire capability. That is, the orderwire capability must be in itself self-sustaining. When the users are using the system, the orderwire capability should still be in the background and should be able to be resurrected whenever necessary. Therefore, the orderwire capability must be maintained in some form. Whether in residual or active form is immaterial at this point of the discussion.

m. Dissolve Network. It is not particularly easy for a system to know what is available and what is not available. One could conceive of a garbage collection type of action which essentially says "Ah, these things are not being used, therefore, they are reassigned." However, this is highly inefficient in a system as complex as the orderwire structure that is being discussed here. Positive discontinuation should be the

normal operating procedure, the garbage collection function occurring only infrequently. Therefore, an elemental action must exist to decide on the dissolution of a network. The activities entailed here would essentially nullify the channels available or on standby and the paths of which are no longer required. The system in this situation should reconfigure itself in accordance with other criteria.

### 3.1.4 SYSTEM BALANCING

3.1.4.1 System Balancing Procedures and Process. To acquire a background for understanding the value of the communication system to the user, see 3.1.7, Services.

To achieve a balanced network, multimedia transmission paths are essential. Optical, uhf, and line-of-sight transmission media are good examples. The driving function behind network balancing is survival; that is, maximizing the probability of successful message transmission. The factors which tend to diminish network survival are many and varied. They can be exemplified in terms of direct enemy action such as overt frontal attack, indirect enemy action such as the disruption of the ionspheric layer via nuclear blast, weather-related malfunctions, internal degradation such as equipment failure, improper identification, bad location, moving stations, etc.

In considering the orderwire requirements for balancing the network or the communication system, we can consider two separate aspects of the problem. First, the system may not be in equilibrium; ie, setting at the optimum state such as in the initial turn-on of the system. The problem is how to get to the optimum state in a reasonable manner. Second, the system may have been in an optimum state. These are two quite different problems, and the set of procedures to optimize the system from the initial position state of the system is quite different from the set of procedures necessary to bring the system back to an optimum state after a finite change. We must also consider the interaction of the two sets of procedures in the case in which the system has not yet achieved the optimum state after a change occurs which calls into operation the procedures to bring the system back into the optimum state. The desired solution would be to have the two sets of procedures identical, in which case we would not have to worry about this type of interaction. However, very likely they are not the same set.

Consider that the system is in an optimum state prior to the change. In this situation what are the procedures that bring the system back into optimality? Can such procedures be linearly related? That is, can a fixed amount of work distributed over a finite number of nodes accomplish the task of returning the system to an optimal state? Or is the amount of work at each node of the system proportional to the logarithm of the number of nodes, linear with the number of nodes, or some higher function, such as quadratic, cubic, or exponential? These questions have to be answered.

Consider some of the changes that can create a nonoptimal state of the system.

a. A new user requests the use of the system. Here the system may no longer be optimal because the system is not currently serving this user.

b. A user of the system discontinues the use of the system. Here the system may no longer be optimal because an unused channel exists.

c. An arc (link) of the system degrades. Another solution may now exist that is better.

d. A link improves. In this case better utilization of the link may be possible.

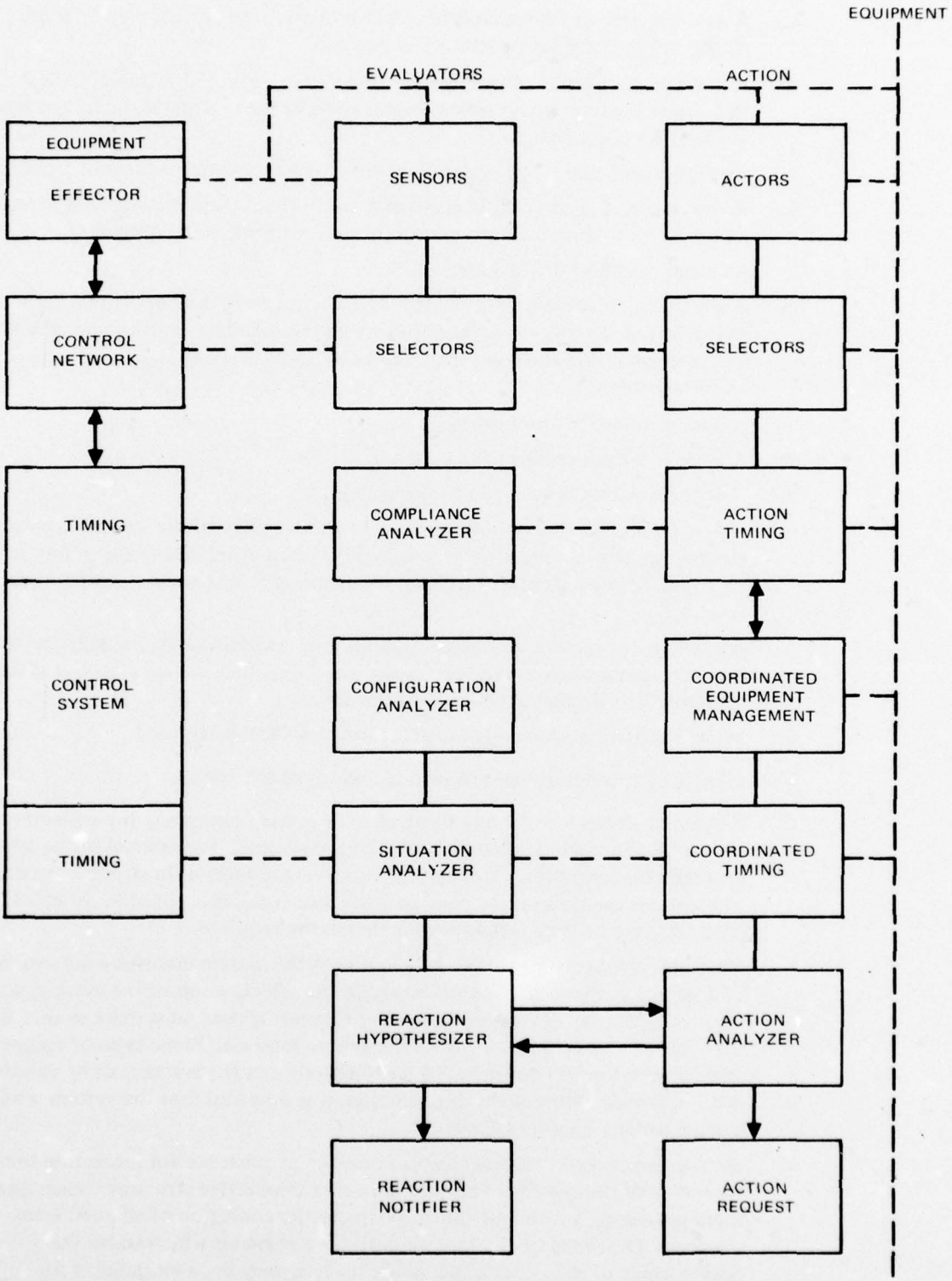



Figure 3-2. Network generation procedures.

e. A new opportunity for establishing a link exists. The system may be more nearly optimal if this opportunity is pursued.

f. A new communication threat arises. The system may be less susceptible if this threat is encountered rather than staying in the existing state; ie, the figure of merit function changes.

g. An equipment fails. The system may have to reconfigure to provide service.

h. A new piece of equipment is made available. The system may be able to readjust its resources to a more nearly optimum state with the new equipment.

i. A communication threat vanishes. See f.

j. A change in available media occurs. This differs from the opportunity criterion in that it is not a discrete change and it may be for the worse; ie, one may have available six 3-kHz slots and then one of the slots is removed by someone coming up in that slot.

k. A node is removed from the system.

l. A node is introduced into the system.

m. New information is available to the system.

n. A new command tree is established. The survivability of the system depends strongly on the command tree. Survivability here is related to the ability of the communication system to carry out the commands of the command control system.

o. A new data collection system configuration is established; ie, the patterns of required communication needs change and the system is most effective if it has configured to anticipate communication needs.

p. System constraints are added or removed. EMCON is imposed.

The following are functions and elemental actions of the system:

a. Elemental Actions of System Control. The system essentially is a collection of all the using activity and connectivity of the system. The control of the system generally involves the collecting of data and some evaluation structure in order to maintain the system in a peak performance condition. In order to accomplish this, the following elemental activities are required:

b. Maintaining History Records. Any action in the system involves a movement of data in a system. In order to ascertain the effectiveness of the system, some type of measuring device or mechanism for storing data must exist so that the data can be studied for improving the system later on. Some type of archival system therefore is required in order to ensure that the system can be updated and improved. Without this information, it is doubtful that the system would survive for any length of time.

c. By the same token some mechanism must be in existence for measuring the performance of the system. This may be a set of collective structures, each one of them measuring a different thing, or simply the collection of all good transmissions. Or it may be one overall unifying parameter which states the performance of the system. Whatever the case may be, a mechanism for measuring this performance must exist.

d. At any instant some type of structure of maintenance of the status of the system must be considered. That is, the system is really composed of a lot of activity. Some type of mechanism should exist so that in any instant we might be able to determine what each part of the system was doing. This not being possible because of the complexity, at least from a local level the system status must be maintained.

e. The status of the system is really a state of existence. If something should happen to the system, it is necessary that we retrieve and build up the system from some previous point. This means that backup status must be maintained in order that the system or part of the system can be rebuilt, to continue on with the functions when some drastic effect has disrupted the system. Therefore, some means of maintaining a backup status must be incorporated into the system control. Orderwire to do this must be considered.

f. For the maintenance of system control we have to maintain the orderwire capability. The orderwire capability must be, in itself, self-sustaining. When the users are using the system, the orderwire capability should stili be in the background and should be able to be resurrected whenever necessary. Therefore, the orderwire capability must be maintained in some form. Whether it be residual or inactive form is immaterial at this point of the discussion.

g. Expanding Connectivity. The system, in order to provide some level of survivability and availability to the user, must provide some mechanism for ensuring that the connectivity is as complete as the equipment in the system will allow. Therefore, the system control must have the ability to expand the connectivity independently of user requests solely upon the basis of possible availability to the user and survivability of the system in anticipation that the orderwires need to be transmitted later on. Thus, the system control must be able to utilize the capability of the system in order to build new connectivity so as to expand the connectivity. By the same token the system must be able to evaluate and remove connectivity that is no longer effective.

h. Control Priorities. The hierarchy of priorities for the user can be established more or less independently of the concept of the system. But the system has to maintain that implementation that command control would impress upon the system for the concept of priority. But there is another more important concept of priority that must be considered. There is contention between the users of the system itself. The only mechanism whereby a system can ensure that it is capable of continued support of the users is the ability to control the priorities in such a manner that its own communication requirements can be met in some form that is in accord and competition with the priorities of the user. Thus, the system must be able to come to control this relationship between its own needs and the structures that the user requests. There are certain cases, however, in which the priority of the system does not override the user's priority. In these cases, the system must be able to take on the instruction that even at the expense of the system itself these particular communications must be maintained.

i. Maintain Alternate System Control Centers. Underlying the concept of system control is the concept of contention resolution. It has been shown that there is only one way to resolve contention completely, and that is from centralized control. The centralized control does not have to be particularly elaborate. In many of the smaller systems it is nothing more than a round-robin test-do-test-do-test-do. In this system, of course, it would probably be more complex than that. Nevertheless, at some central point in the entire system a system control must exist. All activity can be controlled in theory from that single point. There is no contention deadlock. When that system requests or demands something, all other parts that that center commands or directs must comply, using the highest-priority structure. This is the only way that conflict of equipment can be resolved. The only known method of complete conflict resolution exists when the command control structure of the system is a tree (a structure in which there are no cycles). This is accomplished only if such a tree exists and there is a single point of control. However, if such a central point of control exists, the reliability of the system can be degraded if something should happen to that central control. Therefore, the system must have secondary or alternative system control centers that can be activated if the first system fails to respond to some particular interrogation. In essence, the total system should be constructed in such a manner that any node of significant size must, in itself, be an alternative system control center for a subgroup of communication nodes. When we break up a control tree by removing the highest-level node, a number of trees are left, each having a highest-level node. If there is still connectivity between the nodes between two different trees of control, then some type of hierarchy must exist to ensure that one of the new control tree heads will be configured as if it were the head of the whole tree. Alternate control centers would then exist. This way, the total conflict resolution system would continue to exist. Therefore, one of the functions of the control system is to ensure that some contention resolution structure would take over if the primary control center were removed.

j. The system must maintain a system control center. The control center must contain enough knowledge to ensure that it can properly control the activities of the remaining control system.

k. The system control must have some capability of communicating with the operators of the system. Therefore, a display of the system status to the operator is required. The operator subsystem, of course, may interpret an existing status structure for the operator, but nevertheless a display must be provided.

l. There must be activities to maintain the system information and distribute it in such a manner that control is effective.

m. One of the functions of system control is the distribution of keys.

n. One of the functions of system control is to maintain security and integrity. The principal problem is to make sure that no knowledge of the system can be captured and, thus, compromise the entire system. That can be assured only if some mechanism exists for detecting when a node is no longer performing in accordance with the requirements of the security and/or integrity of the system.

The basic idea of system control is that control is distributed. There is processing capability at each major node. Ideally, the system would do the processing at each of the nodes to ensure that the system provided appropriate optimization. If a solution can be found that is linear in its data processing requirements, then the addition of a new node to the system does not increase the processing load of the nodes already in the system. The desire to have such a solution does not make such a solution exist. If a linear solution does not exist, then other solutions must be sought. A log-linear solution is desired if a linear solution does not exist. If a log-linear solution does not exist, then an $n^2$ solution is needed. An $n^3$ solution is unworkable except in a very small system. An $n^2$ solution exists for the transportation problem. Whether the transportation problem archtype can be made to fix system optimization is not known at this time.

Backing away from the centralized picture of the system, the problem has features that resemble the Koopman's Welfare Economy; ie, a fixed set of assets to be used to their maximum usefulness. The welfare economy problem has an attractive solution space, in that global optimization can be achieved by local optimization once a set of prices has been established. The setting of the prices is a global problem, however. If a model can be established that allows for determining prices on a distributed basis, or if prices do not change rapidly with the detailed use of the system, then a partial solution may exist for the optimization of the system.

These questions have to be addressed more carefully to determine whether a workable solution can be found. Optimization is a global property. We must consider the total system in the optimization discussion. At any instant, the system is in a designated state and each of the equipments in the entire system is set to a particular setting. Also, at that instant, a set of users has active requests to be satisfied. Based on the state of the system, the environment, the position of the nodes, and the connectivity of the system, a better optimum may be possible. The set of user requests, together with the new desired state of the system, will produce a value to make changes to the system. From the state of the system, and requested services and potentials, a program of change can be devised to make an improvement. These changes are then carried out to accomplish the desired results.

The problem with this scenario is that it assumes that some overbeing exists with the appropriate knowledge to formulate the required set of changes to accomplish the desired goal. This overbeing is the system itself, and, further, the knowledge for making the changes is distributed. We could conceive of a plan whereby all knowledge is communicated to the appropriate nodes. Such a mechanism would be unwieldy and slow and would presumably generate greater overhead than some other methods of distributing control. Further, some operational scenarios will not permit communication in both directions.

NOTES:

The method of connecting nodes with the communications will be through spelled names. Qualifiers will be used to expand and contract frames of reference.

If no orderwire is taken, then useful communication is not accomplished.

If all the channel capacity is used for orderwire, no useful communication is accomplished.

If a communication pattern is established and no orderwire capability is present, the following is true:

a. It is not adaptive to new requirements.

b. Nature and the enemy will cause service to deteriorate.

The principle of use is stated as follows:

The expected value of the use of the communication derived from the execution of the orderwire communication exceeds the expected "cost" of the orderwire.

In this "value" is a criterion which is compacted in terms of bits transmitted and derived from user "value" judgments. Cost is the value placed on the data bits that could have been transmitted if the orderwire had not been transmitted.

If the orderwire had not yet established a communication link, then the value of the bits could have been zero, whereas the value derived from the orderwire is certain to have some positive value. Therefore, orderwire will definitely have some positive value, and should be executed.

In a communication system that ceases to function, any action, no matter how small, is better than no action.

Random processes will probably be a required part of the system's behavior. Deterministic rules will generally produce deadlocks and ineffectual adaptation.

Value to the action of the orderwire outside the strict value of the user can be negative if it is possible for the system to fail by actions of the orderwire communications.

The value of system knowledge is finite. The integrated value of the knowledge can be determined, in concept at least, by considering the improvement in the overall transmitted value if the knowledge were available vs absent. The value of the user knowledge is generally indicated by the priority of the transmission. However, other finer shades of value can also be considered. For instance, in blocked transmission, the value associated with the later bits is larger than the value of the first bits. This is derived from the general concept that if a block is interrupted in its transmission, the first part of the block will have to be retransmitted to get the data across. Thus, the value of the later bits of the block after n bits is $nP/(B-n)$, where B is the number of bits in the block and P is the value per bit at this priority. Consider bit values with a priority scheme. In the formulation, both value and cost are expressed in terms of the same quantities and a balance can be considered. In the theoretical sense, it appears that value and cost can be related to money in terms of dollars per bit. But the relationship is elusive in the sense that it is derived from the tactical values of dollars per action. This type of relation is not available to the communication system management and, therefore, does not serve as a useful measure. However, it is still desirable to have a uniform measure of the value of the bits within the system.

3.1.4.2 Interaction of System Nodes. In order to understand orderwire management, it is necessary to examine the decisions possible at any given node. Certain factors must be considered. First, users place requests into the system in a stochastic manner at a particular node. Nature and the enemy can place obstacles to the utilization of an already established arc, or open up the media to permit the creation of new arcs. Command control structure can change through command initiative or through enemy action. Finally, equipments can cease to operate properly. These factors can be considered inputs to the system since they are not under direct control and no decision

structure within the system affects them. With the foregoing in mind, decisions at a given node can be considered.

Some of the decisions on actions that must be made are:

The removal of users from arcs

The readjustment of traffic flow over arcs

The cancellation of users

The change of user priority

The change of equipment

The elimination of arcs

The alteration of arc utilization

The formulation of new arcs

Other decisions that can be made are:

What data should be transmitted regarding ownsystem structure?

What messages or information should be stored and forwarded at a later time?

What messages whould be forwarded immediately?

Should paths continue or be altered?

What is the general structure of a new arc composition?

Consider the decisions associated with a single arc. The node can make decisions as to:

Which user data should be channeled through the arc

Whether the arc should continue to exist

What degree of error control should be placed upon the arc

Whether information can be of an orderwire nature or must be transmitted over this particular arc

It may be possible to create a graph in which each vertex is associated with one of the elements for which decisions must be made and further to generate a pricing and pay structure for the optimization of the network.

Consider a particular vertex of such a graph. This vertex represents the utilization of a communication arc. The graph represents the exchange of services for a token. When a service is performed, a node pays for the service to other vertices of the graph. In effect, a token of appropriate value is transferred if a communication arc utilizes a particular piece of equipment. Using these pseudo prices and income, the system evaluates, for instance, the effectiveness of arcs.

This analysis is based on Koopmans' activity analysis for economic considerations (ref. 1). In this study, he shows that even in a nonlinear system there is a way to consider a pricing structure and thereby separate decision making regarding one part of the system with respect to the other parts of the system. This technique promises to have

[1]Koopmans, Three Essays on the State of Economic Science, McGraw-Hill, New York, New York, 1957.

application for determining the value of arcs and whether they should continue to operate or new arcs should be created to provide the service. The intent is to fragment the optimization technique to the point that it can be effectively implemented via local decision making structures.

Whether the Koopmans' technique is adequate for communications optimization is not clear. It does have definite characteristics that are appropriate for a communications system. The principal one is the ability to decentralize decision making so that it is unnecessary to have a single point making all the decisions for all movements of data in the system. Instead, a pricing structure is set up to make optimum utilization of the system which is self-organizing in the sense that prices change with demand and availability of items of communication equipment. While the technique offers rewards in system optimization, the problem must be examined in more detail before it will be known whether the technique is appropriate for the optimization of a control system.

3.1.4.3 At-Node Balancing. A number of users, each wanting a particular service, are competing for a fixed number of channels. In addition, the system provides a mechanism for storage that some users will tolerate – a short period of delay before service is required.

Assume some value has been assigned to any user/channel combination with highest value for the shortest path and proportionally lower values for longer paths. The method by which values are assigned is still an open question and will be discussed later.

As an approach to allocation consider the communication system as a Koopman's economy (sometimes called a welfare economy) where in the problem requires optimum allocation of a fixed set of assets. With this approach, a set of values can perhaps be derived and the entire system optimized by the collection of optimizations for each node.

Each user requires a certain capability to service his needs, but each channel has a limited capacity. The user does not want his data chopped up, but in special cases this may be desirable. Penalties can be seen by considering the results of multiple arcs or channels being established to serve a particular user. Two penalties are paid. The first penalty is that orderwires are necessary to establish the multiple arcs or channels to reconstruct the data at the receiving end, and this penalty may be substantial. The second penalty is the risk that the two parts of the data do not arrive simultaneously with the requirement for a mechanism to reconstruct the data. Because of the penalties involved, this discussion will assume that the technique of chopping up user data will not be used.

This restriction limits the problem but does not remove it. Now users and in some cases multiple users can be assigned to channels or arcs. Arcs can be subdivided into multiple channels. Allocation even under the simplifying restrictions is complicated. In some cases, fractional users are not permitted, so users must be stacked in such a manner as to optimize service.

The optimization problem must also be considered for two cases. The first case requires the start-up situation with no existing solution structure. The second assumes the system has been operating at an optimum, but a change has created a nonoptimal solution. Allocation must be adjusted to return the system to optimal.

Another aspect of the allocation problem that must be considered is the question of hysteresis. This addresses the fact of a change in channel costs (reduction in the

assigned values) because the channel must include the orderwire function. For a change to have a net increase in value, it must cause an increase of more than the loss due to using the channel for the orderwire.

Allocation must also be able to consider a store-and-forward mechanism. The user may be able to tolerate a decrease in throughput if data are buffered through such a mechanism, but changes in assigned values must reflect these system changes. For example, a user may be coming in with data at 2400 baud, but the only channel available for transmitting forward is 1800 baud. Allocation procedures must consider assigning values to store-and-forward at the same rate and at a reduced rate.

In order to effectively determine whether a particular configuration is optimum, a methodology is needed that shows when an optimal solution is obtained. Many of the techniques that can be easily implemented cannot compare different optimal solutions one against the other. Assignment of a high-valued user to an arc may lock out two lesser optimal solutions whose combined service would have a value exceeding the value of the service to the one high-valued user because this user takes too much channel or arc. A methodology is required which moves from a suboptimal to an optimal solution in a smooth and consistent fashion. It is desired that the solution approach optimum monotonically.

3.1.4.4 Start-up Procedures. The following outline presents the steps to be followed in start-up.

1.0 Note that there are only two ways to start up:

- 1.1 Through a magnetic tape or disk pack (Fixed common procedures)
- 1.2 Through a physically secure wire line

The operator must select which method will be used.

2.0 At the node, only one channel is open; all others are dead. For this one channel:

- 2.1 A fixed language structure exists
- 2.2 Knowledge of on-board equipment is embedded
- 2.3 Various "experts" are available
- 2.4 No knowledge of world or local communication is available except for the one channel
- 2.5 Single channel is one-way only (necessary by tape or disk operation)

3.0 Data knowledge entered into the new node:

- 3.1 Data for crypto to key variable
- 3.2 Storage for crypto
- 3.3 Necessary specification of parameters and procedural information to establish a transmitted signal to a virtual receiver somewhere. There may be several options. In the wire line case the parent node makes a special virtual arc for the operation. In the fixed disk or magnetic tape, several options will be considered.
- 3.4 Necessary specifications to establish a virtual receive arc or arcs
- 3.5 The new node monitors for a unique receive for itself with the original primitive language.

3.6 The wire or tape is accepted further to give as much of the world models as practical. When tape is completely read correctly, it is erased. Wire line continues with exchanges until manually interrupted.

3.7 When a receive connection is found, the new node attempts to solicit an acknowledge channel for its transmission attempts by transmitting appropriate information.

3.8 When a link has been fully established, a dialog is set up to test and optimize both arcs of the link.

3.9 System tells operator of completion of one link.

3.10 New node system activates a search for maximal spread of connection space using one link.

3.11 The users of the communication system may begin using the system at this point, but the system will give itself a high priority for secondary information.

3.12 After several connections are made, the new node calls a language compressor function which tries to optimize names and syntaxes to reduce orderwire overhead; ie, stabilize a normal language.

3.13 At this point, the wire line can be disconnected without too much disadvantage. It is desired that the wire line be maintained until the total knowledge base or system information of the parent node is completely transferred to the new nodes.

3.14 The new system sends a request to a coordinator node for price-cost value it should use in utilization of its system.

3.15 The new system sends a request for information on its part in the command line.

3.16 The new system begins to activate its "expert" systems such as hf weather and satellite positions.

3.17 The new system sends its availability state to the entire communication system for direction updates.

4.0 At this point, the new node is essentially not significantly different from any any other nodes with respect to state If the new node has a further function to meet, it may begin collecting the knowledge it needs to carry out this function. It may become a coordinating node, for instance.

Optimum solutions are not, in general, continuous functions of the values of the controlling parameters. To say this in another way, if an optimum solution is found for a given set of parameters, and if one of those parameters is altered by a very small amount, a new optimum solution may be a considerable distance from the original optimum solution. However, in real systems for which it costs to change from one solution to another, such as the requirement for reallocation and rerouting orderwire, it is not always advantageous to make the change. Many times, the first solution is still near optimum and the advantage of the new solution is small. In this case, the cost change can outweigh the advantage that might be gained.

If the optimality space (feasibility region) is convex, then incremental changes to the new optimum on a value-gained basis will provide a smooth transition from one solution to the next, with little cost disadvantage. If the region is not convex, then all the changes may have to be made at once. In this situation, "thrashing" may occur to generate a loss equal to or greater than the benefit that might be gained. In such a situation, care must be exercised to ensure that the new solution is stable.

Another problem with the consideration of optimal solutions is that the parameters do not in general change continuously, but in discreet steps. For example, a user requests a service, an arc becomes jammed, a user gives up a channel, or an ionospheric layer is stabilized. Depending on the point of view, these may be small changes or large changes. The disadvantage which results when a user gives up a channel and the system does not take advantage of it is the value of that channel. On the other hand, if the system does not respond to a user request, the effect may be catastrophic to the user and the system. In the latter case, movement to a more optimal solution is mandatory.

3.1.4.5 Quadratic Adaptive Routing Algorithms. A technique that appears to offer improvement in traffic routing is that of quadratic adaptive routing. In a recent artical, C. E. Agnew[1] of Stanford University contrasted the routing performance of two simple models in a store-and-forward communications network. Results showed that adaptive routing, which assigns messages to output channels so as to minimize the apparent message delay, induces equilibrium in the network which does not minimize the average message delay. Thus, the aggregate of many local optimizations does not yield a global optimum because adaptive routing does not take into account the effect of its current decision on the future state of the network. However, the form of the optimality condition suggests that a modification of the adaptive algorithm will result in optimality. This modification would consist of a substitution of a quadratic bias term instead of a linear one in the routing table maintained at each network node. Better routing strategies have to be developed.

### 3.1.5 SURVIVABILITY

3.1.5.1 System Survivability. Survival is dependent upon several aspects of the communication system. System parameters are those parameters embedded in the system's body of knowledge. The body of knowledge contains descriptions of the physical parameters of the equipments and media which are time-dependent. The time dependency deals with the establishment of specific equipment settings in a certain sequence, at a predefined time, as well as the general allocation of equipment assets at certain times to establish media paths. This system knowledge is machine-resident. This may be described as system policy relative to automatic system management.

Distinct from this is knowledge resident in a man but not in a machine. This may be described in terms of a human policy maker's decision regarding command control aspects of system utilization. Both sets of knowledge, machine and human, impact upon system survivability. Further, whether knowledge resides in a friend or a foe will also enhance or degrade system survival.

It is therefore mandatory to establish some type of figure of merit by which a system may be graded according to its survival potential. This in turn must be weighed against basic system limitations – natural, generic, and actual. Here a natural limit is one imposed by the laws of physics; a generic limit is one imposed by the class or family of equipment; and an actual limit is one arising when transmitted data differ from received data. To obtain some indication of the magnitude of the limits, it is necessary to define the capacity of the graph, the nodes and arcs, etc, for all media, as well as the equipment involved at any specific time.

---

[1] Agnew, CE, *On Quadratic Adaptive Routing Algorithms*, Communications of ACM, v 19, n 1, p 18, January 1976

Various measures of survivability should be examined. If equipment performance degrades, the figure of merit should reflect the degradation. If equipment totally fails, the node might be considered dead. If coordination between nodes is lost, due either to enemy action or to failure of internal timing standards, etc, the node is dead. A policy must be promulgated in order to reestablish communication. If an enemy destroys or captures a node, it is dead. If an enemy jams all media and communication can be resumed the survivability of a node may not be impaired. Therefore, the value of a system can be determined by its quality of service or the criticality of the information.

Survival in combat requires that the unit be able to receive intelligence and orders or commands from the commander. This requires the ability to receive a message from command. The current command structure must exist in essence in a single tree graph and the communication system must know this single tree graph to effectively accommodate the command structure.

Any possible changes to the command tree need to be known so that necessary communication with regard to the changes can be anticipated. An orderwire must be generated to update the command trees.

Survival is dependent on timely message delivery. The value of a message to survival decays with delivery time in some exponential fashion. Thus, the value for survival is greatest when time of delivery is shortest; and as time passes to take action for system survival, the value becomes less and less.

### 3.1.6 OPERATOR INTERFACE

The following list covers standards for alphanumerics and graphics:

a. Military Standard 188, Appendix E, Table 2, for ASC11 and OCRA (currently)

b. Military Standard 1472 for Human Engineering (always)

c. Military Standard 454 for Safety (always)

### 3.1.7 SERVICES

Delivery of Message. Delivery is the process of transferring data from one user site to another. This transfer can be on the basis of a continuous or near-continuous type of connection, as in ordinary speech; or it may be considered to be discreet as in message delivery. The principal reason for existence of a communication system is the delivery of data. Under the category of the delivery of data, there are several types of delivery processes. First, there, is the specialized delivery for which a receipt is required. A receipted system is one in which there are two-way communications. First, the data are transferred from one site to another; then a message is generated at the receiving site and transmitted back to indicate that the message has been received. The degree of receipt can be reduced by considering that the system will ensure that the message will be delivered without a receipt; that is, the user puts in a message and has an assurance that the message will be delivered. However, he does not necessarily have to receive the receipt to know that. This is a point-of-entry rather than a point-of-delivery receipt system. Also, there is the general type of delivery system in which the message is put into the system and there is a concept that it probably will be delivered if it is a discrete message. It may be a "number of times tried" type of

system. Here the system will try to deliver the message not more than n times. It will terminate after n tries, or earlier if successful. With respect to deliver, on the continuous systems which, like speech, are more or less self-receipting, since they are generally full or half-duplex, the system essentially returns a signal by virture of a feedback link, for instance, that tells the sender that he has got through. This is not a formal but an informal delivery system.

Timeliness of Delivery. The second user value is timeliness of delivery; for instance, delay or blockage, whichever is more appropriate. Timeliness refers to the time to deliver the message. Some messages have to be delivered in less time than others. In the case of the blockage structure, the blockage has to be removed prior to a certain point. The timeliness of delivery is tied very closely to delivery itself. If delivery time is infinite, of course, no message is delivered. The concept of delivery requires that there exist some timeliness. In the case of the communications system, we can suppose that there exists an absolute delivery time relating to the physical communications; that is, a delivery in approximately a month. However, this is not particularly useful when most users require delivery in far less than a month. First, we start out with delivery times of less than a second. A second category would be delivery times in the order of a second, of 5 seconds, of 10 seconds, of 30 seconds, of 1 minute, of 5 minutes, of 10 minutes, of 1 hour, of 6 hours, of 1 day, of 1 week, of 1 month. Each of these levels puts a particular requirement on the system and the system must be configured to relate to each specified delivery time requirement. Another question relating to delivery times remains. Shall the delivery time be the point at which the message begins to be delivered or the point at which delivery is completed? Now, in the case of continuous connections, the concept of the time at which the message has arrived has little meaning. The concept of when the message initially starts delivery is the criterion which must be used, since the other has no useful meaning. However, in the case of discrete message types, or in some types of networking, the moment of delivery is considered to be timely in relation to the time when the message is completed in its delivery, not to the time it starts. This implies relationships between timeliness and duration. For instance, the duration of the message may be 5 minutes, in which case the timeliness would have to be 5 minutes or greater. This structure of what is meant by timeliness would have to be resolved in order to adequately define the value to the user. However, the resolution is more of a semantic resolution than system concept resolution.

Data Rate of the Transmission. This is probably more correctly described as data rate of intake, since the data rate of transmission can be independent of the data rate of reception. However, there are some relationships existing in the sense that the data rate of transmission must in some way be greater than the data rate of reception of the data. The data rate as seen by the user is the principal user value quantity. The data rate can be varied. In present Navy systems, for instance, we might consider data rates on the order of 50 baud, $75 \times 2^n$ baud, 16k baud, 50k baud, 100k baud, NX 100k baud, 5 megabaud, and higher. Each of these rates puts a different requirement on the system. Some of the requirements cannot be met by any existing communication systems. When the user specifies a particular rate, when he enters or is originally introduced into the system, the rate must be compatible with the system. It is possible to conceive of a system in which the user can change his rate dynamically. This need not be considered at this moment. The point is that the user of the system has a particular rate and has to talk with the system to change that rate. This rate is generally dictated by the equipment, such as a 75-baud teletype, for instance. In the case of 5-megabaud rates required in certain types of transmission, delivery in a timely manner through the electronic communica-

tions system is unlikely except for short distances. That may be adequate in some cases. It may take a special physical movement of data to accommodate those users that require very high data rates. Likewise, the data rate has great impact upon the store-and-forward requirements of the system. If it is a store-and-forward type of value associated with the users, then, for high data rates, the longer the message and the longer the delay permitted, the greater the amount of storage capability required in the system.

Selection of Destination and Routing. It goes without saying that, except for the simple-one-arc system, the user must have the capability of establishing some selection over the destination and possibly over the routing. However, the concept presented in this orderwire structure is based on the assumption that the system will determine the routing, not the user. However, there is a second type of routing which must be considered. This is the selection of the destination and the selection of any site copies or other addresses that the data should go to independently of, or in addition to, the primary destination. This could be called secondary routing. Secondary routing, of course, is important to the user and should be considered as to what impact it should have on the system, in particular the concept of conferencing in speech. Here, we have the situation in which the data are transmitted to a number of users and any one of the users would also have his data transmitted to the others of that group. This conferencing capability is a multirouting function, in which even the primary selected destination is not used but a group of destinations is used. Now, the amount of structure associated with the selection routing is large. The structures can be designated in different ways – multistructure or single-destination. For instance, a broadcast is a multistructure for which the destination is many-valued. We can also consider the dimension of selection associated with a local destination; a destination in the local group; or a destination in the task force, in the operation area, or in the world. Each of these has the problem of control of information flow as dictated by the command control system. For example, the command control system may dictate to the communication system that certain data must be restricted in distribution. That also would have to be part of the system concept.

Error Rate. The error rate of a message is another area in which the user may want to exert some control. The system will attempt to deliver the message at the best error rate available within the constraints of the operating capabilities of the system. When a user specifies an error rate, it is assumed that the user is specifying the maximum error rate that his equipment or system can tolerate, in which case the system may have to reconfigure to ensure that the error rate seen by the user is less than that dictated by the user. There is also the concept of different types of error. The system may have to consider how to reconfigure, for instance, to avoid burst errors. This involves providing some method of control over rerouting messages that are clobbered in transmission. Another problem is that the system may not be able to determine real error rate exactly but only to estimate it and can only give the user a concept for a utilization of the system with a probable error rate; and the system may not know that the error rate is higher or lower than that which the system estimated. This is the case with a system which has to probe the channels in order to determine an error rate. The probing of course interferes with the transmission itself and therefore has to be limited in some way so as not to upset the entire channel. The error rates that one can conceive of as being realizable are 1%, one error in 1000, one in 10000, one in 100000, one in a million, one in 10 million, one in 100 million, one in a thousand million (billion), and one in 10 billion. Error rates better than one in 10 billion can be achieved through ARQ techniques, but with reduced throughput.

Delivery Protocol. A certain amount of protocol is necessary for the exchange of information. As indicated elsewhere, this protocol actually embeds a considerable number of relationships within the data of the transmission. The simplest types of protocols are essential, with respect to what the user wants; these are things such as bits, bytes, and block data. These are simple low-level protocols; many of the higher-level protocols are embedded in the user system and not in the communications system. The lower-level protocols are essentially those of coordinating delivery of data. Much of this work is not really pertinent to the communication system or under the orderwire operations. However, some of it is; for instance, in many systems bit count integrity is required from end to end. If this is the case, the system has to ensure that no extra bits are inserted into the bit stream and no bits are removed. So the system needs to have some knowledge of this. For instance, in a byte-oriented system, the system knows it is dealing with bytes and can effectively chop the data, for purposes of rerouting and for purposes of store-and-forward, into integral units called bytes. It is important for the system to know it can deal with the data in terms of bytes rather than bits and we might then have the problem of byte count integrities. However, in a few systems there is a problem of character conversion.

While not a primary function of the communication system, another type of information concerning protocol is the data grouping. For instance, many vocoders now being constructed use blocks of information. The breaking of the blocks is important, especially in conferencing, when one system must be interrupted to put in the data from another source. In this case, it is necessary that the system know the limits of the blocks. In systems in which end-to-end security is required, the blocking information still must be available to effectively accomplish secure voice conferencing in the atmosphere of end-to-end security. If this arrangement is possible, it can be accomplished only if the blocking information is delivered to the system separately from the data itself. There are also higher-level protocol structures: for instance, structures associated with a netting operation. Here, the protocol is associated with the ordering as to what kind of bits go where and when. In the message delivery system, we have to deal with the actual contents of the message as far as the blocking of the message is concerned. Sometimes the message header itself contains the destination of the message. In these types of systems, the communication system has to become more intimately involved with the data, and the delivery protocol is absolutely required at a higher level; namely, the understanding of what is being transmitted to the extent of being able to interpret the header of a message. Delivery protocols, however, are not as involved in the overall behavior of the system, since all the data can be reduced to the simple bit-by-bit delivery for the purposes of communications from one place to another. Delivery protocols, therefore, have value to the user and may increase the cost of communication systems by virtue of requiring them to process more complicated protocols. However, they cannot form an intimate restriction upon the use of the system, since the system is valued more in the area of data transmission, and this is in terms of bits, rather than in terms of protocol.

Duration of the Message. The duration of a message is related to the data rate of transmission and timeliness of delivery. The duration of the message, however, is an independent dimension. We can conceive of several kinds of duration, and these are, in some respects, related to protocol. The duration of a message can be anywhere from something less than a second to a second, 5 seconds, 10 seconds, 3 minutes, 10 minutes, 1 hour, 6 hours, 12 hours, 1 day, 1 week, or 1 month, or the system is simply a connected system in which the channel goes from one end to the other end and the duration is infinite for all practical purposes. The duration essentially tells how long the message will be utilizing the capabilities of the system. The duration of a message is simply that portion of a message which tells how long the user

will be utilizing the facilities of the system. That, in conjunction with the data rate, will tell how much capacity the system has to have in order to accommodate the transmission requirements. In the case of infinite duration, the system has to dedicate some channel to this particular user, and, furthermore, when the error rate becomes sufficiently high, the system must switch from one channel routing to another to make sure that the message or the duration of the communication is continued uninterrupted.

Availability of System to User. Another dimension of the communication problem that the user sees, is availability. Availability can be reoriented in terms of timeliness of delivery or in concepts of availability or the blocking of the system. It is slightly different from timeliness of delivery, since we could be talking about the blockage of the system in the sense that the user must, in fact, hang up and start over again. The user comes into the system and the system says it is not available for use. This is blocking. This availability is modified by some concept of time. For instance, we can consider availability in the case in which the user comes on the system and the system will always take his message in a store-and-forward mode making the user feel the system is always available. The system itself may cause a blocking structure that says that the user cannot use the system, and, therefore, the user must wait and try again. The concept of availability, however, must be modified to some structure associated with availability. For instance, we might consider the case in which the system has a period of 1 second grace in which to respond. If the system has the availability of only one bit time, then the system may say that this system is not available; whereas, if the system is able to have a dialog within 1 second, then some time within that second the system will respond. A time is put into the structure of availability – availability within 1 second, within 5 seconds, within 10 seconds, within 5 minutes. This type of availability, however, means that the system will have to have some type of dialog with the user to inform him of when the system will become available. In the case of 1 second or 5 seconds, the dialog may be simple. In the case of 5 minutes, however, a ring-back capability may be required in order to assure the user that the system is now available. If it is a hardware system, however, it may be that of letting the system essentially take on the data as soon as the system is available, and thus the time period is meaningless. However, it was desired that the system always have availability within 5 minutes (or some fixed time).

Store-and-Forward Requirements of the System. Many activities of communications require the delivery of a message but not necessarily instantaneously. Thus, the system has to store the message in some form for a period of time. Storage location is dependent entirely upon the dynamics of the system itself. For instance, it may be able to transmit the message to the first leg of its journey, and store at the next node down in the system. In this case storage is not local but removed one step. The store-and-forward requirement is very important in most record-type traffic, and to a limited degree in some store-and-forward voice-type traffic. Store-and-forward in the netted situation, of course, is permitted but only in a certain way. The store-and-forward allowed in voice could alleviate the system requirements to have a continuous channel between the users of the system. This would allow a certain latitude on the part of the system to respond to voice and, thus, avoid the requirement that the system provide a full-time channel for a single user. In order to utilize the capabilities of the system along this line, it would be required that the blocks of voice which do not contain significant information associated with speech be so identified so the system can essentially answer for those and condense them in some kind of coding structure, which would then allow for the more-efficient use of the channels independent of the user. The capacity of the system to store and forward is a parameter over which the user should have some say. However, for the most part this capability will be built into the system and will not allow the

user to stipulate much in the store-and-forward structure. In the construction of the system, the user would have to provide the requirements for the system. However, there is something to be said for allowing the user to stipulate some type of restraint upon store-and-forward utilization of a system. For example, the user may say that he wants minimum store-and-forward structure, since there are going to be a few bits here and there no matter what you do allowing for a small delay in the transmission of the data: 1 bit, perhaps 4 bits, even as much as 10 bits. Then, for instance in speech, the user may stipulate that store-and-forward may not be greater than 100 bits, thus causing no significant lag in the behavior of the system as far as the voice goes. Even 100 bits can be permitted if the user system does not require immediate responsiveness of the system. The user could specify 1000 bytes, 10000 bytes, 100000 bytes, a million bytes, or several megabits. In the case of several megabits, significant impact upon system design is involved. It is clear that requirements for megabit capability for store-and-forward are not really going to be answerable by the normal part of the system. It may be necessary to essentially put these off on some type of archival storage for transmission by physical transport. However, this is an option. Furthermore, there is some requirement that all data have some aspect of store-and-forward structure in the sense of archivalness. How much store-and-forward should be done is unknown at this time. This is a system development and system building function that has to be answered at some level in the management for the system requirements for the design of the communications system.

Distance. The user generally selects a destination and is not particularly interested in how far away it is. However, from the point of view of the system, there is impact with distance involved – distance here being not so much the physical miles associated with the source and the sink of the information as the types and the numbers of steps necessary to transmit the data from one point to the other. The real consideration here is to transmit the data from one point to the other. The capability must be given the user to somehow or another bound the distance without permitting him to excessively stress the system. In other words, the user might say make his call provided it is not a greater distance or greater complexity than as specified. This is something that has to be worked out as to exactly what kind of behavior the system should respond with. Also, the distance aspect may be a command control type of function, in which one level of system control tells the communication system to limit the distance of another user's structure of the system; and the command control says that this particular user can call anybody within this framework. For instance, a distance control is provided that is perhaps not considered a user but is dictated outside the system nevertheless.

The user has to consider a priority value associated with the system priority. A system, of course, does not know a priori how much value it should give any particular user. This has to be worked out by command control. Once it is worked out, then a priority is associated with the particular user. Priority here is more a software concept than the priority generally utilized in the communication nomenclature of the present Navy system. However, priority of some type is required. The priority of the present system is essentially some type of indication; and that type of thing will still have to be supplied to the system to specify the priority of any particular user. The system, of course, must respond to some type of priority structure. There is no way for the system to operate if priority is not a critical governing factor. Now, for instance, in the communication system called the Bell Telephone system, there is very little priority involved at all. All users are of equal priority. But in a command control system, equal priority has no meaning. There are certain messages that have definitely greater value than other messages. That value judgment must essentially be dictated by the command control system. Another aspect of the priority problem, which is not necessarily user related, is the nature of the tradeoff among priorities. If one has one level of priority and another has

another level of priority, the question concerns the sacrifice required. For instance, a higher priority may sacrifice a thousand times the utilization of a lower priority. Is there a tradeoff here? It says 1000 times sacrifice of the lower priority is acceptable or not acceptable. This structure of relativeness of priority is completely unknown. There is some indication, for instance, in the case of interruption, that the concept presently existing in certain systems is that the higher priority not only interrupts the lower priority but interrupts in such a manner as to disrupt the continuity of data at the lower level. Whereas, at some point, as priority decreases, the circuit does not interrupt lower priority. It simply waits until a natural break occurs in the line of communication structure. Some of these problems have to be addressed to determine the relationship between priority levels and the result integrated into the system by the command control structure.

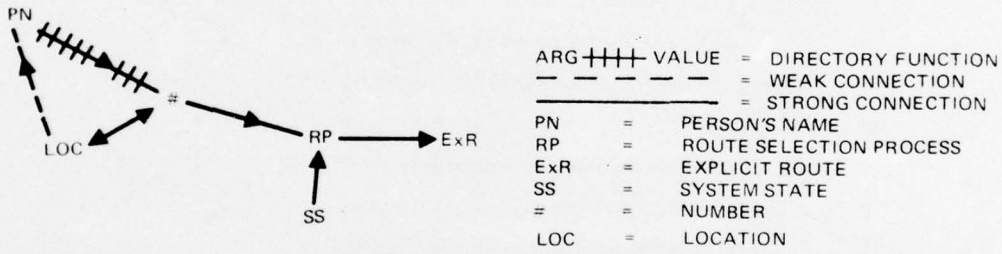
Security of the Data. There are various ways of talking about security of data. There is the security of the protection of the traffic that is flowing; that is, of the data actually transferred. Then there is the concept of the security for the cover of the information when the traffic flows. Another problem is that of different levels of access. For instance, it is a directive that all communications shall be covered, and, preferably, that all traffic should be covered in traffic flow sense. The question is how to restrict information from one set of users while providing it to another set of users. The system itself is geared to transmit data from one set of users to another, but there is the danger that the communication system itself can be distorted in such a manner that information can be extracted without the knowledge of the user. In order to facilitate system utilization, a type of end-to-end security is being developed and will be part of the system. The question remaining is the problem of traffic flow security. Internal traffic flow security is not important as long as external traffic flow security is had. It does not appear that traffic flow covering within the system itself should be required. The ramifications associated with this particular choice are whether or not the traffic flow must be secured within the system itself and from other users in the system. The system cannot effectively utilize its capacity to its best advantage if it has to have internal traffic flow security. The solution to this problem is to provide the capability with the realization that the penalty for using traffic flow security is costly. That the traffic flow security as seen from the world external to the system should be maintained for the total system itself is preferred. This traffic flow information is still not available to those external to the system. Traffic flow information may be available to other members of the system, but not outside the system. The user places a value in the classification of the data. This classification, however, has very little impact upon the primary functions of the communication system.

Now the user interface has to be constructed in such a manner as to allow classification and the control of classification. This does not have particularly great impact upon the hazard of the system with respect to transmission, since classification does not change priorities or change the decision making as far as the principal routing of information is concerned. It is primarily associated with the problem of end-user access, which is of little interest as far as the orderwire structure is concerned. There is some concern, however, about classification in the structure of error control. The system may have to do something special in order to assure that classification of messages is not in some way mutilated and the message receives at lower or higher classification as a result of the mutilation. The system, therefore, must consider methods of decoding and classifying to ensure that the classification is delivered in error-free condition regardless of whether the message is received in error-free condition.

Special Comments Concerning the Directions to the Users. Many of the discussions in the preceding paragraph have been made under the assumption that the command control system has some control over what the user must accomplish and consequently some control

over utilization of the system. It is paramount that the command control system have this control. There is an issue, however, of whether the actual user of the systems should also have some information about the extent of the penalty, or cost, of the capabilities that he is using. Users of the system should be aware of the difficulties they can impose on the system so that they use it in a way that does not disrupt or degrade the system unnecessarily. There is a certain amount of communication that is required but is not generally considered to be necessary at any instant. The system does not know this kind of information unless it is given to the system. For instance, in a store-and-forward system, if the user says that the timeliness of delivery should be less than 1 hour, then essentially the system decides as to when in that hour to deliver the message. This means that the user must have a feeling for the system so that he does not specify that the timeliness of delivery should be 1 second arbitrarily. This is important in the sense that the overload of the system or the undesired utilization of the system should be considered. Now the principal problem here is that a crisis may be going on in one part of the system and another part of the system is not aware of the crisis.

Directory. This is a function which translates common references to explicit references. For telephone communications, it is generally a translation of persons' (entities) names to telephone numbers.




In general, the variances of reference and the size (number of bits) specifying the terminal point are too vague or too long to be practical. The directory is a document which aids a human to select the proper terminal point (person, command) from among the various ways of specifying that point and to compress the notation to a size suitable for control.

Rate of Change of Directory. Since the directory is to aid humans in the selection of appropriate control codes to route messages, it should be a document that is changed often enough to ensure efficient use of the communication system.

a. Efficiency drops when proper routing of messages (calls) is not accomplished because a person cannot find the proper listing in the directory.

b. Efficiency drops when "information" is called to acquire information normally found in the directory.

The causes of efficiency loss are:

a. Improperly designed directory

b. Loss or destruction of directory books

c. Not containing recent (new) persons/commands, etc (failure to update directory)

d. Incorrect entries
e. Communications system requires too-specific addressing
f. Lack of commonality in "information" function
g. Global link being used for task that can be done locally
h. Too much transmission of directory information

Human limitations in using a directory contributing to inefficiency:

a. Not enough cross referencing.
b. Categorization lacking.
c. Too many entries.
d. Indexing not adequately channeled (page limits, etc).
e. Not enough confirming information in item; fully qualified vs ellipsis specifications.

Problems of directories:

a. Directory information has to be available to every user.
b. The question of use of common reference instead of director information in automatic directory.
c. The problem of printing and collating.
d. Users generally use the same items over again.
e. Users generally use recent references again.
f. Effective total direction is required.
g. Personal direction uses station directories.
h. Problem of scanning in automatic directories.
i. Problem of feedback in automatic directories.

Orderwire Directories:
An orderwire directory differs from a human directory in that:

a. Orderwire directory is not printed
b. Orderwire can change from day to day.
c. Ambiguity is not intentionally present.
d. Compression is not necessary (it is already compressed).
e. Confirming information is stabilized.

Function of the orderwire directory is to select appropriate routing and methodology. Automatic directory has more chance of use in this environment.

### 3.1.8 ELEMENTAL ACTIONS

An elemental action is the lowest level of activity expected at a particular level. If an activity can be conceived that can be distinguished from other activities and is not made up of smaller parts at a given level of organization, then that activity is an elemental activity.

An elemental activity at one level may be made up of many activities at a lower level. These types of elemental actions are enumerated in the first of the listings that follow. The succeeding listings enumerate the elemental actions. The purpose of the elemental actions is to aid in formulating what is to be transmitted in the orderwire system. Composite actions are sums of these listed elemental actions.

Type of Elemental Actions (EAs)

Equipment control

Arc control

Equipment scheduling

Network control

System control

Orderwire

Path scheduling

Media expert

Time sequencing

Operation control

Operator control

Maintenance control

EAs of Equipment Control

Turn equipment on

Set parameters

Test if working properly

Run

Shake down for maintenance

Turn off

EAs of Arc Control

Steer data into and out of equipment

Test connectivity

Control equipment

Generate orderwires

Receive orderwires

Measure error rates

Optimize error rates

Report connectivity failures

EAs of Equipment Scheduling

Fetch an equipment

Release an Equipment

EAs of Equipment Scheduling (Continued)

Status of equipment fetch

Remove equipment

Place equipment into system

Report equipment performance

Test equipment

Resolve priorities of equipment use

Recall equipment

Report recall of equipment

Queue of equipment requests

EAs of Network Control

Build user paths

Build user networks

Build protocol for user networks

Request channels

Request arc generation

Request arc standbys

Test viability of network

Dissolve network

Add nodes and arcs to network

Remove nodes and arcs from network

Maintain network

Maintain clocking posture for network

EAs of System Control

Maintain history record

Measure performance

Maintain status of system

Maintain backup status

Maintain orderwire capability

Expand connectivity

Control priorities

Maintain alternate system control centers

Maintain system control centers

Display system status to operators

Maintain system information distribution

Distribute keys

Maintain security integrity

EAs of Orderwire

Translate special condensed language to normal language

Translate normal language to semantics control language

Translate semantic language to execution sequences

Translate semantic language to normal language

Translate normal language to special language

Sequence inputs

Separate outputs

EAs of Path Scheduling

Generate status of path utilization

Request remote arcs

Request create arcs

Load arcs

Create channels

Delete channels

Maintain channels

Make channels

Create paths

Determine possibilities of channels

EAs of Media Expert

Request weather (hf and other)

Determine effect of weather on media

Determine joining environment

Create jamming

Recommend AJ techniques

EAs of Timing Sequencing

Coordinate different rate processes

Provide time-out function

Provide time-out clocks and frequency standards

EAs of Operation Control

Establish EMCON

Establish user connectivity constraints

EAs of Operator Control

Display status

Call up status details

Generate reports

EAs of Operator Control (Continued)

Translate from semantics descriptions to human language

Translate from human language to semantic description of creation

EAs of Maintenance Control

Provide alteration of control of system

Provide test tools

Provide description tools

Provide equipment reports

Provide equipment injection and removal

## 3.2 PROPERTIES OF THE KNOWLEDGE BASE

### 3.2.1 THE STRUCTURE OF KNOWLEDGE

Knowledge, as communicated, is generally structured as a series of bits bound into characters, words, phrases, clauses, statements, paragraphs, sections, chapters, books, shelves, topics, and libraries. The minimum amount of knowledge that can be conveyed is a statement. Parts less than a statement do not have the aspect of truth or falsity. Sometimes a part has an implied meaning of existence, in which case an implied statement exists. The string of characters does not have any value unless there is an implied attachment between the string of characters called words and existential or intentional existence to which the word refers. For any given frame of reference, the existential references are those outside this frame of reference. The intentional references are those which are described in the knowledge representation; eg, "ships" is an intentional reference – that is, ships stands for the collection of objects in the frame of knowledge such as USS NIMITZ and USS OKLAHOMA CITY. The USS NIMITZ, on the other hand, lies outside this frame of reference. In another knowledge frame of reference the USS NIMITZ is an intentional object. Propulsion system 56784, navigation system 47839, etc, are the existential references.

Taking all the frames of reference together, the atomic existential references are the signals that go to change a piece of equipment or the names given to the sensor data.

The statement in turn must be linked to the generation of connections between the words. There is an immediate connection in that the words are all grouped together in the statement. This rather trivial connection is not adequate. The operations of reasoning and command cannot be applied to the sequence of connected words that a statement forms in the input stream. A set of conventions must exist that translates the statements into a knowledge representation that has aspects and rules whereby the operations of reasoning and command can be applied. The most appropriate representation in the memory is the representation of an ordered graph with each node and arc namable. Some arcs of the graph refer to objects outside the graph. Generally, this connection will be made only through names. The graph will not have pointers to these exterior objects, but the graph will have pointers to names. The system will interpret this structure to generate activity associated with the name; eg, (turn on [modem type K number 5 powers] ). The interpreter when it finds an operation word such as "turn on" calls a program that can generate the appropriate signals to the indicated equipment.

Two issues arise with respect to representing knowledge in terms of a graph. The first issue is "Does the graph have to be printable?" If not, an ancillary question might be "How does one check out the system?" Only tree graphs are printable in the practical sense of the word. Directed acyclic graphs (DAGs) can be printed in theory but may take years of time and tons of paper. Cyclic graphs cannot be printed at all. The second issue is whether both the nodes corresponding to the nodes in the communication network and the arcs corresponding to the communication links must be named. The second issue is related to the first in that DAGs and cyclic graphs can be represented if we are allowed to name the nodes, and are allowed to print only the trees, using the names to stand for the nodes and their descendants. The greatest difficulty with this technique is that successive printing of the graphs will not necessarily produce the same print even when the graphs are of equal structure. The names give rise to different breaking points and therefore the printings are different.

However, since the consideration here is for a self-organizing, automatic rerouting system, we must be able to build and destroy arcs and nodes of the graph. Therefore, we must be able to identify a node or arc, and thus some naming is necessary. Further, for purposes of checkout, the graphs must be printable. Again, naming is required. Further automatic naming will sometimes be required; ie, a subgraph may be generated having no natural names and therefore not printable. Thus, the system must generate names to tag the appropriate nodes. The naming of nodes is straightforward and has been done for many years in compilers. The naming of arcs is another matter. Even though a node is named, a process can examine the named node and find it difficult to determine what the name is, or even whether it has a name. This latter problem sometimes is not too important, since the incoming orderwire has the names and thus can identify the node. But when a description of a set of node relationships is required to be transmitted, the selection process that picked a node may not know the name of the node.

The naming of a node is generally accomplished by associating a pointer from the node to the string of characters that spell the name, the node being a unique address of the system. Some aspects of the node are in general not available unless the item is a so-called atom of the system, in which case pointers, features, values, properties, and spelling are available. In a knowledge system that is to be updated by orderwire, each principal node must have the structure of an atom. The apparent structure of the arcs leaving the node is that of a list of items in which the ordering is arc ordering. Traditionally, arcs have not been named, but it is known that if we are to represent graphs with named arcs in a knowledge system, then the knowledge system should also have the named arc feature built into its structure. Further, once we consider naming arcs, then the arcs have the aspect of looking like nodes.

The representation of knowledge could be the collection of input knowledge streams together with all derived knowledge expressions. A far different arrangement would be to completely digest all the knowledge in some highly interconnected representation. In both these cases we are faced with a serious problem: too much memory is required. In the first case, the derived expressions will become numerous as the system is being used. Further, significant relations are not apparent at the time the new information is arriving. In the second case, the growth of memory is high at the arrival of any information. The second case may not be possible since an infinite number of derived quantities may be possible and no criteria are posited to control the growth of the graph. If the first case is modified to delete all derived knowledge after an evaluation has taken place, the memory size becomes manageable, but the execution time becomes unmanageable. One of the advantages of this

method is the availability of techniques with which to conjecture. In the graph model, the second case is tied into the graph and is not easily removed. The problem discussed here is not the same as the information retrieval problem, although it has some of the same features. The information storage and retrieval is a degenerate form of the problem.

The second method has quite an advantage in the quickness with which significant information can be detected. The methodology by which this problem is resolved has to be addressed in the design of the orderwire system. We are not free to dismiss it. Some possible directions for a practical solution are to consider some of the better parts of each method.

The method of using input statements with their syntax for the storage of knowledge is attractive because the process is linearly ordered and can be easily time-labeled. To handle derivations and conjectures, we simply add new information in the same syntax as the original input or add new conjectures in the same syntax. The method of getting rid of failed conjectures is simply to back up the stack of conjectures and derived inputs.

The hidden problems of this method are large. The most notable is the problem of determining whether new statements are significant. The second is that the method of deriving significance can take up enormous amounts of processing time and space. The removal of outdated and changed status is relatively simple; however, the resolution of contradictions by virtue of new information is difficult. A new statement may only contradict part of a previous statement, but if we remove the previous statement, the system is incorrect; and if we leave the contradiction in, the system will cease to function. However, once a decision is made to remove a statement, its consequences can be removed without too much difficulty. This can be accomplished by scanning forward and eliminating all derived statements with the given statement as a base. The labeled graph representation of knowledge has great power to decide whether an incoming statement has meaning and, if so, what to do about it. However, if every input statement is analyzed and put into the graph of knowledge, the derived information from that new information can be enormous. The totality of the derived information cannot be stored in the graph, for there is no computer that has large-enough storage and fast-enough computational speed to accommodate this feature.

In addition to the problem of placing both information and derived information into the system, there is the problem that once the information is in the graph in a digested manner, it cannot be eliminated easily. First, it must be dated, for this will be the only way to tell whether the data have been updated. Secondly, the data will have to be packetized, for this is the manner of memory storage devices. It is not at all clear how to packetize knowledge graphs in a reasonably efficient manner. Another problem is that if we use conjecture analysis, then the information or conjecture whether true or not is placed in the graph. The question then arises as to how to remove conjectured data that are false. The normal method of graphing knowledge implies that parts of the graph will point to the new relations, not that the new relations will point to the graph. If the new data were to point to the graph, then garbage collection could be used to remove it. Perhaps a new kind of garbage collector could be devised to handle the problem. The relational data base offers a third method of considering the knowledge system. It is similar to the concept of graph analysis but has the advantage of allowing for easy modification of the information.

There are three ways of approaching the problem. First, leave the form of the input statements as is, but supplement with derived statement forms. In order to use this type, extensive dictionaries would be required. These structures would have to be edited from time to time just as the graph is edited, and the execution rate of such a system might be low. The second method is to analyze inputs and extend the graph of knowledge to connect with the new knowledge. Conjecture and time would have to be added to the graph to allow for resolution of contradiction and falsities. A garbage collector that removes hidden or covered

data from the graph will be required. Parts of the graph can be embedded in programs which will allow for very rapid evaluation of the effect of the knowledge. The third method would be to embody the graph within a relative data base. This technique would be required for memory efficiency and when only moderate speeds are necessary. It is possible that all three of these methods could be used to address the total problem.

The problem of derived information is serious. If the system does not derive some information, then the response time of the system will be slow. However, if all derived information is stored, there is no hope for implementation. In mathematics, a similar problem arose. In this case, many derived expressions are generated and, when one is found that is of small form, it is stored and the longer derived expressions are removed from further consideration. Or if a name can be used to unify large numbers of similar expressions, then a new form is generated with a new name. The name is defined. Then these new forms are used, when possible, rather than deriving the relations from the original expressions. Unfortunately the mathematical model is not wholly applicable to this case. There are no significant event-related aspects to the mathematical model as in the knowledge problem. Forms that are derived at one time do not have to be removed later because of the changing situation. Here, as in the management problem for the communications, events occur that require the removal of previous forms; ie, the appropriate knowledge changes. Since the knowledge base can change, much of the derived knowledge will have to be discarded from time to time. Therefore, it does not pay to carry the derived knowledge too far. Some balance between derived knowledge and requested inclusion and questions must be achieved. Further techniques are required to determine which piece of derived knowledge has changed.

The simplest type of knowledge change is one in which the new data directly contradict one and only one item of the data base. However, in dealing with the organization of the data base and the contents transmitted over the orderwire, occasions arise in which the new knowledge is not compatible with the present data base, but no single data item is challenged directly. Only after analysis can it be shown that certain knowledge is suspect or erroneous.

Extension of Knowledge. The enlargement or extension of a body of knowledge presupposes existing knowledge with a language capable of describing it. Enlarging the existing body of knowledge can be thought of as updating a file of specifications. The extension of knowledge, however, may be thought of as enlargement of the total communication system; ie, increasing the number of nodes in the system. To incorporate other nodes into the system requires the consideration of several points. Can the new node communicate in an existing language? If not, a specialized language must be created before any communication can occur. Having satisfied the language compatibility requirement, a link to the new mode must be established. Once this has been accomplished, orderwire data may be transmitted, thus extending the body of knowledge to a new node. When the orderwire data have been transmitted, this link can be made available for user traffic. This situation can occur often; for example, when a new ship joins an existing task force (a permanent node) or an aircraft comes under temporary command of a ground controller.

3.2.1.1 Equipment Knowledge. Equipment Control Procedures and Processes. Each node has control of its own equipment. It also has knowledge regarding its equipment specifications, capabilities, and the specific settings for all its own equipment necessary to accomplish communications.

Control and setup of equipment must be a stepwise procedure which is established in accordance with communications specifications as well as individual equipment specifications.

Orderwire Implementation and Equipment Functions. While it is expected that a standardized suite of equipment will be used, each node may possess functionally unique abilities which imply the use of different classes of equipment. This gives rise to the problem of where equipment knowledge should reside.

3.2.1.2 Local Knowledge. There are two aspects of the system knowledge that have importance in the system. First there are the sure knowledge of the information about self-node and the more or less sure knowledge about the communication structure of the neighboring nodes (those nodes that have a direct arc connection). The second type of knowledge is that of the not-neighbors of which the system has only partial knowledge. The local knowledge requirement is high, since there must be explicit correspondence between the orderwire languages and each arc and channel. Also, the local neighbors can be considered to have the most important knowledge as far as routing and condition are concerned. The orderwire exchange is higher between neighbors than between remote nodes. The knowledge about neighbors is the largest other node knowledge. Lesser knowledge is stored about remote nodes.

A Special Note of Knowledge About Channels: There is a subtle difference between a used channel and a constructed channel. A used channel is a constructed channel which must be assigned to a particular defined communication function. A constructed channel is one that singly exists and can be assigned. A channel has end-to-end communication capability. A path is a set of connections that can support a constructed channel but in which there may or may not be a channel. A constructed channel can be manipulated in such a way as to divide or recombine. Orderwire will use a constructed channel because it is already present. From the moment of creation it has an orderwire language or operation system attached. Refer to figure 3-3.

3.2.1.3 Global Knowledge. Knowledge about the entire system is required but not in the detail required for the neighbors. More knowledge is required about the groups to which this node belongs. These groups could be command control groups or sensor groups. Since the entire system is more or less changeable, changes to the total system can be accomplished. This constitutes global knowledge that the node must remember.

### 3.2.2 CONNECTIVITY RULES

Consider the connectivity of a communications network in terms of the connectivity of a communications graph; a node in the graph represents a node in the network and an arc in the graph represents a communication link in the network. At any instant there exists in the real world a true connective of the graph. That connectivity must be represented by the collection of all the knowledge graphs of each of the communication nodes. Connectivity is useful only if it is represented in the graph knowledge.

Since the communication is by pairwise connections, the knowledge must be doubly stored in the connective graph. This knowledge must be distinguished from derived knowledge. We will call this knowledge about the true connective as seen by the ends of each arc the primary knowledge of the connectivity. The secondary knowledge of the connectivity is that which is communicated to other nodes to ensure that their knowledge is complete enough to carry out their responsibilities.

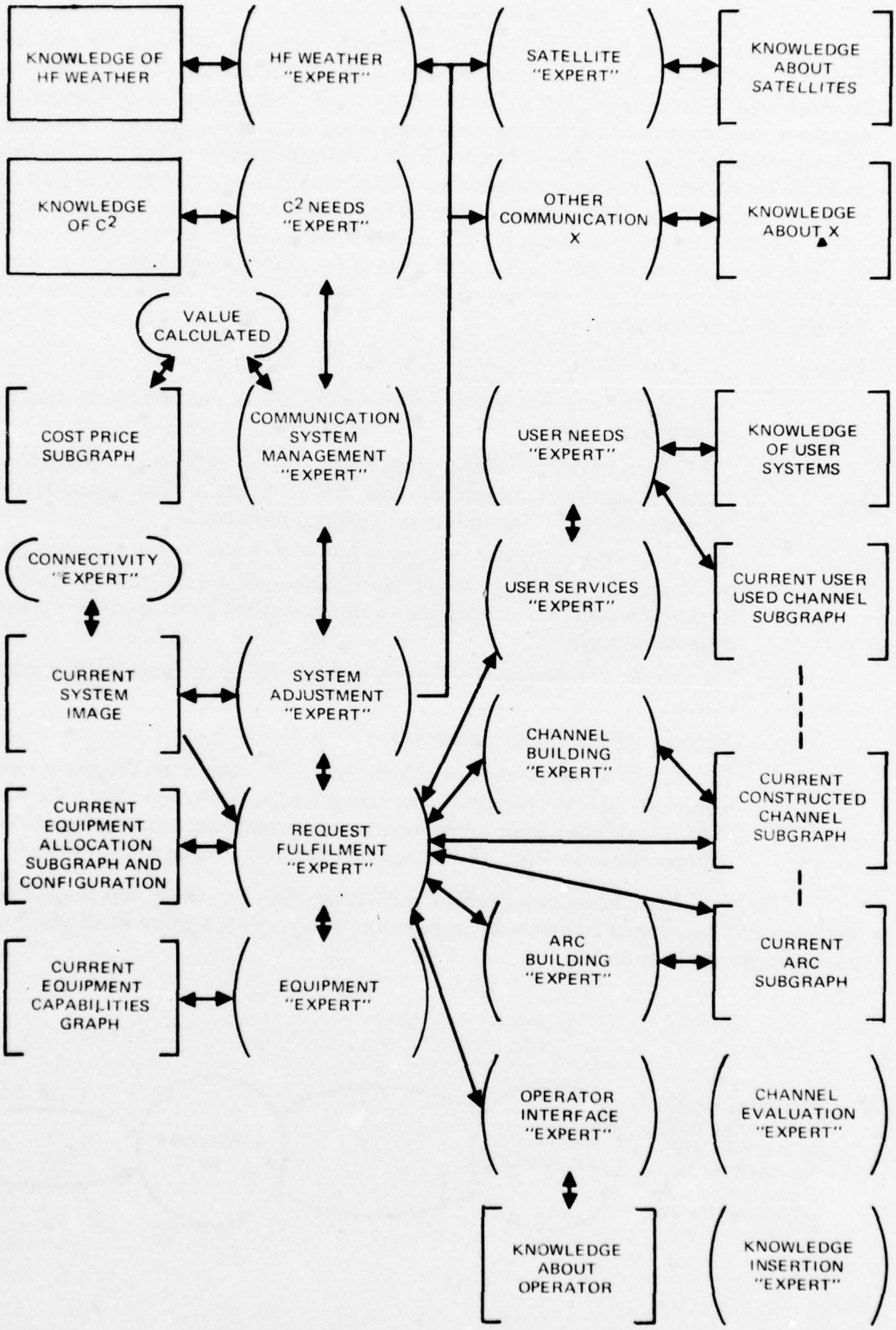



Figure 3-3. Knowledge and "expert" interaction.

Each node must have either explicit or implicit knowledge of the entire connectivity. Since a connectivity of the entire graph must exist, the representation of that knowledge must be considered. The knowledge of the connectivity of the node with its nearest communication neighbors must be specific. The knowledge at each node should be considered self-centered; ie, the knowledge is specific about the connections to the neighbors and general for distant nodes of the system. Specific knowledge is straightforward: one arc of the graph for each arc of the connectivity. The concept of general knowledge must be more explicitly explored.

The concept of communication graphing is that a group is a collection of nodes and the collection has a name. The name can then stand for all the nodes in the group. The membership of each group must be general knowledge for those nodes that will use the name for building their own connectivity.

Rules:

a. The knowledge of connectivity of the entire system must be represented in the graph at each node.

b. The vertices of the knowledge graph must be either explicit or a group name. The total of all names of the explicit nodes and of the group members must contain all the names of every node of the communication system.

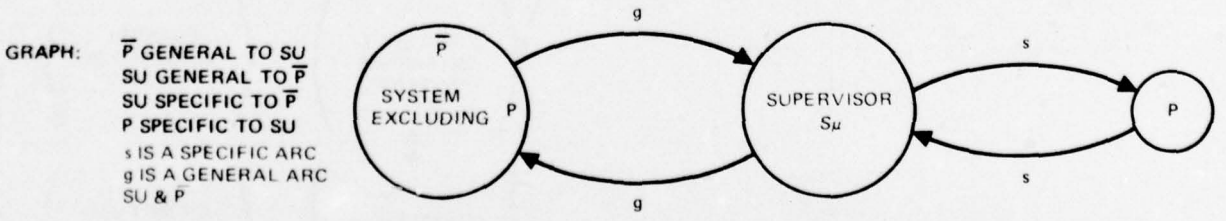
c. Knowledge of membership of a group is not necessary, but if a group is given without stating membership, some higher-level node of the system must have a knowledge graph containing that group to ensure that the lower level has a properly connected graph.

d. The specific knowledge of connectivity with at least one member of a group is required.

e. A group must be totally connected.

f. The representation of the connectivity at a node cannot use the group names of which this node is a member. The reason for this is obvious. Since the node must have explicit knowledge of the neighbors, it cannot use a mechanism that represents only general knowledge of the neighbors which the group name provides.

The very simple nodes (the pendant nodes) have their knowledge represented as a single connection to their communication superior. It represents a group of all other nodes excluding the pendant node.



This graph represents the knowledge at a pendant node. It is not necessary that a formal representation of the graph be stored at the pendant node except for the representation of itself and supervisor SU. The form of the graph is identical for every pendant node. If the pendant node is not constructed as a pendant node (a degraded higher-level unit), then a more formal representation is required. Since membership relationships form a Boolean lattice, the concept of "not set" node has meaning; ie, all the nodes excluding the designated set or node.

This graph has all the properties that have been described. First, it contains every node by virtue of the fact that the group $\overline{P}$ and $\overline{P}$ by definition contain all nodes. Second, P is not a member of $\overline{P}$. Third, SU is a member of $\overline{P}$. Note that the definition of specific arcs as those arcs whose details are known cannot be made between a node and a group (between $\overline{P}$ and $\overline{P}$). An intermediary is required that is a member of the group. Note also that general arcs can exist between two groups. Every arc to or from a node in the knowledge base that is itself in reality must be specific. Arcs to or from nodes that are not themselves may be specific or general.

Very little utility can be attached to a group name unless it is totally connected. If it is totally connected, then connecting to a member of a group ensures that there is a connectivity path to each of its members.

### 3.2.3 THE REPRESENTATION OF KNOWLEDGE

These issues are discussed in D3.2.4:

Frames of reference

Time and space qualifiers

Class qualifiers

- $(\exists\ y)$ "There exists"
- $(\vee\ y)$ "There is a unique one that"
- $(\llcorner\ y)$ "All"
- $(\urcorner\ y)$ "It is not true that"
- Facts

Qualifiers for probabilistic functions

- Conditions probability
- Estimated probabilities
- Measured probabilities

Hypothesis generation

- System conjectures
- Operator conjectures

Searches

Goals

Conflicts

Falsehood and irrelevants

Problems of insertion of new knowledge

Problems of deletion of falsehoods

Problems of removal of irrelevancies

Problems of not enough knowledge

Problems of too much knowledge

Problems of verification of knowledge

Problems of communicating knowledge

### 3.2.4 THE REPRESENTATION OF KNOWLEDGE

Frames of Reference. In a communication system such as has been studied in this document, it is necessary to consider the knowledge carefully in some frame of reference. For instance, the knowledge concerning syntax on a particular channel is really only the syntax for that particular channel. It is not necessarily the same syntax for any other channel. If an orderwire message changes the syntax on a channel, then it will be changed for only that channel. The frame of reference for that channel has been detailed to be more explicit. This is particularly true of abbreviation in the orderwire language. In some other instance of the language for some other channel an entirely different set of abbreviations may exist. Further, the frame of reference changes with the medium that is being utilized. Terminology used to control an hf arc is different from that used to control a satellite arc.

Time and Space qualifiers. When an event occurs in the system that is even remotely associated with a node, an arc, or a channel, there is generally specific information about that event. This knowledge has qualifiers associated with it that indicate time and location of the event that gives rise to the knowledge. An action audit may require this information. But more important, knowledge can be dated and located. Knowledge represents stable conditions of the world which has truth in the frame of reference that is valid only for a limited portion of the earth. Valid weather information at one node may be worthless for a node a long distance away but good for a close node.

The Concept of Time. Along with every new piece of knowledge are various aspects of knowledge. The following times are associated with a piece of knowledge:

Time of arrival at the system

Time that the knowledge was originally created

Time of deletion from the system

If the knowledge is made up of parts and if the parts are altered, then these in turn have associated times. There also must be a way of relating times to the associated knowledge. Identification of knowledge by assigning a name provides a reference and a means of talking about it. Without a name, a piece of knowledge cannot be talked about and time or other knowledge cannot be related to it. From all this, it becomes clear that time must be related to a name. Also, a time line for a name is a useful idea. This line has a time for when the entity comes into existence, when new branches or new relationships are created, and when the entity is destroyed.

Some concepts or names do not have a natural beginning or end. For these cases there must be a provision to name not only the time, but also the type of time. If the entity has no beginning or no end, or neither, an explicit time is not required.

Now in addition to the intrinsic relationships of time to name, there are also times relating to the arrival of information. Each piece of information arriving that tells some-

thing about an entity must of necessity have a time associated with its creation, so that coordination of the data can be made appropriately. The transmittal of information from one source takes less time than from another source, and, as a consequence, later changes to the status of an entity may arrive before earlier ones. The system would not be able to determine relationships if the source of the change were not indicated in some manner by the time itself. Time must play an important part in the building of a knowledge base and be an intrinsic part of the communication orderwire.

Remarks pertaining to time can also apply to equipment. When a piece of equipment is introduced into the system, it begins its existence and must be given a unique name. This name can be used as a basis for intrinsic time. Similarly, when an action generates a new arc, the arc has existence by virtue of existence of a name and time of creation.

When an entity is introduced into the system, the name given to it must be unique to prevent confusion and ambiguity with a previous entity. In addition to being unique, generic relationships should be evident by characteristics in the name. Thus, the guiding communications on an item will always describe it either implicitly or explicitly.

Channels also require names like the arcs that contain them. Times of creation, times of annihilation of an arc, and times of annihilation of a channel within an arc all require names. If a channel is going through an arc and the arc is annihilated, then the channel must have been annihilated also, but if the system was able to redirect the channel before the arc was annihilated, the channel and its name would continue to exist after the arc disappeared. Thus, the continuity of the name is preserved along with the channel itself, and the new modifications can be talked about by use of the assigned name.

It is foreseen that, for each different item, arc, channel, or equipment, a time line is required to record the history of the item. If this history is recorded, the generation and annihilation of relationships between nodes or entities of the knowledge base can be traced.

The understanding is that the data are related by virtue of belonging to the same orderwire sentence and the same orderwire interpretation environment. The orderwire interpretation environment is another way to talk about the state of the system regarding the orderwire channel.

For the purpose of discussion, the pairs of an orderwire channel will be assumed to be ordered in accord with increasing time. When writing the sequence of data time pairs on paper, pairs on the right should be considered to be later in arrival than pairs on the left. Further, if the time of arrival is not a fixed increment, it is assumed that preparatory processing has ensured that all pairs are present. In general, it is not necessary to carry along the time in all orderwire words; but there are important cases in which time is the key to orderwire interpretation. These are not the general cases and should be considered to be orderwire constructs which are translated to the normal syntax after arrival. Even in normal orderwire, time is still an important aspect of the orderwire control. However, time should not be considered to be an important part of the syntactic translation. For example, changing the state of the receiving machine or syntactical translation procedure will require that orderwire arriving late does not cause an unintended alteration. We could argue that this is a semantic problem and not a syntactic problem. The general syntactic translation is that part of the procedure that takes the sequence of words and constructs a representation in the node processor of the action that is to be performed as described by the sequence of words. The constructed representation is called the semantic representation. The realization of the semantic representation is the exercise of control by the machine to carry out the actions that were described by the orderwire statement. To summarize, the sequence of symbols or words is received over the orderwire channel. This sequence of words does

not in itself give rise to the actions desired. This sequence of words is not, in general, a good representation of the actions to be performed. It lacks many important parameters that are required for the execution of the action. Translation to a good representation is required and in many cases binding of names. The resulting structure of the translation is in general a tree or graph and not a sequence of words. This is the semantic tree, also called the semantic representation. The semantic tree is still nothing more than a state of the memory of the host processor. To perform the actions desired by the orderwire, a semantic interpreter is required that causes switches to be changed in the communication equipment.

Class Qualifiers. When dealing with knowledge, we have to deal with rules and collections of relationships. One aspect of such collections is the scope of the rule or statement. The typical qualifiers are:

($\exists$y) "There exists a __y__ that"

($\forall$y) "The unique __y__ that"

(<y) "Every __y__ is"

($\neg$ y) "It is not true that __y__ "

Fact: "A is/____ related to B"

The problem with these qualifiers is that the frame of reference is not always clear. Also, a true statement may not be known to be true. More important, the number of statements that may be true may be so great that storage in the system is not possible unless explicitly entered manually. Some statements can be so close to others in their scope of application that they are not effectively different, but their forms are distinguishable. The system has trouble removing the redundant one.

Qualifiers for Probabilistic Functions. A large number of decisions are based on a best estimate of what could be. The measure of the value or criterion is the probability that such and such is true. The question then arises as how to handle the probabilities and conditional probabilities that one encounters in the knowledge representation. Static calculation models do not, in general, work for uncertain data. Bayesian techniques are generally preferred since, as new data become available, the assurances of the derived information can be increased or decreased. To do this, however, some inference trail must exist to determine the descendants of a probability change. In addition, the initial values for the distribution are generally not too well formulated. One of the techniques is the greatest uncertainty and another is that of operator evaluation.

Hypothesis Generation. When the system is required to accomplish some activity, it may have to act on a model of the world that may be inconsistent or uncertain. A hypothesis of what is probably correct must be generated and then tested. In some difficult situations the system may decide that the operator may be a good source for a hypothesis. In this situation the operator is asked a question and the operator gives advice. Also, the system may have generated several hypotheses as to the world knowledge or to how the action should be accomplished. If time permits, the system may ask the operator to reject some of the hypotheses on the basis of the operator's judgment. The question of what kind and how many hypotheses should be generated arises along with the question of whether the operator should be consulted. Some actions may be expensive to test. It may be advantageous for the operator to suggest or reject formulated hypotheses. Some hypotheses are generated with incorrect frames of reference, and the operator is more likely than the system to spot these inconsistencies.

Searches. When dealing with a knowledge base with derived and historical knowledge, it is sometimes desirable to search the knowledge for similar situations. For example, if the system "recognizes" a similarity between the current situation and a previous situation, it may examine the previous situation to see whether advice was given and adjust the hypothesis building strategy accordingly. Further, the operator may ask a question of the system that requires the system to search its knowledge for the appropriate information.

Goals. Many of the operations necessary for the execution of the automated orderwire require that the request be set up as goals. In this manner present techniques of AI can be used to formulate strategies for satisfying the request that arrived. Also, it provides insight into changes that would be required to be made to accomplish the desired goal.

Conflicts. There is a normal set of conflicts and contentions that arise in a busy system. Some of these can be resolved by priority. However, many conflicts will have the same priority level. The resolution of conflict has to be done by other means, but resolution of the conflict will have to be made. One technique exists in the command control area; namely, the rank of the officer, or date of rank if two have the same rank, resolves the conflict. Whether extensions of this method or similar resolution by organization can be used is unclear.

Falsehood and Irrelevants. It is impossible to keep contradictory and irrelevant information from being entered into the system. Very little of the data will be intentionally false, but, by the nature of sensors, the improper scoping of frames of reference, inexact advice of users and operators, and errors in transmission, some data will be incorrect. Other data will be correct but completely irrelevant. A question arises as to how much screening will be necessary. How are the effects of false information kept to a minimum?

Problems with Insertion of New Knowledge. When new knowledge is introduced, either by the orderwire communications received, by operator entry, or by extractions from measuring or sensing equipment, this knowledge must be integrated with the knowledge already present to make a meaningful addition to the knowledge base. Unfortunately, this is not a simple task. The knowledge may not have "hooks" which allow for immediate connections to knowledge already present. The system may have to set up goals that permit a building of the existing knowledge system to form possible relationships between old knowledge and new knowledge. This may fail. The node may have to generate requests to other nodes to clarify the meaning of the new knowledge. The new knowledge may be irrelevant at this point, or irrelevant altogether. The "hooks" are names, parts of names, situations bearing upon the entry of new knowledge, structures of names, and pure structures. Also, when new knowledge is added to the system, that knowledge may be false, may become obsolete, or may become incorrect by virtue of changing circumstances. When this happens, the knowledge must be marked and possibly removed from the system. In addition, it is not enough to remove the knowledge; all its descendants must be examined to ensure that they are not also incorrect. If action was taken on the basis of incorrect knowledge, the possibility arises that counteraction is required.

Removal of False Information and Inconsistencies. When a knowledge base is constructed, it is possible that several parts of the knowledge base form a contradiction. The system may detect this contradiction or it may not. If it does not detect the contradiction, it may cause the system to act erroneously. In fact, if care is not taken, any contradiction in the system can cause the system to go wild. The use of probabilities of truth can help but cannot solve the problem. Most systems that handle knowledge are based on formal logic, and formal logic is generally insensitive to rationality. If the pieces are rational, the results are rational; but if any part of the knowledge is in error, the result can be in error also. There is no mechanism to catch all irrational and irrelevant reasoning. Only

where there is also a feature that controls the logic to ensure that the elements of the implication are relevant to the reasoning can there be some control over the generation of irrational results not associated with the erroneous knowledge. Once falsehood is known to exist in the system, the problem becomes how to remove the falsehood, and this is not a trivial problem. If the falsehood is detected by an inconsistency, what is false? The inconsistency can be formed in a very intricate manner. A false statement is not easily detected. Once a statement is determined to be false, how do we remove it? To introduce a new statement that is a contradiction to the given statement does not help. The deletion of the statement must be accompanied by an examination of the descendants of the statement. Careful audit trails are needed to do this. In addition, results that were derived may have been correct. New data will support the statement. When and how this new support for the results should be introduced into the system is not clear. Obviously, removal of a good statement should not take place because its ancestry was a bad statement of knowledge. If a derived statement of knowledge is known to be false, that is sufficient to conclude that one of the ancestors of the knowledge was false; but it does not tell which one.

Problems of the Removal of Irrelevant Knowledge. The knowledge base is always acquiring new statements for the system, but the memory and processing power of the system are finite. It is desired to remove any information from the system that is not useful in a short time. Removal need not be destructive, because the information can be placed somewhere else in the communication system. It will not be immediately available, however. Enough knowledge must remain to carry out the function of the node. Identification of irrelevant information is almost as important as the removal. Once it is identified, one still has the problem of removing it. It is not desirable to have to contradict knowledge to remove it; what is needed is a forgetting function that depends on size, permanency, and date of the knowledge. Also, simple removal of the original statement is not always appropriate, since a statement may carry more than one relational structure, only one of which is still being used. A garbage collection function is required.

Problems of Not Enough Knowledge. In present systems in which there is not enough knowledge, a human is interfaced with the system to supply the judgment necessary. In some systems the designers know enough about the system to build decision making into it to ensure that an adequate decision is made. In the last analysis both these techniques are employed in this concept of a communication system. The difference between the subject system and present systems is the nature of the knowledge that is missing. The knowledge in this system has aspects which are uncertain and for which no well codified rules are known. (In a system of this complexity it is possible that the codification of the rules can never be totally accomplished.) But once a situation arises that needs human intervention, it can be studied, and rules can be generated to take care of the situation. It is necessary to make allowances for the detection of such incompleteness of knowledge and plan for the introduction of rules that can take care of the situation.

Problem of Too Much Knowledge. One way of looking at this problem is to view it as the presence of too much irrelevant knowledge. This concept has been addressed above. The alternative is to recognize that data collection can generate as much data as we desire but the data are useless unless they can be processed and digested. The most powerful technique of digesting lots of knowledge is to digest small portions only. In a communication system it may be desired to make the selection at the source rather than try to move all the knowledge to the node at which the decisions are being made.

Problems of Verification of Knowledge. In composing an action, knowledge is used to formulate the appropriate course of action. It is one thing to keep various possible alternate statements in a knowledge base and another to execute a nontrivial action as a result of that knowledge. To carry out all the different alternatives is generally neither desirable nor possible. As a result, certain statements of the knowledge base take on a value depending on the validity of the statement. A methodology is needed to determine whether some other action can take place that verifies the critical statements.

Problems of Communicating Knowledge. Generally, when knowledge is available at one node and needed at another node, some method of communicating is needed to cause the appropriate decisions. There are several ways that this can be accomplished. First, the knowledgeable node sends the knowledge to the node requiring the knowledge. This in turn presents problems in that the node requiring the knowledge may not have the appropriate frame of reference to receive and interpret the knowledge statements properly. Second, the knowledgeable node may not be able to extract the key features of the knowledge that need to be transmitted. Third, the node requiring the knowledge may send the problem to the knowledgeable node and ask the knowledgeable node to solve it. But this has the same problems in that the node requiring the knowledge has other knowledge that bears on the problem. How to extract this knowledge from the node requiring the knowledge to transmit to the knowledgeable node is the same problem as discussed for the first case. Finally, there are no compromising and anticipatory methods that try to coordinate the knowledge before the appropriate decision is made.

## 3.3 PROPERTIES OF THE ORDERWIRE LANGUAGE

### 3.3.1 INTRODUCTION

3.3.1.1 Overview. The general syntactic translation is that part of the procedure that takes a sequence of words in the orderwire language and constructs a representation in the node processor of the action to be performed. The constructed representation is called the semantic representation. The realization of the semantic representation is the exercise of control by the machine to carry out the actions that were described by the orderwire statement. To summarize, there is the sequence of symbols or words received over the orderwire channel. This sequence in itself does not give rise to the actions desired. This sequence of words is not, in general, a good enough representation of the actions to be performed. It lacks many important parameters that are required for the execution of the action. These parameters are contained within the communication system's body of knowledge and an extraction and binding process is required. This process is performed by the semantic interpreter. The sequence is depicted in figure 3-4.

Elemental Actions of the Orderwire. The orderwire activities are essentially those of transmitting symbolic structures. All lead to a sequence of executions, and to a consideration of the requests and translation into an orderwire language for reception at the other end. The elementary activities of the orderwire, therefore, can be considered and categorized in the following types of activities.

a. Translate the special condensed languages into a normal language. This activity consists of converting bits as they arrive to the normal language of the orderwire.

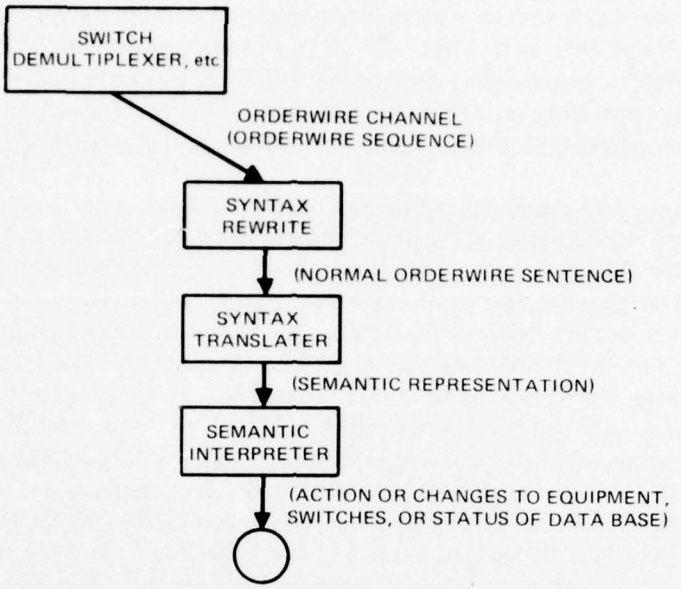



Figure 3-4. Translation steps in interpretation of orderwire language.

b. Convert the normal orderwire to a semantic control language.

c. Generate from the semantic control language the necessary execution sequence whereby the activities to be requested, including execution, are carried out.

d. Since the activities of the system will be implemented in the concept of semantic language, a request for activities of any part of the system for the purpose of translating data for another node would convert from the structure of the semantic language into the normal language of the orderwire.

e. In individual cases in which the normal language is not being utilized, such as in networking, the orderwire structure must translate the normal language into the special language for a particular channel. Another elementary activity of the orderwire system would be to sequence the inputs into a channel and separate the outputs of the channel into the various subchannels and activities required.

3.3.1.2 Usual Syntax of the Orderwire Language. Syntax is the expression of the structural relationships between the pieces of a sequence of symbols (starting in the case of the orderwire problem with the sequence of bits coming over the orderwire channel) to some sequence of actions called the semantics of the language. In part, this language is developed from human usage of communication channels dealing with symbolic forms.

The advantage of continuing to use the same symbolic alphabet as that now used is primarily in the maintenance and development of the orderwire network, language, and processes. It is for this reason that both the normal and expanded orderwire languages use ASCII as the primary alphabet, with the symbols interpreted in the usual manner where appropriate. It is conceivable that a better alphabet and encoding could improve the system from a theoretical point of view. However, the loss of the ability to communicate with the technicians and engineers that maintain and develop the system would be too great to even consider such a language at this time.

A formal syntax is absolutely required for an automated orderwire. An informal syntax such as the English language has too many ambiguities to be practical for an automated system. A finite amount of ambiguity may be tolerated or even desirable if the ambiguity can be removed by frames of references. Machines are not capable of removing all the infinite forms of ambiguity in the informal languages.

The general overall structure of the normal orderwire language is as follows:

The language is layered: the results of syntactical analysis of one subcomponent of the language are passed on to become the input of the next component of syntax analysis. The most notable layer is that in which a translation of the bit stream of the orderwire into a character string of ASCII characters takes place. In systems that use Baudot, first the bits from the bit stream of the orderwire are translated into Baudot and then the output of this layer is translated into ASCII characters. The next layer of syntax seeks information on the frame switching syntax. The layers beyond that consist of a sublanguage, followed by a layer of lexical analysis. The next layer of the language syntax controls numerical analysis. There is at least one higher layer that has control; however, there may be higher layers yet. It is not important to consider whatever higher layers exist at this point.

The principal sublanguages of the normal orderwire language are the meta language, the knowledge languages, the change languages, the semantic languages, the data languages, and the computer languages. These classifications break down into further sublanguages. The structure can be seen in the following outline of languages:

Knowledge
- Executive
- Knowledge update
- Formulate frame of reference
- Naming
- Screening

Data
- Raw data
- Graph compression

Change
- Remote control
- Request action
- Request information

See figure 3-5. The meta language is the language in which new syntax can be described. The knowledge languages are those that allow for building and changing the data base,

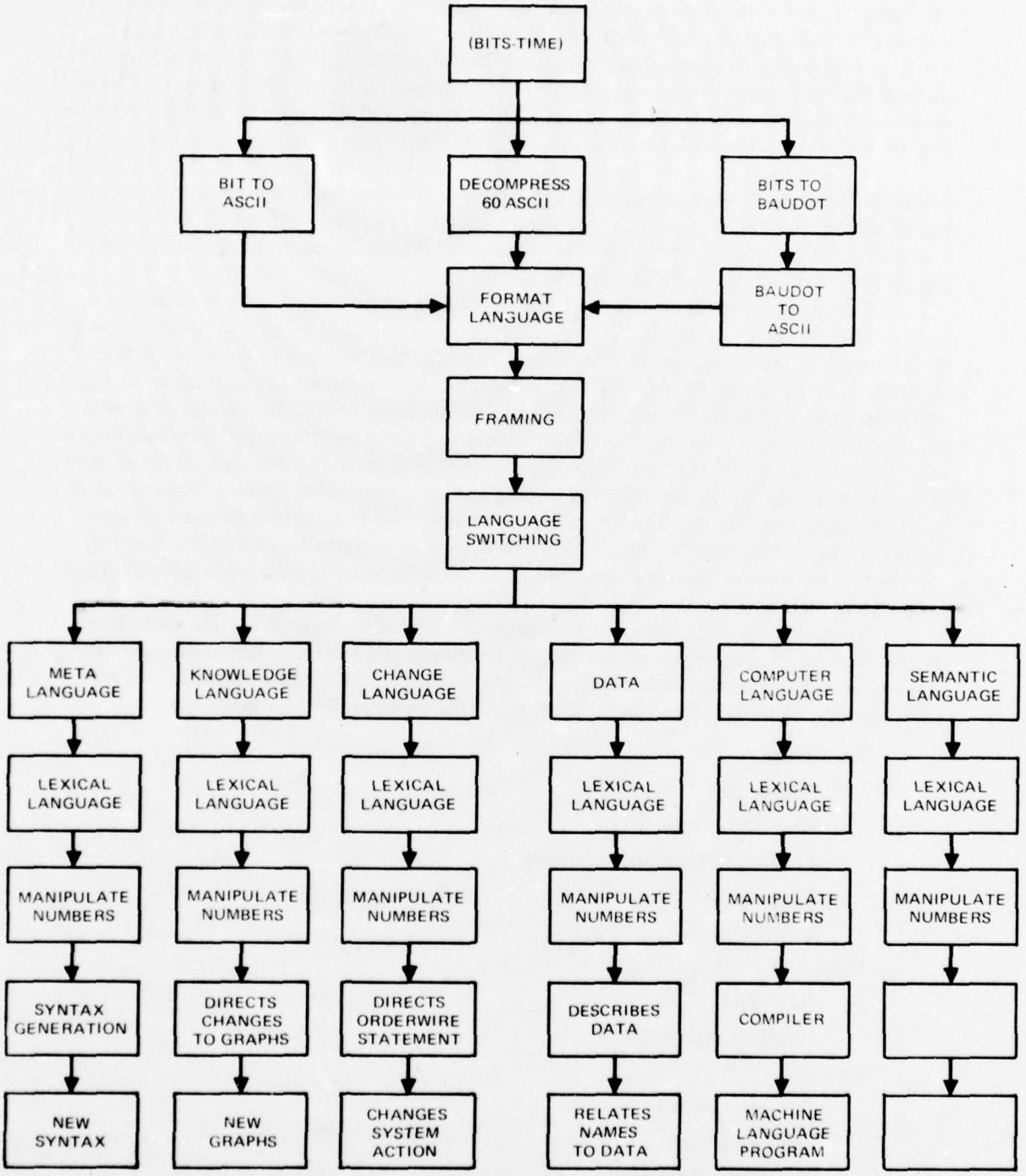



Figure 3-5. Automated orderwire language structure.

describing the environment, or describing the desired action. The change language is the language that invokes action. The semantic languages are special languages that can do particular functions. The computer languages are those that can be compiled to generate machine code whereby a new semantic process can be introduced.

### 3.3.2 BIT LANGUAGE

The ordering input is a sequence of bits. However, for the consideration of processing, it may be required to consider the input as a vector consisting of bit value, time, and environmental description.

Only the bit value is received. The time will normally be available. Messages that receive meaning by virtue of the time of arrival are those that must have clock to determine meaning and require this additional factor. The environmental picture may or may not be useful. The transmission of information has only aspect; it is the conveyance of surprise. If there are environmental factors that affect the semantics and syntax of the data, the factors must have the following properties:

They must be known by all parties of the communication.

They must have a change character whose changes take place on a time scale of approximately the time of bit arrival.

The clock has this characteristic, but it is hard to conceive of any aspect (other than clock-related things) that has this characteristic. Something that may be known to both parties that varies slowly would not provide the necessary semantic interpretation, since the change would have to be made quickly and the surprise of the selection is more than one bit's worth.

The conclusion is that the syntax-semantic interpretation must be restricted to the bit values, time, and functions of time.

If we examine communication in more detail, it is clear that communication is derived from observation of the environment. For the purposes of orderwire, it is assumed that the derivation has already taken place and that we are dealing with a coherent package of bits. Some communication can be thought of as arriving simultaneously, such as that derived from frequency division multiplexing. In this case, we are dealing with a word rather than a bit.

In summary, the input to the syntax analyzer of an orderwire channel is represented by a sequence of pairs, the first member of which is data received in the form of a word of one or more bits and the second member the time of discrimination (time of arrival of that word).

(first word, time)

(second word, time 2)

(third word, time 3) . . .

(nth word, time n)

It is possible to conceive of parts of the data arriving by different circuits. If this is the case, we still have to coordinate the data into a single stream of pairs. If the two sources are not coordinated and have individual meaning, then we are not dealing with a single orderwire but with two orderwires. For the purposes of design, the possibilities of

interweaving of bits from two or more sources does not alter the discussion of the analysis of orderwire because the coherent stream must be constructed prior to the interpretation.

It is not enough that there is a collection of pairs. It would appear at first sight that because we have the time explicitly, that is all that is necessary for proper interpretation. This is true in one sense, but not in another.

### 3.3.3 CHARACTER LANGUAGE

USASCII, the USA Standard Code for Information Interchange, is contained in MIL-STD-188C. The following has been abstracted to define the character language.

1. Control Characters:

   NUL: The all-zeros character.

   SOH (Start of Heading): A communication control character used at the beginning of a sequence of characters which constitute a machine-sensible address or routing information. Such a sequence is referred to as the "heading." An STX character has the effect of terminating a heading.

   STX (Start of Text): A communication control character which precedes a sequence of characters that is to be treated as an entity and entirely transmitted through to the ultimate destination. Such a sequence is referred to as "text." STX may be used to terminate a sequence of characters started by SOH.

   ETX (End of Text): A communication control character used to terminate a sequence of characters started with STX and transmitted as an entity.

   EOT (End of Transmission): A communication control character used to indicate the conclusion of a transmission, which may have contained one or more texts and any associated headings.

   ENQ (Enquiry): A communication control character used in data communication systems as a request for a response from a remote station. It may be used as "Who Are You" (WRU) to obtain identification, or may be used to obtain station status, or both.

   ACK (Acknowledge): A communication control character transmitted by a receiver as an affirmative response to a sender.

   BEL: A character for use when there is a need to call for human attention. It may control alarm or attention devices.

   BS (Backspace): A format effector which controls the movement of the printing position one printing space backward on the same printing line. (Applicable to display devices.)

   HT (Horizontal Tabulation): A format effector which controls the movement of the printing position to the next in a series of predetermined positions along the printing line. (Applicable also to display devices and the skip function on punched cards.)

LF (Line Feed): A format effector which controls the movement of the printing position to the next printing line. (Applicable also to display devices.) Where appropriate, this character may have the meaning "New Line" (NL), a format effector which controls the movement of the printing point to the first printing position on the next printing line. Use of this convention requires agreement between sender and recipient of data. (So long as the current operating procedure of ending (or beginning) a line with CR-CR-LF is observed, no operational problem arises from interconnecting "NL" and "Non-NL" machines.)

VT (Vertical Tabulation): A format effector which controls the movement of the printing position to the next in a series of predetermined printing lines. (Applicable also to display devices.)

FF (Form Feed): A format effector which controls the movement of the printing position to the first predetermined printing line on the next form or page. (Applicable also to display devices.)

CR (Carriage Return): A format effector which controls the movement of the printing position to the first printing position on the same printing line. (Applicable also to display devices.)

SO (Shift Out): A control character indicating that the code combinations which follow shall be interpreted as outside the character set of the standard code table until a Shift In character(s) is reached.

SI (Shift In): A control character indicating that the code combinations which follow shall be interpreted according to the standard code table.

DLE (Data Link Escape): A communication control character which will change the meaning of a limited number of contiguously following characters. It is used exclusively to provide supplementary controls in data communication networks. DLE is usually terminated by a Shift In character(s).

DC1, DC2, DC3, DC4 (Device Controls): Characters for the control of ancillary devices associated with data processing or telecommunication systems, more especially switching devices "on" or "off." (If a single "stop" control is required to interrupt or turn off ancillary devices, DC4 is the preferred assignment.)

NAK (Negative Acknowledge): A communication control character transmitted by a receiver as a negative response to the sender.

SYN (Synchronous Idle): A communication control character used by a synchronous transmission system in the absence of any other character to provide a signal from which synchronism may be achieved or retained.

ETB (End of Transmission Block): A communication control character used to indicate the end of a block of data for communication purposes. ETB is used for blocking data where the block structure is not necessarily related to the processing format.

CAN (Cancel): A control character used to indicate that the data with which it is sent are in error or are to be disregarded.

EM (End of Medium): A control character associated with the sent data which may be used to identify the physical end of the medium, or the end of the used, or wanted, portion of information recorded on a medium. (The position of the character does not necessarily correspond to the physical end of the medium.)

SUB (Substitute): A character that may be substituted for a character which is determined to be invalid or in error.

ESC (Escape): A control character intended to provide code extension (supplementary characters) in general information interchange. The Escape character itself is a prefix affecting the interpretation of a limited number of contiguously following characters. ESC is usually terminated by a Shift In character(s).

FS (File Separator), GS (Group Separator), RS (Record Separator), and US (Unit Separator): These information separators may be used within data in optional fashion, except that their hierarchical relationship shall be: FS is the most inclusive, then GS, then RS, and US is least inclusive. (The content and length of a File, Group, Record, or Unit are not specified.)

DEL (Delete): This character is used primarily to "erase" or "obliterate" erroneous or unwanted characters in perforated tape. (In the strict sense DEL is not a control character.)

Note: SO, ESC, and DLE are all characters which can be used, at the discretion of the designer, to indicate the beginning of a sequence of digits having special significance.

2. In serial by bit transmission, the coded character will be transmitted low order first; ie, the bits will appear on-line in the order b1, b2, b3, b4, b5, b6, b7, b8 (parity).

3. The following table shows mnemonic abbreviations and printing symbols used in ASCII.

| COLUMN | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|---|---|---|
| b7<br>b6<br>b5 | 0<br>0<br>0 | 0<br>0<br>1 | 0<br>1<br>0 | 0<br>1<br>1 | 1<br>0<br>0 | 1<br>0<br>1 | 1<br>1<br>0 | 1<br>1<br>1 | ROW |
| b4b3b2b1 | ← NON-PRINTING-14 → | | ← 96-SYMBOL PRINTING SUBSET → | | | | | | |
| 0000 | NUL | DLE | SP | 0 | @ | P | ` | p | 0 |
| 0001 | SOH | DC1 | ! | 1 | A | Q | a | q | 1 |
| 0010 | STX | DC2 | " | 2 | B | R | b | r | 2 |
| 0011 | ETX | DC3 | # | 3 | C | S | c | s | 3 |
| 0100 | EOT | DC4 | $ | 4 | D | T | d | t | 4 |
| 0101 | ENQ | NAK | % | 5 | E | U | e | u | 5 |
| 0110 | ACK | SYN | & | 6 | F | V | f | v | 6 |
| 0111 | BEL | ETB | (APOS) | 7 | G | W | g | w | 7 |
| 1000 | BS | CAN | ( | 8 | H | X | h | x | 8 |
| 1001 | HT | EM | ) | 9 | I | Y | i | y | 9 |
| 1010 | LF | SUB | * | : | J | Z | j | z | 10 |
| 1011 | VT | ESC | + | ; | K | [ | k | { | 11 |
| 1100 | FF | FS | , | < | L | \ | l | \| | 12 |
| 1101 | CR | GS | - | = | M | ] | m | } | 13 |
| 1110 | SO | RS | . | > | N | ^ | n | ~ | 14 |
| 1111 | SI | US | / | ? | O | _ | o | DEL | 15 |

Figure 3-6. American standard code for information interchange.

### 3.3.4 PRIMITIVE LANGUAGE

3.3.4.1 Introduction. The primitive language must be locally ergodic. That is, it cannot depend on absolute time for proper interpretation. The interpretation at one moment must be the same as the interpretation at any other to within sentence boundaries. The same sentence transmitted at two different times will have the same meaning. The way to accomplish this is to require each sentence to have as a first character a start code, or a beginning of text code. For documentation purposes, it is too difficult to have binary syntax. Therefore, a character syntax will be used. Each character will be bounded with start and stop codes. ASCII will be used with special interpretation of the characters. It is clear that ASCII is not the most efficient character set for communication, but it is adequate. It has the advantage of allowing us to talk about the characters and have the reader understand. A Barker code can also be used for a start code when synchronous techniques are used to improve efficiency and reliability.

Rewrite rules could be used to reduce the ASCII characters to a set of characters for higher efficiency. The techniques for compressing character strings are known and need not be discussed further here. The efficiency improvement by using character compression is about a factor of three. The efficiency of the normal orderwire language over the primitive orderwire language is a factor between 100 and 2000. The first problem is to describe the mechanisms and requirements of the primitive language and then describe the methods of reducing to a normal orderwire language.

In describing the primitive language, it is desired to use words rather than symbols where appropriate, but the language itself would use the symbols. The words give flavor to the language.

In the following description, a lettered word will refer to a symbol; a capital lettered word will refer to the words as spelled. If the capital lettered word is underscored, it refers to a class of capital lettered words and will probably not be the actual word used in an orderwire message.

3.3.4.2 Disjoint Sublanguages

A. Executive Language

The executive language is one that controls the sublanguage on the orderwire.

B. Meta Language

The meta language proposed here is not as flexible as the Vienna Definition Language. However, the language here is similar to those that have been implemented by using Martin Kay parsing and therefore there is assurance it can be adopted. The notation is similar to that used in the REL system of Bozena Henisz Dostert and Frederick B Thompson at California Institute of Technology.

The minimum element of consideration is that of a bit at a given time. The normal alphabet is that of the ASCII character set. The bit has to be identified with some existing element of the ASCII character set. Yet they cannot be permanently bound to characters because other syntaxes cannot use these characters. The technique that will allow the binding will be called position binding. In general, the meta language syntax is

layered also. The first layer attempts to recognize one of the following list:

SYN: CHK: SET: CND: TIM: SEM: SMN: RTN:

Each of these has its own syntax. Further, each has its own name.

SYN: This is the syntax production rule.

CHK: This is the test for the features in the elements of the production rule.

SET: This is the formulation of the new features given the syntax is found.

CND: This extends the language to handle left and right contexts of the rule. In particular, it is used to handle negative contexts; that is, the absence of some characteristic.

TIM: This part of the rule controls the time of arrival part of the syntax. In character type syntax, it is normally not present. But, in bit-by-bit syntax and special language syntax, it becomes very important.

SEM: This part of the syntax statement causes immediate activity in the system. It is a trap mechanism to cause shifts in the frame of reference so that further language development will be done in the appropriate frame of reference. It is also used for statistic gathering.

SMN: This is the part of the statement that controls the generation of the appropriate action when the statement is parsed.

RTN: This is a part of the statement that allows for the return to the executive that some part of the syntax has been completely parsed.

C. Raw Data Language

Since the communication channel is for the purpose of transferring data from one node to another, a language mechanism is required to determine the starting point and sometimes the extent of the data to be communicated over the channel. This language is for the transitioning from the orderwire language to the user data. If the user data are of fixed-size blocks, this language keeps track of how much of the block has been transmitted. If there is a marker in the data to indicate end of data, this language mechanism looks for the marker. If the data are to be controlled by an alternate channel, then after the start of the data transmission on the channel, the language transfers its input to the alternate channel.

D. Graph Compression Language

Sometimes it is more expedient to transmit the part of the information for the updating of orderwire knowledge by means of a compressed format. This compression mechanism constitutes a language with its appropriate syntax and semantics.

E. Knowledge Update Language

The knowledge update language is designed to facilitate the adding of new knowledge, the removal of outdated knowledge, and the changing of knowledge in an efficient manner. It is not for the purpose of creating requests or actions, but for the purpose of improving the world model at the node receiving it. This language should use the frame of reference and naming language to determine the appropriate place at which the new knowledge is to be introduced. It should have a syntax that can also determine whether the information is a deletion, replacement, or introduction so that appropriate action can take place with reasonable efficiency. A special structure should exist in this language to identify so that duplication and ordering of arrival will not cause incorrect knowledge.

F. Graph Manipulation Language

The graph manipulation language provides the function of allowing a new graph modification and coordination operation in much the same way as the meta language allows for a change in the syntax. It is not a common language of the orderwire, but it provides the necessary structural changes that cannot be accomplished by any of the other language functions. It is strongly recommended that INTERLISP 10 or an equivalent be used for this function with the appropriate changes in the naming and execution to accommodate the communication system. It would not be a part of the universal orderwire language. It would be brought into operation by a specific orderwire.

G. Remote Control Language

This language should be integrated with command and request and update languages. The exact form should be an item for study. It is a specialized language that allows efficient manipulation of specific equipment. It will probably be the same as that used by a node to control its own equipment. The system executive talks to the lower levels of the node system with this language. The need for the capability of remote control languages arises in that some channel creation processes are more effectively accomplished by a controlling node. Thus, while an arc is being set up, one end of the arc relinquishes control to the other for a specific set of equipment. In this case the orderwire is effectively a remote terminal to the node that is in control of the equipment coordination and setup. At the same time, since it is part of the node's language for that orderwire channel, it is properly coordinated with the naming and usages of the node systems.

H. Command Language

The matter of command vs request is more a matter or priority and expediency. The command language is used only in highly critical situations, in which the system may suffer considerable damage, to support a drastic action. It is characterized by a shortcutting of many of the system safeguards. (There is a definite risk that it will cause the system at a node to fail to perform at all.) It is also used when a safeguard must be modified or bridged.

I. Request Action Language

The request action language is the language of an orderwire that gives the receiving node a goal to seek and expects some result at the appropriate time. A priority is generally transmitted with the action request. The receiving node takes the request and adjusts its activities to accommodate the request. The actions must be logical within the framework of the system. If they are not, or if there is an unresolved ambiguity, a return request for further information may be necessary in order to continue. Loss of integrity of the node system never occurs.

J. Request Information Language

The request information language is the complement to the knowledge update language. It is a request for information from the receiving node that the node may or may not have. It may be able to send out further queries to other nodes or just to that specific node. The corresponding syntax in a natural language is called the question.

K. Formulate Frame of Reference Language

When communicating, even when using the same syntax, the knowledge which is brought to bear using the syntax is important. In computer systems this process is accomplished by such mechanisms as log-in procedures and job control languages. Neither of these two examples really applies to the orderwire problem, but the underlying rationale of limiting the semantic world on the incoming orderwire is needed. In a two-party arrangement we could consider all knowledge to be valid, but even here there are times that the semantic base needed is smaller than every knowledge fact that has ever accumulated on the channel. Historical information is important to reduce the necessary communications to improve error control.

In a multichannel system such as the Navy system, it is even more important to have mechanisms to control the meaning of a dialog. Thus, a frame of reference language is needed.

When the channel is set up, the node system associated with it is given a previously structured language system and a related knowledge base upon which to operate. We could assume that the orderwire that is transmitted over a channel is about that channel. It is automatically set to have a frame of reference that consists of knowledge about itself, channel, and the initial language system. At the beginning of the channel no other knowledge base has appropriate meaning. However, the node operating system will not leave the channel to continue to have that knowledge base. The node operating system will want to use that orderwire channel to talk about users or other channels.

The normal naming system does not work too well at this point, since choice of dictionary is under discussion. Thus, a special naming system must exist for this, the framing language. It could be the fully expanded names, since these are unique no matter which frame of reference we are in. However, the fully expanded names are still too costly in channel utilization. A better approach is a symbol string that traps the language into a high-level allocation system. In this system short names may be available to handle long-named items. Naming is necessary even at this level. The topics of discussion in the framing language are decisions as to which sublanguages will be used and which knowledge basis will be combined for use. The changing of the frame of reference language is a procedure that needs definition, but the change will not be executed until after finishing with the present communication that evoked the change. This does not mean the changing of the structure of the language, but selecting which of the existing sublanguage dialects should be used.

The frame of reference language helps in the "frame of reference" problem in that boundaries of activity are defined, but further control of frame of reference is needed to completely handle the problem.

The naming language can greatly control the frame of reference problem. A name is only given if there is a definite stability or abstraction required. A frame of reference for an action is always related to stability or abstraction. Therefore, in theory we would suppose that names could be given to a frame of reference. This is easier said than done. It is not known whether a name can be generated in a mathematical sense for every frame of reference, since the set of names; of necessity, must be countable, and frames of reference may not be countable. But from a practical point of view, a system cannot be designed that espouses a noncountable set of frames of reference. In order to accomplish an appropriate naming structure to handle the control of frames of reference, it is necessary to at least name a Boolean class operator. Naming structures form a DAG rather than a tree.

L. Naming Language

The naming language is used to connect items of the semantic net to items of the orderwire statement. Almost every language has a syntax different from that of the action-knowledge syntax. The usual naming language as found in languages such as FORTRAN is an alphanumeric string with the first symbol a letter and the immediate neighbors of the string nonalphanumeric. When systems were designed, these naming conventions were too restrictive. Other characters that interacted with the system were introduced such as < . . . > . . . in Tenex. This allowed names to be unique to a program without being awkwardly long. This feature is also required of our system.

Features of the naming language:

Mostly alphanumeric

Delimited by nonname symbols

Composite names are permitted

Numbers are always some type of index

Versions are permitted

Control of version is permitted

Version time control (part of semantics)

Locking position in call-up

Normal visibility

Visibility control

Minimum confusion

Minimum required specification

Consider the name to be composed of

Alphabet parts

Number parts

Example of a Naming Language (tentative)

< name >: = < name string > | < name string > · < name string >

< name string >: = < name char % | + name string > < name char >

< name char >: = + all print symbols except "<" = "=" "_"

and all single symbol operators >

< named >: = +name > "< "=" "<"

Notes on Syntax

1. Lower and upper case will be interpreted as upper case letters.
2. Symbols that have meaning in any sublanguage are excluded.

Comments on assignment in naming processes. Note that most process have names that are used for the purpose of bridging over the stepping procedure of a single-channel input, or in a process to bridge back in a loop. There is no desire to save each step as a new version, but many times, for obtaining system histories and troubleshooting, it is desired that histories be created. At every level of assignment, a method of turning

a new version assignment on is desired. At other times, to avoid excessive overhead, this new version should be turned off. In some systems, version numbers are used as indexes. In this system version numbers should be used actively only on slowly changing noncyclic names processed in the order of hours and not normally on operations that cause assignments of less than 10 seconds. Old versions will be off-loaded at regular intervals, each several minutes.

Possibly the following mode of action should be considered.

a. A specific frame of reference is kept for those names where version update is always required at each new assignment.

b. At times during a specified network every assignment causes only one new version to be formed.

c. Special operators will cause the system to not create a new version but to keep a running set of versions not to exceed "n" copies with the latest as the current version and previous ones corresponding to older ones in order.

d. Time of last active retrieval should be stored on all named items.

e. The concept is that if it is locked into the naming structure, then it is important enough to have some action monitoring.

f. If a variable is so temporary that monitoring is not desired, then it should be compiled into the system in such a way that the naming structure is not evoked.

g. Variables that are part of the orderwire description are always considered to be important enough for some monitoring.

h. Example of a type of name that one may have thought about is polling bit of a round-robin network protocol. By the rewrite concept, that one bit is written in terms of a normal orderwire language. These sections will be added over and over again. Monitoring the name may be desired or it may be considered too much overhead.

Note that a name description is considered as a class. If it contains no semantic value, it is an unbounded variable. If there is one and only one semantic reference, it is an item. If it has more than one semantic reference, it is the class of all those references.

Every item of the system has a unique name. The collection of all unique names is a tree.

Version refers to history and disposition of reuse.

Index refers to order of items of a group of items organized by name.

The symbols : ; shall mean index.

No space is permitted in a name.

Indirect names are permitted.

Floating-point numbers or fractional numbers are not permitted as indexes.

Floating-point extraction is done prior to naming.

Examples of a naming language syntax:

SYNTAX OF SYNTAX TERMS

| | |
|---|---|
| \| = | or |
| < > | limits for new noun |
| := | define as |
| ← | prefix context |
| ⌐ | not |
| → | post context |

<octal> := 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7

<dec> := <octal> 8 | 9

<upper> := all upper case letters

<lower> := all lower case letters

<letters> := <upper> | <lower>

<letsub> := other special char that act like letter

<%> := %

<all char> := <%> <ascii>

<let> := <letters> | <letsub> | <all char>

<letter string> := ⌐<let> ← <let> | <letter string> <let>

<Integ> := ⌐<dec> ← <dec> | <Integ> <dec>

<atomic name> := <letter string> → ⌐(<dec> | <atomic name>
| <letter string> <integ name> | <atomic name>

<big name> := ⌐<atomic name> ← <atomic name> | <big name> <atomic name>

<ver name> := <big name> →⌐(":" | <atomic name>) | <big name> ":" <integ spec> ";"

<integ name> := <Integ> →⌐(<dec>)

<integ spec> := <true name> | <integ name> | <expression> | "*"

<index list> := ⌐<integ spec> ← <integ spec> | <index list> <space> <integ spec>

<pre index> := "[" <index list> | <pre index> "," <index list>

<index> := <pre index> "]"

<indexed name> := <ver name> → ⌐("[" | <esc>) | <ver name X index> | "*"

<name spec> := ⌐<indexed name> ← <indexed name> | <name spec> "." <indexed name>

<escaped name> := <indexed name> <esc>

<escaped name spec> := ⌐(<indexed name> | <escaped name> ← <escaped name>
| <name spec> "." <escaped name>
| <escaped name spec> <escaped name>

<name spec> := ⌐(<indexed name> | <escaped name>) ← <index name>
| <escaped name spec> <indexed name> | <name spec> "." <indexed name>
| <escaped name spec> →⌐(<index name> | <escaped name>) |

<left anchored name> := "$" <name spec>

<right anchored name> := <name spec> → ⌐(<let> | <dec>)

Here a mechanism is introduced to allow for remaining in situ. Other parts of the language will allow for assignment of new values to variables. These will likely have other defining statements not like this one.

<name def> := <true name> "<" "=" "<" "<" <true name>

<true name> := <name spec> | " | "<class name spec>" | "

<class name spec> := <true name> | <class name spec> <space> <true name>
| <class name spec> "," <true name> | <class name spec> " ⌐ " <true name>

Ordinarily the semantic references work from the last frame reference set and work right to left (first arrived). However, $ means that the identification works from left to right.

$DOG

will try to match DOG to highest level of the node system. If not found, then it assumes $ SELFNODE.DOG and tries again. If still not found, it assumes the highest level of the order-wire language in effect for that channel. -$SELFNODE.N F5.DOG. If that is not found, it assumes last sublanguage in operation.

SSELFNODE.N F5.NORMAL.DOG. If that is not found, it assumes last frame operation of the orderwire.

SSELFNODE.N F5.NORMAL.KNOWUD.DOG. If that is not found, it assumes closeness in semantic tree last referenced.

Functions relating nodes:

Find individuals of the name

Final value for the names

Set value for the name

Final total specification of name

Final level specification of name

Set display communication level

Test on uniqueness

Force uniqueness

Find the shortest abbreviation

Control versions

Control indexes

Name names

Control case

Special character

OR joining avb (in syntax represented by space) Notes on Semantics

The presence of a name is adequate for recall. Pure numbers are condensed as indexes.

Repeated name strings are collapsed: ie, John-John is equivalent to John.

Setting without an index will increment index first. If index is not specified, highest index is retrieved.

If - = minus is intended, a space must be present.

Setting sets at lowest level of name structure.

Retrieve will attempt at each level up until found or blocked.

An index of ∅ will force to higher level.

Extra specification will not cause problems.

Examples in System

John

John-Dave

Dave-John

Bob

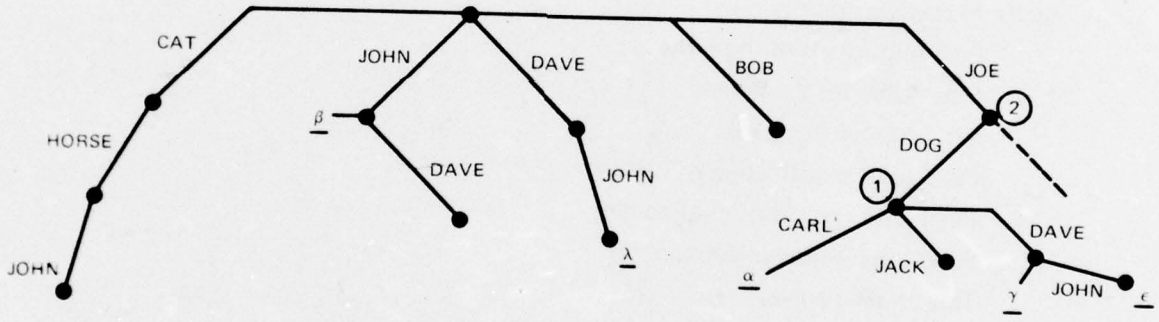
Joe-Don-Carl

Joe-Don-Jack

Joe-Don-Dave

Cat-Horse-John

Joe-Don-Dave-John




Suppose language depth is at 1

Carl will get Alpha

John will get Beta

Dave will get Gamma

Dave-John will get Epsilon

Dave – 0 John – 0 will get Lambda

Dave – 0 will get nothing

Setting Date – 0 will put a new branch at 2.

M. Screening Language

Some communication functions require a broadcast operation. In this situation some of the information over the channel is not to the node in discussion. A special mechanism is required to ensure that the destination is aware that a particular part of the transmission is for the node in question. This mechanism has to be inserted and screened in order for the node to know that the information was to the node. It has similar characteristics to the frame of reference language in that it delimits the acceptance of the information.

### 3.3.3 INTERACTION OF SUBLANGUAGES

The aspects of language change in several ways.

A language can be composed of several interacting sublanguages. A new language can be obtained from an old language by changing the syntax.

This section presents an outline showing the interaction of the various sublanguages. (See fig 3-7.)

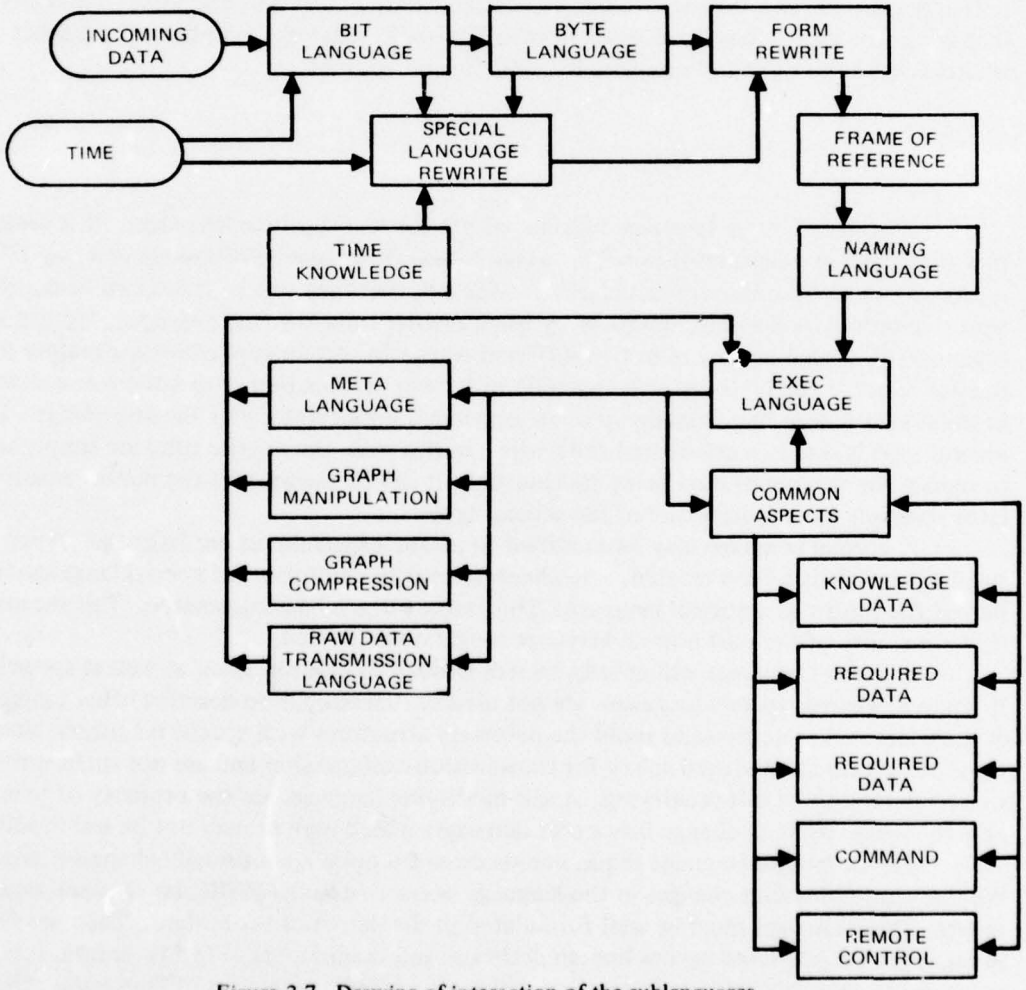



Figure 3-7. Drawing of interaction of the sublanguages.

### 3.3.5 NORMAL LANGUAGE

The primitive orderwire language must have several essential characteristics. It must be able to describe and name actions or parts of actions. It must be amenable to partitioning; ie, many short sentences will lead to one large action. This latter requirement arises because short sentences can be received better than long sentences. In this context, the ordering of the short sentences should not be of great importance.

The normal orderwire language is one that has been constructed from the primitive orderwire language but contains the syntax for the primitive orderwire language. That is, a statement written in the primitive orderwire language will still be interpreted correctly. For the most part, the normal orderwire language will be tailored to the particular orderwire channel and may or may not share parts with other orderwire channels. The primitive orderwire language is common to all normal orderwire channels.

It is desirable that the normal orderwire language, together with the primitive orderwire language, be a context-free language or at least have the nonvocabulary part context-free; ie, a symbol may be substituted for a string of characters. If contexts are required, then only left contexts and features should be utilized. Left contexts define those characters that are to the right. Since one reads from left to right, whatever is to the left precedes whatever is to the right and modifies its meaning and content.

### 3.3.6 SPECIAL LANGUAGES

Special orderwire language falls out of the context sensitive languages. It is desired that they have two features if possible. First, is that they have built into them a trap that returns to a normal orderwire language; and second, that they can be translated back into a normal orderwire language uniquely by using rewrite rules (language escape). We can then organize the special orderwire in two different ways. In certain applications, a remote manipulator concept can be formulated wherein orderwire information does not cause a change in the semantic tree that is made up of all embedded initial states plus the knowledge. The second type is simply a condensed orderwire. In this case, the rewrite rules are simply used to reduce the volume of data being transmitted. It can be shown that the remote manipulator concepts are a special case of the second type.

A normal language may be modified to create a special-purpose language. When a special language has been created, a mechanism must be built into the special language to permit the return to a normal language. This is known as a language escape. This means that a memory of the past normal language must be maintained.

A normal language will usually be rich enough to describe itself as well as special languages, whereas special languages are not usually rich enough to describe other languages. It therefore makes no sense to build the necessary structures with special languages, since these structures are designed solely for transmission compression and are not sufficiently rich or general to be self-modifying. A self-modifying language has the property of being able to change itself or change into other languages which may or may not be self-modifying.

The orderwire language requirements do not require a continually changing language. What is required is that changes in the language occur to address particular channel requirements. These changes must be well formulated in the design of the system. They are finite in number and generated by the human designers and maintainers. For this reason, it is conjectured that orderwire requirements will need a totally nested set of languages. That is,

from any form of the language used, a new language can be formed derived from the old language and each new derived language provides a more condensed notation. The escape mechanism allows for the return to the previous old language. When the escape mechanism is used a number of times, it will return the orderwire to the primitive language.

A requirement exists for the capability of the primitive language for naming a language and for the inplementation of that language. A named language can be generated from the current language or by additions to a named language. The orderwire shall be used to make these modifications to itself or other orderwires.

There must be a language attached to any active orderwire whereby that orderwire is interpreted. Orderwire languages are considered to be slowly changing structures, and, therefore, changes in orderwire language for a particular orderwire application must have statements carefully checked. Verification of changes will be required in the orderwire that makes changes to the orderwire language.

# 4.0 IMPLEMENTATION CONSIDERATIONS

## 4.1 CONDENSATION OF ORDERWIRE INFORMATION

The basic function of a communications network is to communicate information. Any overhead function such as orderwire information reduces the amount of information a communications network is able to accommodate. Therefore, we are encouraged to minimize the number of bits of orderwire information that must be transmitted to convey the orderwire message.

### 4.1.1 COMPRESSION OF GRAPHICAL REPRESENTATIONS

#### 4.1.1.1 Linear Graph Copying Technique

a. Copying of Graph

   1. A technique exists which will allow us to copy a graph in a number of steps proportional to the number of arcs.
   2. The technique requires that the area being used to copy be measurably distinct from the area where the original graph exists.
   3. The technique requires fewer steps if the original does not have to be preserved.
   4. The technique consists of generating an image of a node of the original graph, and converting its branch to point to the new node. If, however, the branch already points to a new area, it is not copied but made to point to new node images.

b. With the above algorithm, the transmission of graphs copying a linear graph is possible using only a number of steps pertinent to the number of arcs.

### 4.1.2 THE GRAPH REPRESENTATIONS IN COMPRESSION

We envision the graph compression problem as a two-stage procedure:

1. The conversion of graph to digital data
2. The digital or probabilistic coding of the digital data

Compression can occur in either stage or both stages of the coding procedure. In general, the more compression that takes place in stage 1, the less will be possible in stage 2. Also, if stage 1 incorporates relatively little data compression, stage 2 can provide relatively greater compression. If the digital data compression scheme to be used is optimal, it may be wisest to let the second stage handle all the compression while the goal of the first stage should be to convert from graph to digital data as faithfully as possible. In general, the more compression achieved in the first stage, the greater the loss of information that is irretrievable.

Another point to be considered involves the coding of the digital data. It may be desirable to perform a one-to-one transformation on the bit stream before it is processed by the data compressor. The purpose of this compression is to reduce the "apparent" entropy of the bit stream; ie, the bit stream the data compressor sees. If the data compressor compresses bit sequences on the basis of the relative number of ones and zeros contained in the sequence -- ie, compressing sequences with mostly zeros or mostly ones to a greater extent than sequences of approximately equal numbers of ones and zeros -- then it would be advantageous for transformed data to contain a greater disparity between the number of zeros and ones than the original data. Such transformation is possible when there is some inherent structure in the graph. In other words, the branches are not connected to the nodes entirely randomly. For instance, let us assume that many of the nodes have branches terminating on a specific node, say node P. Furthermore, let us assume that it is the first branch emanating from a node that is likely to terminate on node P. Also, let us assume that, on the average, there are three branches emanating from each node. Then, in accordance with this coding scheme, the binary code for node P will be occurring periodically in the data stream in some statistical sense. Assuming no nodes are labeled, the code for node P is likely to occur as every fourth symbol for the conversion that separates the node. Therefore, if we performed a transformation on the original data in which we subtracted the code for node P from each fourth symbol in the original sequence, we should generate a zero for every fourth symbol. The transformed data sequence would then contain far more zeros than the original sequence. In this case, the "apparent" entropy of the data will have been decreased and the potential compressibility of the data will have been increased.

The characteristics of the graph structures that we wish to deal with in general are depicted in figure 4-1. There are a number of nodes, and each node may have any number of branches coming into it or going out from it. Branches going into a node have an arrow pointing toward the node, and branches going out have an arrow pointing away from the node. The branches going out from the node are numbered consecutively starting with 1. In addition, a branch may or may not be labeled. All branches start and end on nodes.

We want to compress the information represented by the graph as much as possible in order to communicate it most efficiently from one point to another. The information in the graph is to be coded in binary form in such a way as to minimize the number of bits required to convey the information.

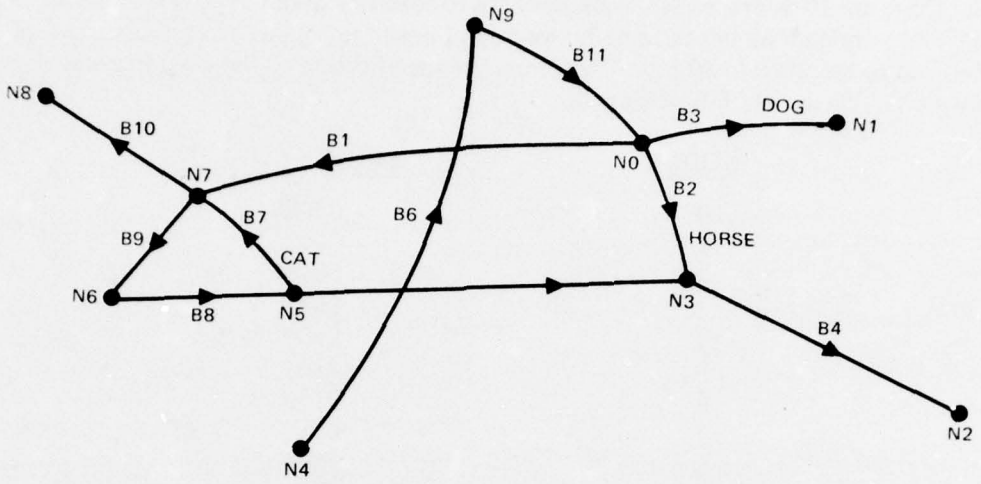



Figure 4-1. Graph structure.

The following coding scheme was developed to solve this problem. Label all nodes starting with 0. Thus, if there are N nodes, the nodes will be labeled 0, 1, 2, ... N-1. Therefore, log N (log to base 2) bits are required to specify a node. In addition to specifying the nodes, we also wish to specify a comma, which is used to prevent ambiguity, and a label flag. These terms will be explained later. Therefore, a total of log (N+2) bits are needed.

The coding scheme works as follows. We list, in order, the binary codes for the nodes at which the branches emanating from node 0 terminate. In other words, the node upon which branch number 1 emanating from node 0 terminates is listed first; the node upon which branch number 2 emanating from node 0 terminates is listed second, etc. When each outgoing branch from node zero has been accounted for in this manner, the binary code for the comma is inserted. The binary codes for all nodes upon which outgoing branches from node 1 terminate are then listed in order, followed by a comma, and then all nodes connected by outgoing branches from node 2 are listed, etc.

In addition, after the specification of each node, if the branch coming into that node has a label, then the binary code for the label flag is inserted. This is followed by a binary code, M bits in length, which represents the alphanumeric label for that branch which connects the two nodes under consideration. This scheme results in the greatest compression, provided most of the branches are not labeled. If most branches are labeled, it would be assumed that the specification of each flag could be omitted.

This scheme could be simplified even further by eliminating the comma. If it were not possible for a branch to emanate from and terminate on the same node, there would be no need to specify the code for the comma. Instead, after all the nodes connected by branches emanating from node 0 had been specified, the code for node 0 would be inserted as an end-of-node-0 marker. Then all nodes connected by branches emanating from node 1 would be specified, followed by the code for node 1 as the end-of-node-1 marker, etc. No ambiguity could result because a branch could not emanate from and terminate on the same node.

Let us construct an example of the coding scheme using the graph shown in figure 4-1. There are 10 nodes, so 4 bits are required to specify a node. Since no branches emanate from and terminate on the same node, we do not need the comma. Let us assume that all labels can be specified in 30 bits. Therefore, the specification of the label flag will also use only 4 bits. We use the following code:

| NODE | BINARY CODE |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| Label Flag | 1111 |

The code would appear as follows:

| NODE | EMANATING BRANCH | TERMINAL NODE | CODE | LABEL |
|---|---|---|---|---|
| N0 | B1 | N7 | 0111 | |
| N0 | B2 | N3 | 0011 | Horse |
| N0 | B3 | N1 | 0001 | Dog |
| N1 | (None) | | | |
| N2 | (None) | | | |
| N3 | B4 | N2 | 0010 | |
| N4 | B6 | N9 | 1001 | |
| N5 | B5 | N3 | 0011 | |
| N5 | B7 | N7 | 0111 | Cat |
| N6 | B8 | N5 | 0101 | |
| N7 | B9 | N6 | 0110 | |
| N7 | B10 | N8 | 1000 | |
| N8 | (None) | | | |
| N9 | B11 | N0 | | |

The code would appear as follows:

| N7 | N3 | label flag | label | N1 | label flag |
|---|---|---|---|---|---|
| 0111 | 0011 | 1111 | (30 bits to specify Horse) | 0001 | 1111 |

| label | End of node 0 | End of node 1 | End of node 2 | N2 |
|---|---|---|---|---|
| (30 bits to specify Dog) | 0000 | 0001 | 0010 | 0010 |

| End of node 3 | |
|---|---|
| 0011 | etc |

After the graph has been converted into a string of binary numbers, it may be possible to compress this bit string even further by using a binary source coding technique. In particular, if there is any structure inherent in the graph, such as a preponderant number of branches ending on the same node, this redundancy would be reflected in the bit string. The bit string has an entropy of its own, and this entropy will allow the bit string to be converted into a shorter bit string containing the same information which can be reproduced exactly.

If just one simple graph is to be transmitted or stored, it probably will not be worth the effort to use coding on the bit string, but if large numbers of graphs are involved, or one very complicated graph, the effort will be justified.

The techniques that could be used for this coding are described by J Lawrence, of the Naval Electronics Laboratory Center (ref 3).

### 4.1.3 METHOD OF CONDENSATION OF LANGUAGE – COMPRESSED LANGUAGE

a. The compressed language consists of using context and time to reduce the volume of data in terms of the number of bits necessary to convey the orderwire message. The advantage of compression is that it reduces message delivery time and increases communication efficiency.

b. The time of transmission of data is generally not used for interpretation of the orderwire because the orderwire arrival of generation time is not well related. However, in some special instances, as in the Link 11 net, the cyclic nature of the orderwire is observed. Communication overhead can be greatly reduced by using a compressed language tied to time of transmission.

   The criteria for establishing a time-synchronized orderwire are, first, that an order of events is well related into a timing scheme. The cyclic aspect of a net is an example. Second, that the orderwire information is sufficiently repetitive in structure that only a small fraction of the data has to be transmitted at each occurrence of the orderwire; and to set up the use of the time-related orderwire is sufficiently simple that the setup penalty is significantly less than the gain. And, third, the errors in the transmission are of a form that allows for reasonable recovery.

c. The criteria whereby the system can recognize an opportunity to utilize the time-related orderwire to advantage will have to be generated in the growth of the system by the system analysts.

d. The general structure of the compressed orderwire is such that the time of transmission is known by both transmitting and receiving sites. Also, each bit following the indicated time has an explicit meaning that is translatable into a part of the normal language statement. The translation process must also be known by both transmitting and receiving ends of the channel.

e. The method of establishing the compressed language is that the controlling node communicates to each node involved a time function, a starting time, and a bit translation procedure, and that the compressed language process is invoked.

---

[3] NELC Technical Note 2801, Application of Schalwijk Source Coding Techniques to Pictorial Sources, by J Lawrence, 4 October 1974 (NELC TNs are informal documents intended chiefly for use within NELC.)

f. The syntax of the compressed language is as follows:

SYN: a description of what is to be done by the orderwire if the bit sequence is x : = x1, x2, x3, etc

TIM: time of the first bit is equal to "time"

CHK: check to determine if flags agree with this sequence of flags. "–bit seq mode," + "compression mode"

SET: 1 + "bit seq mode"

SEM: Clear prior string, and advance the time function. Trim the bit data. Pass result to next parser.

SMN: Trim input bit string prior to syntax universe.

N:

CND:

There will be one of these for each x. In some situations some elements of the syntax description will be missing. SYN is the syntax description; TIM is the time-matching function; CHK checks the features for both the presence and absence of flags; SET sets the new features or changes some of the old features; SEM tells the system what to do if the syntax is found; SMN tells the system what to do if the syntax is not found at this point. SMN is a general function not necessarily related to the specific syntax equation. It is not repeated for each x. CND can be used when peculiar tests are needed. The SYN statements can handle prior (left) contexts, but not posterior contexts (right). The description in the SYN statement is a rewrite of what is to be done, in the form of a character string. In some simple cases this part of the rewrite can be eliminated entirely, and an action, described in the SEM part of the rules action of existing equipment in terms of the formalized equation, may take this form.

g. Other compressed language mechanisms are permitted that do not involve the timing function. These are used to increase efficiency of transmission of certain types of user or orderwire data. The data are generally decompressed into the normal orderwire language prior to interpretation. The transmission of data in non-start-stop formats such as serial data, Hoffman codes, and graph and table transmissions would normally not use the normal language for the transmission. These compressed language issues are not intrinsic to the automated orderwire problem. Particular compression mechanisms will be discussed elsewhere when appropriate.

### 4.1.4 COMPRESSION OF DICTIONARIES FOR TRANSMISSION

Assume there are N words of varying length. They are composed of alphabetic characters and are to be transmitted. It would be advantageous if the words could be coded in such a way as to minimize storage requirements. The following proposes one such code.

If the alphabet used consists of M symbols, then log2 M bits are required to express each symbol. And if there are in all T characters in the list, T log (M) (log to base 2) bits are required, if coding is not used.

The following method has been devised and illustrates how compression can be achieved. First, list all words in alphabetic order. Next, put all words in a line and retain the original ordering and separate the words with semicolons. A new symbol, an asterisk, is introduced to indicate the omission of certain letters. These preliminaries are followed by the application of certain rules:

a. If a series of more than three words begins with the same symbol, indicate by an asterisk following the first letter of the first word of the series and delete the symbol from the remaining words in the series.

b. Indicate end of coding by an asterisk preceding the first word not belonging to the series.

An example of this coding is as follows:

Original form: ARM ARMY; ATTORNEY: AZTEC; BOON

First-pass coded form: A*RM; RMY; TTORNEY; ZTEC; *BOON

These steps are then repeated until no more compression is possible.

Since each symbol is replaced by three symbols (two asterisks and the letter), the same symbol must occur in the same position for more than three successive words in order to realize a reduction. A further example illustrates reduction with multiple passes.

In this example the list of words was taken from the dictionary:

AMEND
AMENDATORY
AMENDMENT
AMENDS
AMENITY
AMENORRHEA
AMENSAETTHORO
AMENT
AMENTIA
AMENTIFEROUS
AMERCE
AMERICAN
AMERICANA
AMERICAN BEAUTY
AMERICAN CHEESE

There is a total of 15 words and 137 symbols. Adding a semicolon would give a total of 152 symbols and produce the following coded form:

A*M*E*N*D; ATORY; S; *ITY; ORRHEA; SAETTHORO; T; TIA; TIFEROUS; *R*C; I*C*A*N*; A; BEAUTY; CHEESE; *****

The total number of coded symbols is 99 so the compression is $\frac{137}{99} = 1.37$.

To achieve even greater compression, each symbol could be Huffman coded according to its statistical frequency. Then shorter bit sequences would be assigned to the more commonly occurring symbols, such as the vowels and the asterisk and semicolon symbols.

## 4.2 EQUIPMENTS

### 4.2.1 INTRODUCTION

Each node in the orderwire system has its own equipment complement to manage. While it is expected that a standardized suite of equipment will be used, each node may possess some functionally unique abilities which imply the use of different classes of equipment. Besides the logistics aspects, this gives rise to the problem of where the equipment knowledge should reside.

The suite of orderwire equipments presently involved consists of antenna controller, cryptos, multiplexers, modems, computers, etc. The computers and their attendant storage facilities will contain data bases and procedural specification.

For each of these equipments which exercises a function which has to be controlled, the orderwire system must have the requisite capability, such that:

c. Every state of the equipment, including power turn-off, can be immediately selected for that equipment by remote control through one or more control lines, except for initial power turn-on.

d. The state of the equipment can be ascertained by interrogation from the control system without modifying the functioning states of the system. However, status response states might not be accessible during this type of operation.

e. Each change of state of the equipment shall be available for monitoring by the control system if it so desires. An interrupt line shall be activated by any change of the equipment state.

f. Significant state changes shall cause a different interrupt line to be activated.

g. Any interrupt lines to the control system can be inhibited by the control system.

h. Each interrupt line can be reset individually, or in groups, by the control system.

i. Two separate control line ports should be available in which the latest control from either port will always be honored. For reliable operation, two control systems are needed. Each must be able to get to the equipment separately. To accomplish this, two ports are needed at each equipment.

j. Controls from either of the control ports can turn the other off. The state in which all control ports are turned off is prohibited.

k. Every significant state of the system should be controllable by manual operation at the equipment switches when not connected to remote control or when remote control makes them act.

l. All controls, including manual, are effective at initial power up; ie, default or manual control is to manual.

m. Remote control can inhibit, or reinstate, the manual controls except for the power-off switch.

n. All manipulation of manual controls causes capture of the manual switch settings and an interrupt line to be activated. This is true whether the manual controls are inhibited or not.

o. All visual indicators show true state of the machine equipment.

p. Each equipment shall have a number of status indicators that reflect the status of the control system and indicate nothing concerning the equipment to which it is associated.

q. Each equipment shall have a manually adjustable set of switches that do not control the function of the equipment but are capable of being read by the remote control system. Seven alphanumeric digits would probably be adequate.

r. Except for the initial power-on switch and the set of switches described in item n, the manual system need not be attached to the equipment. If it is not attached to the equipment, it should be capable of displaying the settings of the switches described in item n, and be able to override these settings.

s. Any sequencing that is normally a function of the equipment self-control can be inhibited or stopped at any step by the remote control system. (Exceptions are made with regard to cascade steps that have no internally well defined states. This is to allow the stepping through a system to determine system flow and faults.)

A communication system consists of moving data from place to place. When sharing assets, it is sometimes required to switch, combine, and separate data streams to effectively utilize the given equipment. The philosophy that has been considered for this sytem is that this data transfer is effected by some kind of multiplexers whether they are separate equipments or another function built into the software of the system. If the data transfer is effected in a computer of the system, then data are delayed and the capability of the computer is greatly reduced. General-purpose computers are not efficient for data shuffling.

### 4.2.2 MULTIPLEXERS

There are two basic types of multiplexers – those that handle analog data and those that handle digital data. For most of this study, the concept is essentially that of control of digital data. Since voice is an important part of Navy communications, voice will presumably be converted to digital information already. The system request for a voice link will have been formulated in some way so as to provide the system with digital control information, and the voice itself will have been converted to digital by means of a vocoder. In order to be effective, the system will have to switch the analog voice so that an analog switch/multiplexer would be involved in the total communication system.

Digital switch/multiplexers are of particular interest since all data movement in the system is controlled by such multiplexers. (Note that for effective voice conferencing it will be required that vocoders provide the necessary signal to determine a block of data for which the vocoder can do rational processing).

The concept of the orderwire system presented in this document assumes that a network or equivalent of completely programmable multiplexers is available and part of the system is on a nonblocking redundant arrangement. The multiplexers can be used in a mode to send data to inexpensive store-and-forward memories when appropriate. Also, the data can be channeled into the control computer for orderwire and other communication needs. With the multiplexers external to the computer, voice channeling can easily be set up in a manner such that there is no perceptible delay and no reset is needed in establishing conferencing. The multiplexer should be asynchronously connected into the network with a clock controlled by the multiplexer. Notice that in this discussion the completely programmable multiplexer is also a completely programmable demultiplexer.

Analog switching and frequency division multiplexing are also needed in the system, but these processes are nearly invisible to the orderwire system. This equipment represents one aspect of the set of processes and procedures that must be manipulated to generate connectivity between nodes of the system. To the orderwire system they do not represent media connectivities of the system. As far as the orderwire system is concerned, the data are put into a port of a modem and magically appear at a port of another modem on another node of the communication system or on one of the modems of its own node. In the process of setting up a connection between two nodes in a subsystem, the node techcontrol system, in conjunction with the orderwire system, communicates to set up the appropriate tuning to affect system behavior so that a bit sent into one modem appears appropriately at the other modem. Once this is done, the orderwire management system considers this a new connectivity of the system to be used to transmit orderwire and/or data. The system may consult an "expert" to determine whether another connectivity can be formed for internal reasons," or, because of degradation measurements reported by the techcontrol, it may release control over a connection. A valid data connection is considered to be between two modems.

If, for example, it is decided that no block data shall be received and then immediately retransmitted, the point of connectivity then moves to the red interface of the crypto. This is not an important difference in the nature of the orderwire.

### 4.2.3 EVENT CONTROLLER

In a communication system of this type many actions must be precisely timed. In the control system time, this is difficult. A special device is needed that can be instructed by the system to generate a pulse on a particular line at precisely the correct time. At other times, the system needs an alerting clock that tells the system that it was asked to respond at this time. Time-outs can be accomplished by this means.

### 4.2.4 ORDERWIRE CONTROL EQUIPMENT

The possibility of an automated communication system is based on the concept that a controlling system exists that can carry out the desired action after analyzing the collected knowledge. The exact architecture is not important at this moment in concept formulation. It is clear that the system will have to have a reasonable capability. Slow channels need to have instant control and fast channels need not have overly compressed orderwire. It is clear that the system needs a large medium-access memory and a reasonably sized immediate

memory with a large library of semipermanent memory (photographic). Accessible archiving is required for appropriate performance analysis and evaluation.

### 4.2.5 ANTENNA, MULTICOUPLER, RECEIVERS, TRANSMITTERS

These equipments already exist, but in the future they need to be remote controlled.

### 4.2.6 MODEMS

A number of different types of modem functions are needed to handle the different baud rates available and the different noise characterizations that exist.

### 4.2.7 CRYPTO

The character of the crypto has been covered in other places. The principal concern here is that the cryptos must have the capability to interact with an AKDS.

### 4.2.8 COMPRESSORS AND EDAC

EDAC is needed to control errors in the digital stream. Compression techniques are needed to reduce transmission requirements and thus communicate with fewer bits.

### 4.2.9 FREQUENCY SCANNER

In order to make the orderwire system as transparent as possible to the user, a frequency scanner unit must be incorporated. It should consist of a highly sophisticated, computer-controlled frequency scanner with a memory storage system so that transmission picked up by the scanner will be stored for future reference.

The scanner will randomly, but effectively, monitor selected possible transmission channels and be able to automatically recommend transmissions to best suit existing atmospheric conditions.

In the event the scanner picks up a strange or foreign transmittal, this information causes a request for identification.

Eliminating the user from the system as much as possible cuts down chances for human error and also reduces manpower requirements in a communication room.

The scanner must be controlled by its own separate computer to reduce overload on the main computer system, and to ensure a working scanner system in the event of a main system breakdown. The latter reason is probably more important than the former, because information gathered through the scanner can be utilized by others.

In present communication systems it is possible for two parties to simultaneously transmit on the same frequency. This happens even though a third party is involved as a network controller. Another present drawback is the inability of a transmitting station to know that a message has failed to get to its destination because of stations transmitting simultaneously except when notified of this fact by an observer station. The frequency

scanner computer system will alert of this. A sample signal will first be transmitted by the sender, and the receiver scanner will pick it up and respond.

The standard sample signal will have to be long enough for the scanner system to complete a full random sampling cycle so that no signals are missed.

#### 4.2.10 MEMORY STORAGE FOR STORE-AND-FORWARD

A memory storage unit should be incorporated in the first stage of the orderwire system. It should have the capability to store, at least temporarily, all received information.

The information must be available on instantaneous recall, and the system must be capable of putting out information at a rate compatible with the orderwire system.

It must be capable of storing information at a rate at least equal to the fastest anticipated transmission.

It must be able to act as a two-way buffer between incoming transmissions and the main orderwire system. That is, if information comes in at a rate faster than the control subsystem can receive, the memory section must be able to store all the data and, when the main system calls for the data, feed the required data at a rate compatible with the main system. If information comes in too slowly, the memory will store it, and later feed it at a faster rate to the main system.

## 5.0 APPLICABLE TECHNOLOGIES

### 5.1 INTRODUCTION

The design and development of an automated $C^3$ system requires the cooperative interaction of a large number of technologies. The following paragraphs describe briefly a number of technologies that have to play a part in the development of a Navy automated network.

### 5.2 MEDIA TRANSMISSION TECHNOLOGIES

The first requirement of any communication capability is ready availability of a communication medium. In the Navy there are many such media each having its own role to play in the total communication picture. Some have long range, others short range; some have wide bandwidth, others narrow bandwidth.

### 5.3 AUTOMATED NARROWBAND HF, SATELLITE, UHF, OPTICAL, PACKET RADIO etc

In an automated system, methods are required to automatically measure the capability of any medium. Special equipments are needed for this function. An example of such equipment is the sounder.

## 5.4 DIGITAL SYSTEM TECHNOLOGY

The principal type of equipment involved in the control equipment design and implementation is digital electronic. This type of equipment supports all the control functions and many of the communication media functions.

## 5.5 MODULATION AND EDAC TECHNOLOGY

The conversion of the analog to digital and the improvement of the communication data involve utilization of modems and EDAC.

## 5.6 FORMAL LANGUAGE TECHNOLOGY

To represent the communicated control, a language of advanced form of control protocol is required. An understanding of computer languages greatly aids in the development of such a system control language.

## 5.7 KNOWLEDGE REPRESENTATION TECHNOLOGY

Information about the state of the system is required at each node of the communication system. The representation of information is important in accomplishing the automation.

## 5.8 COMMUNICATION PROTOCOL TECHNOLOGY

The language of the orderwire can describe the protocol, but step-by-step processes must also be understood to effect communication. Many step-by-step processes have been developed in previous Navy systems and much has been learned in their development. These processes, together with newer protocols, must be embedded in the system.

## 5.9 NETWORK MANAGEMENT TECHNOLOGY

Once a network has grown to more than a few nodes, connectivity management is no longer a trivial problem.

## 5.10 MACHINE HEURISTIC FUNCTION "EXPERTS"

Much of the expertise available to effect communication is found in the people that operate a network, and this is true also of the Navy. Methods have to be devised to transfer the human knowledge of communications techniques to machine knowledge. The technology to do this has been recently developed. It has been called building "expert" systems.

## 5.11 RESOURCE MANAGEMENT TECHNOLOGY

Every node has only a fixed plant of equipment and these are the only equipments available for communications. The resources must be managed by the automated systems to accomplish optimal behavior of the system.

## 5.12 CRISIS MANAGEMENT TECHNOLOGY

A principal reason for the existence of the Navy $C^3$ network is to provide communication in a crisis situation. The nature of crisis must be understood in order to properly design and adequately control the communications function in the time of crisis.

## 5.13 MAN-MACHINE INTERACTION TECHNOLOGY

These are three forms of man-machine interaction. First, there are the interactions of the system with the people (or equipment directly interacting with people) that constitute the users of the system. The second arises from the fact that the system is not perfect and must be supported by people to ensure that it maintains functional capability. The third is the control operator.

## 5.14 TECH CONTROL TECHNOLOGY

The system must provide (or consult with a techcontrol system that provides) the technical control over the path and equipment switching within a node.

## 5.15 SURVIVABILITY ANALYSIS TECHNOLOGY

A clear understanding of survivability requirements and their impacts on control must be gained before an appropriate design and maintenance of the system can be accomplished.

## 5.16 COMMAND CONTROL INFORMATION EXCHANGE REQUIREMENTS TECHNOLOGY

The principal function of a communication capability in military operations is to provide the necessary connectivity to the command control element of the forces. In a commercial system a demand-response system is adequate, but in military operation it is not. If we wait for a demand manifestation in a crisis, it is then too late for effective control. Therefore, the command control structure must be available and integral to the network independent of a demand manifestation.

### 5.17 DATA COMMUNICATION SECURITY TECHNOLOGY

In a military operation, data concerning communications and operations must be hidden from the enemy.

### 5.18 COMMUNICATION ROUTING TECHNOLOGY

The $C^3$ system must know how to move messages from one node to any other.

### 5.19 CIRCUIT SWITCHING AND STORE-AND-FORWARD SWITCHING TECHNOLOGY

Since not every node is connected to every other node, some messages must be routed, switched, and stored-and-forwarded to other nodes. The technology for doing this is currently available.

### 5.20 SYSTEM BUILDING TECHNOLOGY

To build a communication network, powerful tools must be brought to bear on the design and maintenance of the system. System building technology is useful in accomplishing this.

### 5.21 SIMULATION MODELING TECHNOLOGY

One of the key system building tools is the simulation modeling of the system.

### 5.22 IMPLEMENTATION PLANNING TECHNOLOGY

Techniques applicable to implementation of any large system must be applied also to the implementation of an automated orderwire system.

## 6.0 RECOMMENDED STUDIES

### 6.1 EXPANDING STUDY

To design an automated orderwire system for the Navy $C^3$ system will require a thorough examination of the concepts and possible tradeoffs involved. This document, as it now stands, is one small part of the total needed. A more complete study of the automated orderwire concept is required before planning for a design effort. Many of the following recommended studies will expand various aspects of the automated orderwire concept. They will also form a basis upon which design can be initiated. The integrated simulation model will likely evolve into the initial design.

## 6.2 CONSTRUCTION OF SIMULATION MODEL

The most powerful tool excluding the building of a prototype system is the building of a simulation model as complete as possible. An advantage of the simulation model over a prototype is the low cost of a change to the model and the shorter time before such a change can be implemented.

## 6.3 DETAILING OF PRIMITIVE AND NORMAL LANGUAGES

Only directions in the development of languages were mentioned in this document. The actual development of the languages in an environment simulating the $C^3$ system is needed to better understand the tradeoffs in the design of the languages in the area of comparisons and good specification.

## 6.4 DETERMINATION OF A FAMILY OF ROUTING DISCIPLINES

There are many routing disciplines that can be considered for the control of a system. Decisions must be made as to which combinations should be used, and criteria for change of the routing discipline must be established.

## 6.5 DEVELOPMENT OF CENTRALIZATION VS DISTRIBUTION PHILOSOPHY

The problem of exactly where the control of the system should reside has not been well answered. There is reason to believe that a distributed control is required for a fall-back position at least. But it is known that centralized control is required in some networks. The amount of centralized control that is necessary and the nature of it are questions not totally answered. A form of centralized control was discussed in this document where central control only controlled the "pricing" structure for the utilization of the system components. Effective control was given to the local nodes; motivational control was centralized. The optimum combination remains unanswered.

## 6.6 DEVELOPING APPROPRIATE SEMANTIC KNOWLEDGE SYSTEMS

In a large system in which decisions are made on the basis of available knowledge, that knowledge must be represented in a manner that is useful and shows the appropriate relationships needed to control the system. A detailed knowledge design is needed to understand the type and access questions for the knowledge. Another aspect of this problem is how to make decisions on knowledge that is not current, not exact, not complete, and not necessarily consistent.

## 6.7 INTEGRATING 6.3, 6.4, 6.5, 6.6 SYSTEMS INTO THE SIMULATION MODEL OF 6.2

In paragraph 6.2 a simulation model was discussed. Not only must a model be developed, but the concepts of 6.3, 6.4, 6.5, and 6.6 and many others must be integrated into the simulation system and exercised in that system to determine the behavioral tradeoffs. Only in this way, can the interactions of the design decisions become clear. An analytical model of such a complex system is impossible.

## 6.8 DEVELOPING STANDARDS FOR EQUIPMENT PROCUREMENT

One of the factors clearly visible in this study was that we do not have automated orderwire when control of most of the equipment is in a manual manner with no provisions for remote automated control. A few pieces of equipment can be controlled remotely, such as the radio gear on the LHA, but for the most part even new developments in equipment have no mechanisms to allow for remote control. It is noted that the requirement for remote control does not, in general, reduce the requirement for local control of the same equipment. Maintenance of equipment is difficult without local control.

## 6.9 DEVELOPMENT OF SPECIAL EQUIPMENT

*Digital equipment has reached a plateau in processing capability.* The splitting of functions appears to provide increased capacity. For this reason, the removal of a number of functions from the central controlling element is suggested. These functions are multiplexing, store-and-forward for simple channels, event timing control, and media scanning and measuring. A design specification for these equipments should be generated. In the simulation model these functions must be part of the simulation model and not a functional added on.

## 6.10 DEVELOPMENT OF PROTOCOL

A number of network disciplines presently exist that solve problems of narrow scope. The reason for the existence of these disciplines does not disappear with the advent of an automated orderwire system. Therefore, they must also be implementable in the automated system. For the same reason, they must be implemented in the simulation model with the orderwire system languages to ensure that some hidden quirk does not appear later.

## 6.11 DEVELOPING CONTROL OPERATOR CONTROL LANGUAGE AND PROTOCOL

At each node there must be at least one operator whose function is to communicate with the executive of the system to implement control, answer questions concerning data the system cannot get directly, and receive trouble status from the system. The extent of protocol and language needed for the machine to tell the operator about the system and the extent of the latitude of the control statements must be explored.

## 6.12 DEVELOPING SYSTEM MEASUREMENT AND PERFORMANCE CRITERIA

An automatic system requires that knowledge about how the system functions be available to the system. Certain data parameters need to be measured, collected, stored, and centralized. A set of operations must be developed to make meaningful control decisions using these parameters.

## 6.13 DEVELOPING UTILIZATION VALUE SYSTEM

A value system has to be designed to determine when a set of actions should be taken, what the expected value will be, and what the penalty will be for such an action.

## 6.14 DEVELOPING COMMAND CONTROL EXPERT

Since the system is for command control, the order of command affects the behavior of the system. Given a command structure, how does this knowledge affect the parameters and behavior of the systems? What are the rules that translate this command structure into meaningful actions? How do constraining rules interact in the system?

## 6.15 DEVELOPING MEDIA AVAILABILITY EXPERT

There are certain constraints to the utilization of any given medium. For each medium these constraints and rules of operation must be entered into the system with an appropriate processing system to allow for request for operating parameter, for request for operating compliance, and for alarming if certain parameters that change the availability of the channel are substantially altered. This knowledge can be built into the system by the inclusion of the machine "expert."

## 6.16 DETAILING ELEMENTAL ACTIONS

In this document, a number of elemental actions were described. A more careful study of what these elemental actions should be, their interactions, and how are they combined into composite actions should be undertaken.

## 6.17 DETAILING EQUIPMENT SCHEDULING EXPERT

The techniques of organizing the available equipment to provide the desired capabilities and the making and changing of assignments should be build into an expert system. Also, alarming of equipment failure should be made a part of this function.

## 6.18 DEVELOPING CHANNEL EVALUATION EXPERT

In addition to the media experts, we need an expert that determines the usefulness of a channel that has been set up by measuring test data that have been transmitted over it.

## 6.19 DEVELOPING IMPLEMENTATION PLANS

In the foregoing, a number of technical studies were recommended to gain further knowledge about the technology. However, if the Navy is ever to build an automated order-wire system, a plan must be devised that allows for the smooth transitioning from the current system to the automated system with funding increments small enough so that they are not prohibitive.

## 6.20 DETERMINING COSTS

In conjunction with the development of the plan, a development of the measure of cost and value must take place to determine effectiveness and utility.

AD-A038 968 NAVAL ELECTRONICS LAB CENTER SAN DIEGO CALIF F/G 17/2
NAVY COMMAND CONTROL AND COMMUNICATIONS SYSTEM DESIGN PRINCIPLE--ETC(U)
AUG 76

UNCLASSIFIED NELC/TD-504-VOL-5 NL

2 OF 2
AD A038 968

END
DATE FILMED
10-77
DDC

# SUPPLEMENTARY

# INFORMATION

AD-A038 968

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152

1 July 1977

## LITERATURE CHANGE

NELC Technical Document 504
NAVY COMMAND CONTROL AND COMMUNICATIONS SYSTEM DESIGN PRINCIPLES AND CONCEPTS, Volumes I through VIII, 15 August 1976

1. In block 10 of DD Form 1473, change numbers to:
   65866N, X0740, X0740 (NELC Q221)

2. On cover, under date, add:
   Changed 1 July 1977