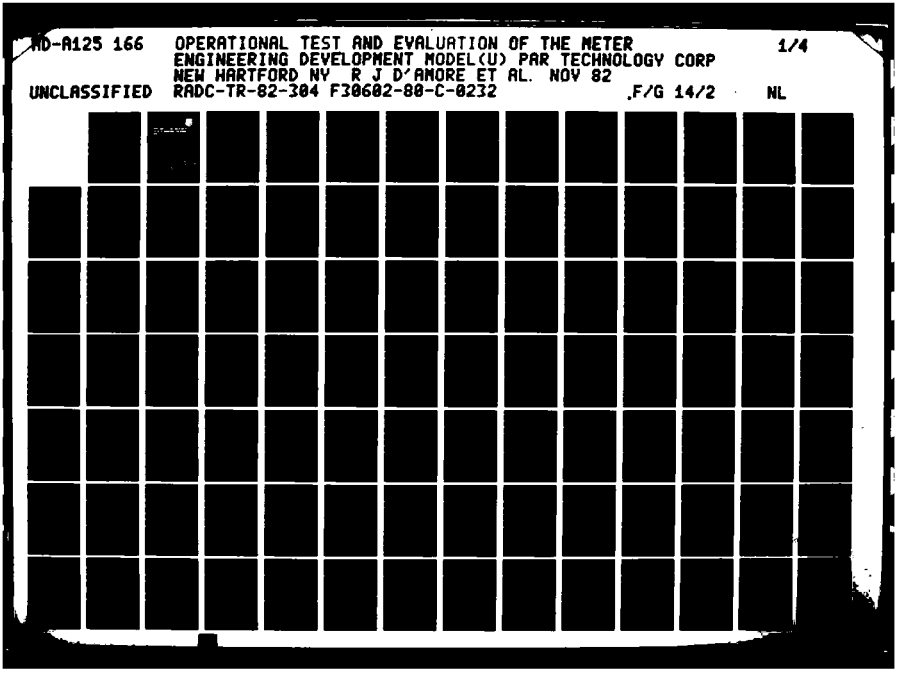
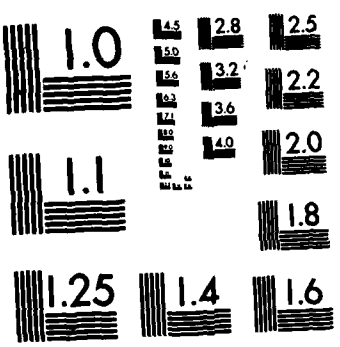
AD-A125 166 OPERATIONAL TEST AND EVALUATION OF THE METER ENGINEERING DEVELOPMENT MODEL(U) PAR TECHNOLOGY CORP NEW HARTFORD NY R J D'AMORE ET AL. NOV 82 1/4

UNCLASSIFIED RADC-TR-82-304 F30602-80-C-0232 F/G 14/2 NL





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

(12)

RADC-TR-82-304
Final Technical Report
November 1982

AD A123106

OPERATIONAL TEST AND EVALUATION OF THE METER ENGINEERING DEVELOPMENT MODEL

PAR Technology Corporation

Raymond J. D'Amore and Clinton P. Mah

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

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

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

RADC-TR-82-304 has been reviewed and is approved for publication.

APPROVED: Thomas L. Coluccio

THOMAS L. COLUCCIO
Project Engineer

APPROVED: John N. Entzminger, Jr.

JOHN N. ENTZMINGER, JR.
Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

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

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: RADC-TR-82-304	2. GOVT ACCESSION NO.: AD-A125 766
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): OPERATIONAL TEST AND EVALUATION OF THE METER ENGINEERING DEVELOPMENT MODEL	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report July 80 - Jul 82
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Raymond J. D'Amore, Clinton P. Mah	8. CONTRACT OR GRANT NUMBER(s): F30602-80-C-0232
9. PERFORMING ORGANIZATION NAME AND ADDRESS: PAR Technology Corporation, Seneca Plaza, New Hartford NY 13413	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 64750F, 19550705
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRDT), Griffiss AFB NY 13441	12. REPORT DATE: November 1982
	13. NUMBER OF PAGES: 304
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES
RADC Project Engineer: Thomas L. Coluccio (IRDT)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Message Retrieval
Associative Retrieval
Statistical Retrieval
Intelligence Data Handling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This report documents the Operational Test and Evaluation (OT&E) of the Meaning Extraction Through Estimated Relevance (METER) System conducted at Hq Military Airlift Command, Scott AFB IL, under RADC Contract F30602-80-C-0232. The two year effort provided for continuing enhancement of the METER System, as well as tailoring it to interface with the operational message processing system that existed at the Hq MAC intelligence facility. As well as the METER development, interfacing and in-

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

stallation, this report covers analyst training and evaluation of METER's potential utility to the intelligence community.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

PAR Technology Corporation acknowledges the contributions of a number of groups and individuals to the successful completion of the METER project.

1. To Rome Air Development Center and in particular, Mr. Thomas Coluccio, the RADC project engineer, for his overall support and direction given to the program.

2. To the Military Airlift Command and those who supported the operation and evaluation of the METER system to include Major Stephen Marshall and SSGT Tim Hynes, MAC/INY, and MSGT. Robert Nielsen, SSGT Wayne Cox, TSGT Lelind Jackson, A1C Marie Hyde from MAC/ADXI.

3. To INCO incorporated, McClean, Va, and in particular, to Messrs. Gary Kincaid, Mike Pheil, Tom Hogan, and Jim Pierce (located at MAC) for their consultation on the MAXI system.

1. Introduction

This report documents the Operational Test and Evaluation (OT/E) of the Message Extraction Through Estimated Relevance System, Engineering Development Model (METER EDM), conducted by PAR Technology Corporation at the Military Airlift Command, Hq Scott AFB IL. The work was performed under contract F30602-80-C-0232 to Rome Air Development Center.

This two year effort provided for the continuing development of the METER system at PAR Technology and for tailoring METER to interface to the MAXI system. The METER/MAXI interface development was initially performed by PAR at INCO Incorporated, McLean, Va. and, after AFIS certification, METER installation was completed at MAC. The final phase of this effort involved training MAC analysts to operate METER and evaluating the utility of METER in an operational environment.

In Section 2 of this report, the reader is provided with a summary of the evaluation results which are described more completely in following sections. Section 3, presents the efforts required to interface METER to MAXI, install METER at MAC, and train MAC users. Section 4, details the results of the METER user-based evaluation. This aspect of the evaluation is based on METER usage data collected online, interviews, questionnaire results and special work sessions overviewed by PAR evaluators. Section 5, discusses METER software reliability and other operational aspects. Section 6, outlines METER software enhancements that were made during the OT/E as well as those that could be accomplished with additional work.

There are five appendices to this report:

1. Appendix A, is the METER User's Guide provided to each METER user at MAC.

2. Appendix B, is the Operator's Procedure Manual used by MAC ADP support personnel to operate METER.

3. Appendix C, is the System Manager's Guide which provides the System Manager with a high-level set of procedures for METER operation.

4. Appendix D, is the METER Evaluation Questionnaire used to collect MAC user assessments of METER as well as user background information.

5. Appendix E, consists of a collection of statistical techniques essentially derived from METER research and development performed over several METER contracts. It also provides the reader with an overview of basic METER technology employed at MAC.

2. Summary of Results

This Section provides a summary of METER project history and overviews the results of the evaluation of METER at MAC. Related details can be found in Sections 3 through 6 of this report

2.1 Project History

The METER effort, as defined in the METER Statement of Work, originally called for interfacing with the SSB system that was to serve as the underlying support facility for analysts using OJ-389 work stations at MAC HQ. Following METER installation at MAC, analysts would then be trained as part of an Operational Test and Evaluation to be conducted by PAR. The final phase in the contract had called for a subsequent enhancement phase in which modifications would be made to the system according to evaluation results.

A complication arose when MAC/IN decided in early 1981 to replace SSB with a new system called MAXI. The switchover necessitated the re-design and subsequent implementation of a new METER interface to MAXI in order to communicate with OJ-389 work stations. Work on new top level METER retrieval modules for MAXI began in the summer of 1981 and was essentially completed by early 1982; approval came from AFIS in early spring of 1982 to deliver METER software to MAC HQ.

In May, 1982, PAR began training MAC analysts as the first stage of the METER evaluation. In May, June, and July, PAR conducted further training, structured work sessions, and interviews; concurrently, detailed usage data for the METER evaluation was collected online through instrumentation code built into METER software run by analysts at MAC HQ. The evaluation was concluded on July 14, 1982, when the METER system was turned off.

2.2 EVALUATION RESULTS

About 20 MAC/IN personnel received either formal or informal training on METER. As discussed in Section 3.5.1, formal training consisted of a training overview, hands-on training sessions, and follow-up sessions with PAR trainers. In addition to the formal training sessions, several users participated in special in-depth work sessions with PAR evaluators. The following conclusions are based on our direct interaction with the MAC user group as well as other special interview sessions and questionnaire responses.

METER CAPABILITIES EFFECTIVELY UTILIZED BY MAC ANALYSTS

- Analysts were able to perform general searches over many different message subsets without reprofiling. To search interactively and to converge on target messages, an analyst must be able to control search strategy in response to intermediate results. With the METER RESTRICT function, an analyst could define particular message-related parameters to constrain or provide a more focused search by METER of a collection of messages. Retrievals may be restricted to messages satisfying header parameters such as date/time group, source, and classication or to messages having or not having specified keywords. The overall METER retrieval process would not be biased by prior message profiling; and users are not required to anticipate in detail what interesting messages would look like.

- Analysts could initiate retrievals using messages obtained directly from MAXI message queues. When reviewing queued messages, an analyst might notice one that prompts him to search for related messages in preceding message traffic. METER allowed this to be easily performed as follows:

1. After displaying the message text at his work station, the analyst would depress the ESCAPE function key followed by the METER function key to invoke METER.

2. The analyst then would depress the RETRIEVE function key to direct METER to look for messages similar to the one currently displayed at his work station.

3. After displaying the text of messages retrieved by METER, the analyst could then print the relevant ones or copy them to his MAXI workfile. This can be done by depressing ESCAPE, WRK (for MAXI workfile operations), and STR (for storing text); the METER interface to MAXI was designed to be fully integrated with all MAXI functions, including workfile operations.

- Analysts could perform retrospective searches for messages without detailed knowledge of what they were looking for. METER does not require highly specific information to initiate an effective retrieval as in the case of other (i.e., Boolean) systems that require rather complex logical queries to be formulated before useful results can be expected. In addition, METER ranks retrieved messages according to estimated relevance; by looking at top-ranked messages first, an analyst can quickly determine whether a retrieval has succeeded. In a Boolean system, on the other hand, retrieved messages are unranked, and analysts have no good way of evaluating a retrieval without reading most of the retrieved messages. METER also aids a user by automatically expanding user queries according to information derived by METER in its statistical analysis of a data base each time it is updated. In addition, METER offline aids summarize data base contents and often suggest queries to ask.

- Analysts could construct briefing papers using METER to get source material. Analysts often use MAXI workfiles to store text of briefing papers or other correspondence created at a work station using the OJ-389 onboard editor. When this text refers to message traffic, a useful function is to be able to locate the actual messages related to it. This is easily done in METER by first displaying text from a workfile or an appropriate part of it at a work station and then initiating a METER retrieval with the displayed text as a query exactly as with the displayed text of a message. The text of relevant retrieved messages can then easily be edited and moved back to a MAXI workfile as described above.

- Users could browse in a METER data base; and analysts at MAC HQ often were able to find interesting though unexpected messages in this way. This capability is helpful since analyst may have difficulty in keeping track of multiple events discussed in message traffic. Messages of interest to a particular analyst may not be routed to him because of inadvertent omissions in profiling and indexing schemes. When METER retrieves messages of interest with a query, an analyst may close in on these using the METER CULL function. The CULL function allows users to inform METER of the relevance of retrieved messages; METER can automatically modify the original query in order to locate more of messages of interest. An analyst may also use the METER RESTRICT function at the same time here to focus on the date/time or other characteristic associated with a message of interest.

METER USEAGE

Most MAC users found METER to be a useful and efficient system. When searching for messages in the METER data base, they primarily employed the basic retrieval functions (RETRIEVE and REPORT) and occasionally the RESTRICT function, judging apparently that these were the most effective. Nearly all

users thought that METER commands were easy to use and that METER output was understandable and effective. In some cases, this simplicity actually led to problems when some analysts began using METER without ever attending an orientation session or getting a copy of the METER User's Guide; these users sometimes misunderstood the relevance histogram and stem summaries in the METER retrieval report form. One suggestion by analysts was for METER to drop histograms entirely and to increase the number of stems displayed in retrieval summaries from 4 to 6.

METER retrievals were typically performed in about 20 seconds and with about 90 percent probability of completion in 40 seconds or less. About half of the users felt that the retrieval times were acceptable; others would have liked more immediate response to all METER requests, including retrievals. On the whole, however, users generally felt that the overall time to complete a task (i.e., to get usable information through a sequence of multiple METER transactions) was quite acceptable when compared to available methods.

METER users would be willing to use METER in a crisis, reflecting general confidence in the effectiveness of METER functions. Analysts were often able to identify particular situations from their own experience where a sophisticated METER retrieval capability would provide information that would have been laborious and time-consuming to obtain through other facilities; these situations were not likely to arise during a short evaluation period, but were seen by analysts as nevertheless being important. Most users said that they would like to see METER kept operational at MAC, provided that it be further tailored. Desired changes generally identified include more online storage, faster response, and minor changes in the format of forms.

EFFECTIVENESS OF THE TRAINING PROGRAM

More time was needed for training, supervised work sessions, and follow-up for all participating analysts. Unfortunately, project and MAC operational constraints limited the amount of initial and follow-up training. There was an acute need for a long-term training program to provide support to METER users as required. Most users thought that individual METER functions were simple to learn, but they often needed help in combining them to make up effective search procedures; some of the more subtle aspects of METER, such as in the RESTRICT form, did not become apparent to users until they actually had the occasion to employ them. Witn a significant turnover in military analysts and support personnel at MAC, an ongoing training program is necessary all the more.

REQUIRED RESOURCES

The performance of METER at MAC HQ appeared to adequate for the purpose of an evaluation, but the somewhat slow processing speed and the limited data base size reduced flexibility in its operation. The PDP-11/45 processor on which METER ran was significantly slower than the VAX 11/780 processor on which METER was developed, as well as slower than the main PDP-11/70 Analyst Support Processor at MAC. This affected on data base update times as well as retrieval times; with updates taking about two hours, the METER messages could not reasonably be updated more than once per day.

A greater problem than processing time was space. The single Bunker-Ramo 1536 disk pack available for METER software and data limited maximum data base size to about 12,000 messages in 12 daily updates. MAC/IN originally hoped to handle up to 21 daily updates together, and several analysts wanted METER to keep messages for longer than two weeks, but this would fill up almost an entire BR-1536 disk pack and leave no room at all for software or user files.

The space problem was made worse by events in the Falkland Islands and in Lebanon leading to a major increase in message traffic at MAC HQ during the METER evaluation period. At times, daily message traffic exceeded 1500 messages instead of an expected 1000 messages per day. Disk space for storing messages and related statistics filled up fast, forcing the METER data base window size to be reduced to as low as 10 updates at one point. This meant that analysts could go back only ten days in message traffic.

In general, METER would have performed better in terms of response time and storage space if alternate hardware/software configurations could have been utilized at MAC. One possibility discussed with MAC/INY personnel was to put METER on an PDP-11/70 with a larger (T300) disk or with separate system and data disks.

2.3 METER PROSPECTS

In Section 6, PAR outlines several possible system modifications. With the software and procedural revisions as described, METER should be a much more flexible system from an operational standpoint. It will be fast enough so that METER preprocessing of message text can be run more than once per day without heavily loading down a processor and severely limiting user access to METER. This would permit short METER data base updates throughout a day, so that an analyst can retrieve messages more recent than the last full update. Performance should be even better with a PDP-11/70 or with T300 disk drives. Overall reliability of the modified system should remain about the same as before, since only local changes to current METER software will be required.

For METER to be maximally useful to analysts, they would have to revise their strategies for extracting information from collections of message text. They must be more willing to

1. seek out messages they want by submitting queries as opposed to simply relying on profiles. Unlike Boolean keyword retrieval systems, METER makes it easy to formulate queries.

2. follow up on messages of interest by looking for additional messages with related content.

3. think in terms of document similarity instead of in terms of predefined subject areas.

4. allow a system to suggest what messages are most worth reading.

METER on the whole provides capabilities that go beyond the basic message distribution by profiles that analysts are accustomed to, but in turn METER requires more active participation by analysts.

Alternative METER test environments could be explored. For example, along with further development at MAC, installation of METER with or without MAXI at other Indications and Warnings or Intelligence Centers would provide a broader test of METER utility. The time already spent at MAC HQ has allowed a thorough shakedown of METER software and procedures in an operational environment and has led to improvements that should make it easier to get METER running effectively and reliably at other sites.

Nonoperational environments are another possibility. Since analysts at operational sites are usually under pressure, they tend to rely on techniques that they have had success with in the past; a better situation for trying METER out might exist at a training center for intelligence analysts, where time is not critical and where no risk is involved. Similarly, a laboratory set up for analysts to experiment with and develop new procedures based on new technology might be a suitable location.

3. METER at MAC

This section provides an overview of the METER system at MAC. In particular, we give some general thoughts on the role of METER at MAC (Section 3.1), discuss the METER user's interface (Section 3.2), and the MAXI/METER interface (Section 3.3) designed, implemented and tested under this contract as part of the operational test and evaluation. (Note that detailed descriptions of the user's interface and the MAXI/METER interface are provided as separate references.) In addition we also discuss work done to prepare or set-up METER to run operationally (Section 3.4) and the training program used to prepare users to use METER operationally in support of the OT/E (Section 3.5).

3.1 METER in Perspective

The primary purpose of the OT/E conducted under this effort was to determine what useful capabilities METER provided to MAC analysts and this is discussed in Section 4. However, as background to the analysis discussed in Section 4, we briefly discuss intended METER capabilities.

METER was primarily intended to provide analysts with an analytical tool to support short-term and long-term message analysis. Messages received at MAC are mostly unformatted text with header information supplying the message source, date/time/group, routing, classification, and subject. Because of the dynamic nature of the all source message traffic received at MAC, analysts are posed with several basic information processing problems: (1) getting messages of interest when they are first received (message routing) and (2) locating single or highly correlated messages stored in a historical message data base as a means of identifying developing event trends.

Routing messages of interest to appropriate analysts is a task currently delegated to MAXI. MAXI allows analysts to develop interest message profiles

or templates that characterize what is of particular importance. To be effective, at a minimum, profiles must be frequently updated or at least reviewed to maximize the probability of receiving the right information. (Of slightly lesser concern is eliminating some of the information that is not important, i.e., the "noise".)

In theory, at least, analysts can use profiling to target messages related to re-occurring events or events having a cyclic behaviour as well as anomalous events that rarely occur or occur without a discernible pattern. Messages entering MAC and fitting the content pattern specified in a profile are then routed to predefined queues for analyst review. In practice, however, profiles tend to be updated infrequently resulting in a low signal-to-noise ratio (many messages of little interest to the user). This places heavy post-profiling time demands on users since they are required to sift through long (300-1000) message queues.

When messages "hit" on certain profiles they are tagged with a set of keywords associated with that profile. These keywords are used to index the messages for retrieval purposes. However, tagging messages for retrieval based on hitting or missing profiles is problematic since its not clear that the message is best indexed for general retrieval or even appropriately indexed for specific users sinced it may not have hit the appropriate profile(s). METER provides an alternative retrieval (and analysis) capability intended to augment the MAXI approach.

METER provides an automatic analysis capability that provides analysts with a tool for identifying subject areas or event trends in a message data base. In addition, with respect to profiling, it allows analysts to seek out messages that may not have been routed to them. Essentially, users can direct queries or information needs to a centralized message data base and browse through the data base by asking questions, and viewing retrieval summaries. In this manner, much of the reading that must be done by an analyst is reduced. When a message looks interesting in the summary, the analyst can

readily display it at his terminal.

The message data base is created automatically by METER. METER uses statistical techniques to dynamically determine which word roots or word stems to use to efficiently index messages for retrieval purposes. Analysts are not required to identify indexing word stems as they are when they employ profiles. Associations between pairs of stems are also derived to provide a dynamic automatic thesaurus capability.

In one sense, then, METER provides a capability for locating significantly correlated messages in the data base. The user can find messages that correlate or are highly associated with another message or with a description of an event found in a briefing paper or simply with a description typed in by the analyst while sitting at his work station. The system was designed to be easily used; since queries can be entered in English (by typing them in or supplying text from a message or paper) and METER processing is initiated with only three command keys (RETRIEVE, RESTRICT, and REPORT).

3.2 METER User's Interface

METER was originally designed to be a highly interactive information retrieval system allowing users to converge on sought-after data in an iterative fashion. The METER system designed for MAC was constrained by MAXI and OJ-389 terminal considerations. Primarily, METER became a block-oriented information system relying mostly on three basic commands and two forms. (The reader may refer to Appendix A for a more detailed description of the user's interface.)

METER users can approach information retrieval/analysis in several ways depending on personnal style; the primary mode of operation involves posing a general query, performing a retrieval, modifying the query and/or constraining the search domain, and retrying the retrieval until suitable results are

obtained. The basic goal was to allow the user to obtain usable retrievals as quickly as possible with minimum keyboard manipulation and minimum reading of message text.

Although METER is conceptually easy to use with only three primary command keys, it does require the user to have some knowledge of how to successively use the primary functions. For example, one basic approach is to first enter a query (either text from the keyboard or display some predefined text, such as a briefing paper on the left screen) and perform a general retrieval using the RETRIEVE function. Often this approach immediately provides useful results. The user may at this point show some of the messages. If one or more messages are found useful the user may wish to direct METER to find more like them by submitting them as a query. This is simply done by showing the message on the left screen and then depressing the RETRIEVE key. This approach makes the general retrieval function more powerful since it relieves from the user the burden of directly providing a highly specific and powerful query.

Generally the most precise approach to using METER involves using the RESTRICT function. RESTRICT allows the user to subset the data base using keywords from the text and/or header contents. Then if necessary, the user can perform a general retrieval over the subset. Clearly, the analyst can perform a general query and then impose RESTRICTions or he can use RESTRICT first and then pose a query. The important point is that RESTRICT and RETRIEVE work powerfully together.

From the user's perspective METER and MAXI provide a fairly simple mechanism for transferring information in either direction. In general, it is possible to transfer any piece of text from a MAXI queue or work file to METER using only two function keys (ESCAPE and METER). This text can be edited using the on-board editor or used directly as a query. Message text located by METER can be transferred to a MAXI workfile using the ESCAPE key and appropriate MAXI commands for storing text in work files.

3.3 MAXI/METER INTERFACE

The METER effort originally called for interfacing with the SSB system, which was to serve as the underlying support facility for analysts using OJ-389 work stations at MAC HQ. The implementation of the interface was largely completed when MAC/IN made the decision in early 1981 to replace SSB with a new system called MAXI. The switchover necessitated scrapping of the METER work for SSB and starting the design and implementation of a new METER interface to MAXI. This was hampered initially by unavailability of MAXI documentation per se.

After a several month hiatus, PAR Technology Corporation began making arrangements to subcontract with INCO incorporated of McClean, Virginia, for them to provide MAXI consultation and access to a MAXI development system at their facilities. Implementation of new top level METER retrieval modules for MAXI was begun in the summer of 1981 along with a MAXI workstation simulation facility to allow development and testing of the modules at PAR. Implementation of a METER subroutine library for communicating with MAXI through ICC began in the fall of 1981. Work was essentially complete by early 1982, with approval from AFIS in early Spring of 1982 to deliver METER software to MAC HQ.

The implementation approach to MAXI/METER was to separate METER as much as possible from MAXI. Communication with MAXI via ICC circumscribed transactions between MAXI and METER and also allowed METER run on a different processor. MAXI retained responsibilities, however, for putting up METER menus and required modification of its code for operation with METER at MAC HQ. These changes had to be validated by AFIS.

Further details on the retrieval modules and the ICC subroutine library may be found in the MAXI/METER Interface Description produced by PAR Technology to document modifications and additions to METER to support operations as a MAXI subsystem. The Interface Description complements the

METER Core System Description, and the two together comprise the documentation for the METER system at MAC HQ. Both the MAXI/METER Interface Description and the METER Core System Description are included as appendices in the Software Documentation package to be delivered along with this report.

A ICC subroutine library was developed primarily to permit as much as possible the use of existing code in developing retrieval modules for MAXI operation and also to insulate these modules as much as possible from details concerning ICC and MAXI. The subroutines were written in DEC PDP-11 assembly language since they consisted mostly of data buffering and the execution of RSX-11 system directives. Although the design of the subroutines was often complicated by subtleties in ICC and MAXI not always apparent in available documentation, the implementation was fairly straightforward. As it stands now, some revision is needed to improve their efficiency, but overall interface subroutines have worked satisfactorily for METER.

3.4 Setting up METER at MAC

At MAC HQ, METER was brought up on a DEC PDP-11/45 processor with the maximum 124K bytes of core memory (650 nanosecond cycle time) and with Bunker-Ramo 1536 dual disk drives. The PDP-11/45 was connected via a DA-11 interprocessor link to a DEC PDP-11/70 on which MAXI was running. All the analyst work stations at MAC HQ were attached to the MAXI processor. METER communicated with the work stations across the DA-11 link through a pseudo device DAO defined by ICC software.

MAXI had to be operated on the PDP-11/70 in order for analysts to use METER. METER was implemented at MAC as a MAXI subsystem available as an option on the top level MAXI escape menu. This allowed analysts to move information in text form freely between METER and other MAXI subsystems. MAXI status information could also remain visible on the right screen of a work station even when an analyst was executing METER commands.

A single Bunker-Ramo 1536 disk drive was made available for METER, mounting a system pack doubling also as the METER data base pack. The formatted capacity of a pack was about 170,000 blocks of 512 bytes, of which about 130,000 was available for a METER data base after subtracting out space required for an IAS operating system with associated tasks, ICC software, and METER programs.

The METER system was configured initially at MAC with a window of 14 update batches. Input to METER was expected to run about 1,000 messages per day. These would be tapped in MAXI as they entered its MSS component and would accumulate on the MAXI message disk pack until a system operator executed the METER QGATES task on the MAXI PDP-11/70 to move the messages over to the METER PDP-11/45. This transfer was scheduled every two hours throughout all three shifts of MAC/INY to conserve space on MAXI message disk pack. A full METER update was scheduled beginning at about 0500 each morning.

Although the METER preprocessor program sequence (QSTM followed by QNVRT) could have been run throughout the day as messages were transferred from MAXI, the practice of MAC/INY was to invoke it once immediately before the daily invocation of the full update program sequence. With an update batch of a 1,000 messages, the selection of 3,500 index stems, and a total of 14,000 messages in the METER data base, the preprocessor and the full update program sequences would each take a little over an hour to complete. Printing out data base statistics took another fifteen minutes after that. User access to METER was disabled during all this time.

With the scheduling as described, an analyst at 0800 each morning would be able to retrieve messages from the preceding two weeks of traffic up to about 0500 of that morning. With the METER short update facility, it would have also been possible throughout the day to make available messages that came after 0500; but MAC/INY was reluctant to do this because it would have required running the preprocessor during the day shift when users were most likely to use METER.

With the IAS operating system and ICC software on the PDP-11/45, only 64K words were left for METER. This meant that METER could be running for only two users simultaneously. With the preprocessor or full update sequence executing, only one user at a time could get into main memory, although because of the MCR command file processor bringing successive programs into different areas of main memory, all users would be locked out at times by fragmentation of free memory into areas too small to load retrieval programs. This and the desire to complete updates quickly with a minimum of interference by other programs led to segregation scheduling of update and retrieval processes.

The limitation to only two users simultaneously running programs turned out not to be too serious. When an analyst is at a MAXT work station, most of the time is normally spent in reading and editing text, filling out forms, and entering commands; these actions can be processed locally at a OJ-389 workstation and do not require any program to be running. Furthermore, when a METER retrieval program or any other MAXI program does run, it generally does not run long. The METER update programs run by system operators on the other hand do run long and tie up memory for extended periods of time.

After establishing a reasonable schedule of operation, the next step was to set up various METER tables to support processing. These included the METER users table listing the 9-digit MAXI ID's of analysts authorized to use METER, a table of the source names to be encoded for message header restrictions, and a table of "stopwords" to be disregarded in METER statistics. The last is especially important with a limited number of index stems because it helps to prevent certain words from being considered when we know about them beforehand.

The tables took several days to prepare by MAC/INY with assistance from PAR Technology. The identification of stopwords took the longest since this involved running METER on small sample batches of messages, examining the index stems derived, and entering new stopwords as appropriate. This process

was repeated later several times after METER went fully operational. It was easy to miss things at the start, and additions had to made to various tables from time to time.

With the data base parameters set in the METER control file and METER tables in a satisfactory initial state, a switch was set in MAXI to feed messages to METER and METER was started up on its planned scheduled. Users were still not permitted to use the system yet in order to allow a sizable data base of messages to accumulate, to give system operators some experience with the system and its scheduling, and to catch any problems in the update procedures.

3.5 METER Training

As part of the OT/E, PAR Technology developed a training program for each type of MAC user: the analyst, the system manager and the METER operator. In each case, training was supported by brief overviews or presentations, hands-on training with a PAR representative and appropriate documentation and manuals. (The METER User's Guide is presented as Appendix A, the Operator's Procedure Manual as Appendix B, and the System Manager's Guide is provided as Appendix C.) In this Section, each of these training programs is overviewed.

3.5.1 Analyst Training

Analyst training was setup to include three phases:

1. An overview which outlined the goals and timetable of the OT/E and which discussed various aspects of the training program. During the overview prospective users were given a brief description of what METER would provide them, the basic function keys they would use and the METER forms. Users were also given a synopsis and a copy of METER training materials at this time. PAR was prepared to conduct this overview as often as necessary to insure that potential users had some context for the

evaluation and further METER training that would follow. Unfortunately, only about ten MAC personnel could attend the session and it was not possible to schedule another session for those who had missed it.

2. The second stage of training involved hands-on use of the system. The active METER users group consisted of about 21 persons. Of the 21 users, thirteen MAC/IN personnel received a terminal training session. Of the thirteen, four personnel had received the METER overview. In addition, three of the thirteen received their training within the last two days of the evaluation.

   The training sessions generally lasted about one hour and included between 1 and 3 persons. First, each trainee was presented with a METER Users Manual and a brief description of what it contained. Next, each user was provided with a short overview of system functions. This was followed by practical use of the system. For example, each user was asked if there was an information need they were currently trying to satisfy via the message traffic. In some cases this involved looking for a message relating to some well defined events of interest to the user. In other cases, analysts presented the contents of briefing papers as representative of the type of messages they would like to locate. The METER trainer then illustrated how METER worked in the context of solving the identified real-world problems supplied by the analyst.

   In some cases, users were not able to provide actual problems to solve using METER. In those cases, training personnel used standard training scenarios illustrating how METER supported several generic information retrieval tasks. For example, one scenario consisted of hypothesizing that an important message had been identified in a MAXI message queue. This message may have been related to air activity of an adversary in a specific geographic location. METER was then used to identify previously received messages from a specific source and time period relating to the key message. The general conclusion of most analysts trained was that

the on-line sessions were most instructional and should be provided to them on a as-needed, follow-up basis.

3. The third stage of training consisted of follow-up sessions in which users could present any problems they had encountered in using METER or discuss how METER was or could be used to analyze message traffic. Only one analyst was known to have participated in all three training phases. This was partly due to the schedule of personnel, system availability, and other operational factors, including the time available to conduct the training and evaluation of METER.

3.5.2 System Operator Training

The METER system operator at MAC was responsible for starting up METER on a processor, linking up with MAXI, initialing METER updates, taking care of files on the METER disk pack, and restoring operations in some kind of problem disrupted METER processing. All pertinent procedures here were collected and categorized in the METER Operator's Procedures Manual. In most cases, procedures were implemented as MCR command files.

System operator training at MAC consisted mainly of walking through a couple of full updates with PAR Technology personnel. This was aimed at showing how various parts of the Procedures Manual worked out in actual practice, allowing the operator to become familiar with the system and with the Procedures Manual at the same time. Operators were also instructed in how to edit the METER system tables, add new users to interpret and modify the METER data base control file, interpret directory listings of data base files, and to restart interrupted processes. INCO personnel collaborated here in showing how to turn on the collecting of messages by MAXI for METER and establish an ICC link between the PDP-11/70 and the PDP-11/45.

Two MAC/INY system operators received this training, from PAR Technology and they in turn instructed other operators. The arrangement was adequate for routine operations, but serious problems arose, however, with recovering from errors. Here some operators, knowing only roughly how METER worked, were often unable to diagnose the error properly in order to get the appropriate recovery procedure. The situation was further aggravated by errors encountered within METER software, including the recovery procedures themselves. This necessitated extensive long distance telephone consultation with PAR Technology on several occasions when serious errors interrupted processing and no PAR Technology personnel were on site at MAC.

On the whole, experience at MAC HQ would dictate making fundamental changes in the training of METER system operators.

- Although it is unrealistic to expect system operators to become experts on METER, they need to know, nevertheless, some basics about the theory and the mechanics of the system in order to be able to put procedures in practice at the right time. Operators need at least to be aware of the kinds of safety checks built into METER and to be able to assess the seriousness of error conditions. In particular, the operator should be able to distinguish problems that respectively relate to hardware, to the operating system, to ICC and other MAXI software, and to METER itself.

- Error recovery procedures should be greatly simplified. Fairly elaborate recovery procedures were initially defined for METER in order to save as much data as possible after an error, but it now appears that time may be more important than data. In an operational environment, it is better to lose some data if this gets a system back up more quickly. Elaborate recovery procedures also introduce more opportunities to make new errors and may turn out in the end to be counterproductive.

- Operators should be able to make periodic integrity checks on METER data base files even though no obvious error has occurred. This would involve the development of one or two new software tools to aid the operator. Integrity checks should help an operator identify anomalous files left around as a result of error and to determine when a data base might have to be reconstructed.

The organization of training programs for system operators remains something of a problem in itself. At MAC HQ, the scheduling of training sessions was complicated by changing work shifts, temporary duty assignments, and leaves. Some personnel responsible in part for METER operation were unable to receive formal training during the weeks set aside for it. In the longer term of course there is also personnel turnover to take into account.

Given this situation, there needs to be a special program to train at least one system operator to train other operators. There was such an operator in effect for MAC/INY, but the arrangement should be made more formal and should be supported with special training materials. This should be done even if there is on site contract personnel to maintain METER.

3.5.3 System Manager Training

A METER system manager has the responsibility for configuring a system for a particular environment, scheduling updates, assisting users, and taking care of any problems that might arise in processing. A full description of the various duties involved here is provided in the METER System Manager's Guide. In MAC/INY, the system manager's job was held jointly by two persons: one in charge of actual operations and one acting as liaison to users.

Training for METER system managers was set up to provide a broad view of METER:

- Setting up system tables to tailor METER to MAC, including special stopword and source name tables.
- Basic METER file management, data base window size.
- Effects of system parameters (in the METER control file) on expected processing time.
- Scheduling criteria.
- System summaries.
- The analyst's view of the system; analyst training.
- Backup and recovery of data files.

Limited time prevented covering all these areas completely with both the persons to be trained, but every item was gone over at least with one of them. The approach was to have PAR Technology personnel walk through each item with MAC/INY staff, while making recommendations along the way. This in effect was on-the-job training with PAR personnel initially acting as system manager, but gradually relinquishing responsibilities to the MAC/INY staff.

The training procedure worked adequately at least for the short duration of the evaluation period. MAC/INY quickly begin to set up its own update and backup schedules in line with expected utilization of METER at MAC HQ, and even worked out special recovery procedures on one occasion when there was an error in METER software. For extended METER operation, however, more formal and more comprehensive training seems called for; major changes would include

the following:

- A more detailed orientation of overall METER system architecture supported by prepared aids. This would aid identification of trouble in the system.

- More emphasis on the METER short update and the null update that only drops older messages from a data base.

- Providing more specific bases for generating summary reports, along with criteria for selecting the parameters for such reports.

- Clearer separation of operational and user liaison duties.

These would require more training time than was available at MAC.

4. The METER User-based Evaluation

4.1 Evaluation Design

Evaluation of METER (or any other information system) in an operational environment is difficult since there is practically no way to test it under fully controlled conditions. In particular, the METER evaluation is characterized by uncontrollable factors, such as a dynamically changing data base, a variety of analyst information needs, and the availability of users. Therefore, the main emphasis in the METER evaluation design was to provide a framework in which each user could be trained, observed, and assisted by PAR evaluators in real-world, problem-solving situations. From these "experiments" or field tests, information was collected as to how well MAC analysts can effectively exploit message traffic using METER. It is noted, however, that the short evaluation time has somewhat limited the number of generalized performance assessments that can be made.

PAR utilized several approaches to collecting evaluation data. The online instrumentation package, which is transparent to the user, collected information as to the commands employed by users as well as some retrieval performance statistics. With sufficient data, a quantitative user profile can be used to corroborate qualitative assessments derived from user interviews. In addition, each METER user was given a questionnaire (Appendix D) to complete prior to the end of the contract. Data aggregated from the instrumentation package, the questionnaire, and the interviews provides the basis for the METER evaluation.

4.2 Model of the User

During PAR's involvement at MAC, efforts were taken to identify the basic message processing tools required to support MAC/IN analysts. The result of this effort was to identify a basic MAC/IN information flow model and to

determine from that model if and/or how METER functions could be used as an analytical message processing tool. The qualitative model is not overly complex and essentially identifies the basic information flow and decision making processes relevant to METER functions.

In general, the model statements make explicit the existence of general requirements for message handling tools (such as profiling and routing functions) as well as analytical tools for identifying events or event trends in the message flow.

The Basic Information Flow Model

Incoming messages, from multiple sources, are reviewed for transmission-correctness on the message receiving system. Messages are then routed to personnel who are primarily interested in immediate or short-term situations. They are alerted to potential or current situations of interest based on "indicators" which are primarily qualitative descriptions of organizational interests. The short-term assessments produced may be immediately routed to various high-level MAC decision makers and/or routed to area specialists for further analysis.

Area specialists provide background support to short-term analysis by determinining if a potential crisis really exists or by providing further historical information relating to a situation of interest. This may also result in explaining why an apparent crisis is not actually a crisis.

Various high-level, decision makers are presented with both the long- and short-term situation assessments provided through analysis. These assessments may be presented as part of daily or weekly briefings and they may be included in messages for dissemination to the Intelligence Community. Assessments are generally "referenced" to specific messages received over time. Therefore,

analysts must be able to readily extract individual or groups of messages from an historical data base.

General Message Processing Tasks

The message review task requires that analysts actively review incoming message traffic to provide immediate assessments in response to rapidly changing events and to perform extensive background studies. This review task can be problematic since it requires analysts to generally perform correlation over time and subject area to produce accurate event assessments. In addition, there is an inherent "routing" problem which can result in analysts never seeing messages of potential interest or, less critically, it can result in analyst reviewing many extraneous messages.

To facilitate handling of messages, the MAXI system is primarily utilized by analysts to assist in the message review process. MAXI provides a profiling capability allowing analysts to template interest areas. Messages that "hit" profiles are tagged or indexed as well as sent to appropriate analyst message queues for review. The routing problem is only partly addressed by MAXI since profiles may not always provide hits on messages that the analyst requires. Also, profiles must be dynamically updated to be useful. MAXI also provides analysts with a text processing capability to facilitate generation of briefing papers, messages for dissemination, and other report-generation-like functions.

The METER system provides a message retrieval and analysis capability allowing analysts to identify specific messages or groups of messages within an historical message data base. For example, groups of relevant messages may be extracted by users based on content correlation and/or dissemination parameters, such as time of occurrence, source, and security classification. Therefore, METER directly addresses the message-correlation-over-time-and-

subject-matter requirement that is a primary analytical task performed by analysts.

Model of METER Usage

Analysts may take several approaches to locating "correlated" messages. For example, if an analyst is reviewing his MAXI message queue, and finds a message of interest, he may "take" that message to METER and direct METER to find messages highly related to it. Another approach may find an analyst writing a briefing paper with one part of the paper being some situation assessment for which the analyst requires supporting messages from the data base. The analyst locates desired messages by giving the appropriate part of his briefing paper to METER and directing the system to locate messages associated with it. Analysts can also directly enter a METER query to obtain useful information.

Analysts are often required to locate very specific messages or message clusters that may have been received from specific sources at isolated times. METER has been used to provide this type of focused search by providing analysts with a restricted-analysis-mode. In this mode, analysts can freely specify the search domain without being affected by predefined message classifications or indexing. The analyst can also change the search domain dynamically as he explores the historical data base.

Most short-term assessments are generally performed using messages received over the last 1-3 days; while long-term assessments or background studies generally require about 30-45 days data to be available online. For the METER evaluation, only about 10-12 days of messages were available because of disk storage. (Storage space requirements for METER are discussed in Section 5.)

Once messages have been located using METER they can be easily copied back to MAXI for storage in work files. The work files may be used to store briefing papers, outgoing messages, working papers, etc. With the METER/MAXI system analysts can more effectively review and analyze message traffic to produce required intelligence assessments.

4.2.1 METER User Profiles Via Instrumented Data Collection

The METER (online) Instrumentation Package provides a METER usage data collection capability. Each METER user was assigned a METER evaluation file on the METER processor and whenever METER was used, the command, command parameters, and the time of use were stored in the appropriate user's file. At the end of the evaluation, summary statistics were computed describing percentage usage for each METER command as well as other performance statistics.

Figure 4-1, provides a summary of METER usage by command across available users. Although more data would be desireable to identify a detailed model of METER users, a discernible usage pattern is still evident. The high count and overall percentage usage of BUILD, GET and SHOW indicate that users typically utilized the basic METER retrieval mechanism. This implies that the basic retrieval function was probably sufficient for general information searches, at least, but it may also point to the need for more training in use of other METER functions such as RESTRICT.

From examination of the SHOW function usage, it is evident that the number of messages displayed (for reading) varied significantly across users. The average number of messages displayed per retrieval, across all users, is 5.52 with a standard deviation of 3.52. This indicates significant user variation in the requirement to display raw message text. This variation is significant when considering that the number of messages displayed was limited to between 0 and 14 during the evaluation. (Note that retrieval summaries presented users with the top most "important" 14 messages from a retrieval.)

There may be many reasons for this variation but disucussions with analysts substantiated the assessment that some analysts felt it necessary to read extensively either because many retrieved messages were thought to be relevant to the query or because they liked to browse and look for other interesting messages. Other analysts more actively used the METER retrieval summary to minimize reading.

Further examination of the data verifies that users typically found other ways beside culling retrievals (CULL) to get better retrieved messages. For example, they may have submitted messages, or other prepared text, as queries to give METER more detailed descriptions of desired results.

Figure 4-1: METER USER PROFILE

	Individual User Number											Cumulative Frequency Statistics	
Function	3	4	11	22	24	32	33	34	37	40	44	total	%
BUILD	7	19	22	9	10	8	9	2	4	3	5	98	16.3
COLLECT	0	1	3	0	4	0	0	0	0	0	2	10	1.7
CULL	2	0	2	1	1	2	1	0	1	0	0	10	1.7
SHOW	10	78	18	40	115	32	31	14	34	4	17	393	65.3
FULL	0	1	1	0	0	0	0	0	0	0	0	2	0.0
GET	9	16	22	7	9	8	9	2	4	1	2	89	14.8
HELP	0	0	0	0	0	0	0	0	0	0	0	0	0.0
Cumulative												602	~100%

The use of COLLECT is rather low, and this implies that the RESTRICT function may not have been used extensively. This data, we believe, reflects some user uncertainty in how to employ RESTRICT with the general query capability to get more useful retrievals. Generally, however, PAR/Analyst user sessions in which the use of RESTRICT was stressed, indicated that analysts thought RESTRICT extremely useful and easy to use. The conclusion is that if more time was available for training, analysts would have become well versed in RESTRICT usage and the frequency of use would be significantly

higher.

4.3 METER Response Times

System response time is one of the basic concerns of the user and one of the METER objectives was to determine METER retrieval times across various conditions. The Instrumentation Package provided a basis for recording METER response time for users during each session. Although it was not possible to also record system loading it is assumed that across the evaluation a typical cross-section of system load conditions was encountered. Therefore, the variation in system response is a reflection of the variation in system loads.

Figure 4-2, provides a summary of retrieval times across all users. Generally, METER users received messages in less than thirty seconds. Clearly, retrieval times can vary depending on system loading and operation but based on the response time standard deviation of about 8 seconds, it seems that system performance is relatively stable across users and system conditions. If the response times were assumed to be approximately normally distributed, with a mean of about 23 seconds, we see that there is about a 90% probability that a response time would be less than 40 seconds. The maximum and minimum response times do indicate performance extremes encountered during the evaluation; but since they differ significantly from average performance and are well beyond 2 standard deviations from the mean it is clear that they are performance anomalies or extremes, rarely occurring.

Figure 4-2: METER RETRIEVAL TIME SUMMARY

Average Response Time	22.54 seconds
Standard Deviation	7.86 seconds
Maximum Response Time	76.86 seconds
Minimum Response Time	2.68 seconds

Finally, discussions with analysts indicate to PAR that retrieval time is not necessarily a significant performance statistic since for most analysts, it is the total time to perform the information search that counts most. Generally, METER satisfied most analyst requests in approximately 10 minutes or less; and this included transferring initial query text from MAXI, performing general retrievals, and/or restricted retrievals in an iterative fashion and forwarding the retrieval to MAXI for storage. In many cases the overall time to successfully complete an information search task was judged to be quite acceptable by users and the retrieval times, above, were only small contributors to the overall time.

4.4 A Qualitative Assessment of METER

Interaction between PAR evaluators and MAC analysts extended the basis for determining the usefulness of METER under operational conditions. This input to the evaluation is based on interviews and work sessions with over 20 MAC personnel.

4.4.1 The User Group

The group of "active" METER users crossed several areas of MAC/IN. Primarily, they represented INO, INW, INA and CSG. Although individuals in these groups differed somewhat in their data requirements, their general analytical requirements were quite similar, For example, in Figure 4-3a, a brief user sketch or general user profile is presented. Note that the "typical" MAC METEP user has been on the job about 16 months, has some responsibility in the preparation or presentation of briefings, and tends to be a frequent MAXI user. In particular, the typical user employs the MAXI BUILD/SEND, TEXT PROCESSING, and RETRIEVAL capabilities. The text processing functions primarily refer to the onboard editor and the MAXI workfile functions. It is noted that most users felt that an enhanced retrieval (i.e., analysis) capability would be useful.

FIGURE 4-3a: USER PROFILE-GENERAL BACKGROUND

USER	JOB TIME	INTEL BRIEFINGS	MAXI USAGE	REVIEW	MAXI "FUNCTIONS" USED BUILD/SEND	TEXT-PROC	RETRIEVAL	RETRIEVAL ACCEPTABLE?
A	8 mos	NO	DAILY	YES	YES	YES	YES	NO
B	48	NO	DAILY	NO	YES	YES	YES	NO
C	7	WEEKLY	DAILY	NO	YES	YES	YES	YES
D	24	FREQUENT	2HRS/Day	NO	YES	YES	YES	NO
E	24	DAILY	DAILY	YES	YES	NO	YES	YES
F	6	SUPPORTS	DAILY	NO	YES	YES	YES	YES
G	3	NO	DAILY	YES	YES	YES	YES	NO
H	24	NO	2-3/Wk	NO	YES	NO	YES	YES
I	3	YES	2-3/Day	YES	YES	YES	YES	NO

The data in Figure 4-3b, provides an overview of user information requirements. Again, the "typical" MAC user reviews about 260 messages per day with a large variance among individuals. In addition, the user generally performs most analysis using messages from the last 3 days however, when background searches are performed, 30 or more days are generally required. (Note that the data in Figures 4-3a and 4-3b were compiled only from questionnaires, however, users interviewed, who did not complete a questionnaire, did not significantly vary from the responses, above. Further, the notion of a "typical" MAC user characterized, above, should be viewed in light of the limited exposure afforded PAR to the total population of potential METER users. The characterization does not attempt to weight any particular user differently from another; and it is recognized that METER may be more applicable to one particular user group at MAC.)

FIGURE 4-3b: USER PROFILE-DATA REQUIREMENTS

USER	# of MSGS REVIEWED	NORMAL TIME FRAME	ONLINE HISTORY REQUIRED	BACKGROUND DEPTH
A	15	1-3 days	30 days	7 days
B	800-1500	1-30	30	30
C	200	UNKNOWN	30	NO RESPONSE
D	300+	1	30	30
E	250-350	1	30	14
F	60-75	1-3	30	30
G	300	1-7	7	VARIABLE
H	30-50	1	10	14
I	90	30-60	60	EXTENSIVE

4.4.2 Special Analyst Work Sessions

Real-world problem solving sessions at a terminal were conducted by PAR evaluators with several MAC/IN users. Several types of information needs were addressed. For example, in one case the general objective was to identify all messages related to a particular aircraft and involving particular countries. The problem was further complicated by desiring that only air threats or engagements be retrieved. METER provided basic tools for identifying messages associated with the required aircraft and countries but it was not completely efficient to either use the current METER RESTRICT function or CULL function to limit retrievals to only threat or engagement messages. It was learned however that the nonthreat messages could be elimited from retrieval by filtering out all message from a particular source. PAR made a small change in the RESTRICT function software and within 24 hours the new capability to filter out sources was available to users. With the new change the user could locate and focus in on those desired threat or engagement messages in just a few minutes.

Another analyst received a message related to a serious incident involving several countries and planes in a specific area of the world. As far as he was aware there were not any messages in the data base that may have

forwarned of a potential crisis in that area. His immediate interest was in what forces were operating in that area. He quickly entered a general METER query in English and in about thirty seconds received several messages relating to force movements that were of direct interest to him. Using RESTRICT to perform a more focused search he found the force movements as well as reports of air activity in that area involving the same aircraft and countries involved in the newly forming "crisis". The total time to obtain the messages using METER was less than 7 minutes.

As the final example, an analyst was attempting to locate messages relating to the status of several airfields in a particular region of the world. Using available methods he needed several hours to get what he needed. However, during a METER work session he was able to get the same information in about ten minutes. In general, METER provided users the ability to perform broad as well as focused information searches across a rapidly changing message data base. METER allowed analysts to change their focus dynamically, and vary their view on the data base. In addition, analysts needed to utilize only three METER functions and could input queries in natural language or free text.

4.4.3 User Perceptions of METER's Utility

Through the structured work sessions, other interviews and response to the questionnaire, user perceptions of the utility of METER at MAC were obtained. For example, the data in Figure 4-4 indicates the commands that users found most useful in METER.

Figure 4-4: METER Command Utility

Command	Overall rating	
	1-----2-----3-----4-----5	
HELP	xxxxxxxxxx	(2.4)
RETRIEVE	xxxxxxxxxxxxxxxxx	(3.8)
HEADER RESTRICTION	xxxxxxxxxxxxxx	(3.3)
KEYWORD RESTRICTION	xxxxxxxxxxxxxx	(3.3)
SHOW	xxxxxxxxxxxx	(3.0)
CULL	xxxxxxxxxxxxx	(3.1)
METER/MAXI Information Interface	xxxxxxxxxxxxx	(3.1)
Offline Statistics	xxxxxxxxxx	(2.4)
	1-----2-----3-----4-----5	

Key To Ratings:		
	<2	Not Favorable
	2	Not Sure (Neutral)
	3	Somewhat Useful
	4	Very Useful
	5	Extremely Useful

Clearly, users felt that the RETRIEVE, HEADER and KEYWORD RESTRICTION functions were quite useful, and these functions comprise the basic METER analysis and retrieval subsystems. The SHOW function was also thought to be useful; as was CULL. However, during interviews it was certain that analysts appreciated the essential function that SHOW fulfills in providing a message display capability. The CULL function was thought to be somewhat useful although most analysts did not frequently use it. The ability to transfer messages (text) from METER to MAXI and vice versa was also thought to be quite useful. In general, most users did not feel that the HELP or offline statistics were essential. In the case of HELP, this may reflect the ease with which METER can be learned as well as the space limitations imposed on METER HELP storage. The offline statistics were made available on a daily basis at various locations, but were seldom looked at.

The data in Figure 4-5 present some user viewpoints as gleaned from the 9 questionnaire responders (7 no responses). Examination of questions 1, 2, and 3 indicate that nearly all users thought METER easy to use and understand.

When asked more explicitly about METER reports (questions 4 and 5) most users thought system outputs were easy to interpret but several had trouble reading the bar chart correctly (for estimated relevance) and getting an idea of message content from the top four major stems. The lack of total acceptance of the bar chart (relevance histogram) as a useful display of message correlation with a query is consistent with some earlier findings from PAR. The primary use of the histogram is to indicate roughly how closely messages match with the query and should not be overemphasized. PAR's view is that the stems are more useful in deciding which message to read as well as what the message content is. Other discussions with users (outside of this questionnaire) indicate that the stems are generally useful; one suggestion is to increase the number of stems from the top 4 to the top 6.

Figure 4-5: User Views Of Meter

Question	YES	NO	No Response
1. Were METER commands easy to use?	9	0	0
2. Did you understand METER output?	8	1	0
3. Was the retrieval summary understandable?	6	0	3
4. Was the bar chart useful?	5	3	1
5. Did the stems provide a good overview of a messages content?	5	3	1
6. Were retrievals generally good?	6	2	1
7. Was response time too slow?	4	3	2
8. Would you use METER in a crisis?	7	1	1
9. Should METER be kept at MAC?	8*	1	0

* Yes METER should be kept after some modification.

Most users felt that retrieved messages were relevant to their queries. In fact several follow-up interviews indicated that some incidences of poor retrieval were associated with software problems during the data base update process which were subsequently corrected. A few other cases of "poor" results were associated with data base size limitations: users did not know that the data was not in the data base.

Several users thought that response time was too slow. Most users tended to associate response time to the time to get some reply from the system and not the time to perform an operation or job task. What they were asking for, we believe, was faster human/machine interaction at a low level; that is, when a user depresses a key, the system responds in about ten seconds or less with some type of display. This aspect of METER performance can be improved, as is discussed in Section 5 of this report.

Most analysts felt confident enough in METER capabilities to employ it during a crisis. This implies that METER supports analytical functions fairly accurately, and that, at the task level at least, it performs in a timely manner. Nearly all questionnaire responders would like METER to be operationally available at MAC provided that some additional basic tailoring is accomplished. Some of these users would employ METER on a daily basis while others would employ it only at key times.

5. METER Software Assessment

In this Section, we discuss the overall performance of the METER system from a software maintenance/reliability perspective. This assessment reflects not only the inherent reliability of METER code but also various operational factors, such as available disk space, that impacted on system performance.

5.1 System Reliability

At MAC HQ, METER ran as would be expected when a system first moves from a laboratory environment into a real operational environment. During the first month of evaluation, numerous software problems were uncovered, the most serious being

- incorrect handing of message data/time groups and source names in several different modules.
- incorrect counting of messages because of vector terminators allowed to cross disk block boundaries.
- bad upper bounds on frequency being set in index stem selection.
- incorrect batch numbers computed in the procedure to drop a batch of messages from a collection.
- misformatting of output records in the dumping of the METER users table, making it impossible to read them back in.

These were all eventually taken care of, but their occurrence resulted in METER unfortunately being unusable for periods as long as three or four days at a time in the first month. This would indicate that the time available for the initial shakedown period was too short. No other major METER software

problems were encountered in the second month of evaluation, though.

The greatest difficulty in running METER throughout the evaluation was with operator procedures for recovery from an error. The procedures as originally defined were too complicated; the operators at MAC found them hard to understand and often could make mistakes in carrying them out. On the whole, attempts at error recovery more often than not aggravated a situation and extended down time. We therefore found it necessary to institute a highly simplified recovery procedure. In effect,

1. First, log in again, unlock the control file, and retry the program where the error occurred.

2. If the preceding fails, delete everything and start over from scratch with backed up batches of input messages as available.

3. Under no circumstances should the operator stop a METER command file before it runs to completion; nor should the operator ever edit a control file for error recovery.

In retrospect, the above procedure implemented at the start of the evaluation period would have saved operators at MAC a great deal of time and grief and would have reduced overall down time. The new version of the METER operator's Procedures Manual incorporates this procedure. From our experience at MAC HQ, we conclude that recovery procedures in general cannot assume a highly knowledgeable system operator. Although two persons at MAC were trained by PAR to start up METER and to run updates, rotation of work shifts and temporary duty assignments and leaves meant that problems would often arise without a thoroughly trained operator around.

During the last two weeks of the evaluation, METER remained operational without serious down time. Several problems did arise because of an operating system disk I/O problem that led to a crash, sporadic events where the UIC of

a terminal would change unexpectedly during execution of METER command flies, and one operator error. For all these cases, METER processing was successfully resumed with only a short delay. Although there is little data yet on which to predict METER reliability, the system does in fact seem to have stabilized enough to run without trouble for at least one or two weeks at a time and to recover from various problems without requiring a heroic effort by system operators.

Some program bugs probably still exist in METER software. The system as a whole, however, has benefited from operational use at MAC in that one can now have greater confidence in METER being able to work with collections of ten thousand or more messages. Software for the MAXI/METER interface also received extensive testing during the evaluation period and proved to be almost trouble-free; at no time did MAXI operation appear to be adversely affected by METER other than having MAXI disk storage tied up to save messages for transfer to METER. METER communication with analyst work stations through ICC worked without problems, although it probably could have been implemented to run much faster.

Any changes to METER will clearly have an impact on reliability. Experience at MAC,however, has shown that adaptations are possibly for a particular METER application without a basic core system change too. This METER core system in effect consists of those modules that define the heart of any METER facility. It generally contains the most mature code in METER and currently accounts for about eighty percent of all code in METER. If modifications to the core system are carefully controlled, then software changes should not greatly hurt the overall reliability of METER.

An overall summary of METER availability during the evaluation period at MAC HQ is as follows:

Availability	945 hours
Updates, users locked out	179
METER backups, file clean up	24
METER software problems	137
MAXI down when METER could have run	141
Processor down for periodic preventive maintenance	36
System crashes, hardware or software	34
Weather or power related shut downs	24
Non-METER system development requiring processor	45
TOTAL	1565 hours

(We thank Sgt. Wayne Cox, MAC/INY, for compiling the above figures from system logs)

Altogether, METER was available to users about 60% of the time during the period of evaluation, though it should be noted that down time tended to be in long stretches. (Note that METER was available about 70% of the time excluding MAXI downtime, weather-related interruptions, and non-METER development time.)

5.2 An Assessment of METER Operation

The performance of METER at MAC HQ appeared to be adequate for the purpose of an evaluation, but the slow speed of processing and the limited data base size definitely reduced flexibility in its operation. In the preprocessor component, the QSTM program required about 3 seconds to process each incoming messages, on almost an hour for 1,000 messages. On a DEC VAX-11/780 under compatibility mode, QSTM takes about 0.7 seconds per message; since the QSTM program is partly I/O and partly computation bound, we expected that processing on a PDP-11/45 as configured at MAX HQ would be on the order of 2 seconds per message.

In the full update sequence, the message indexing program PCMV at MAC took about 0.2 seconds per message. The PCMV program, however, has to go through every message in a data base; for 10,000 messages, this would amount to over half an hour. This seems fairly reasonable nevertheless because PCMV on the VAX-11/780 typically would take about .06 seconds per message. In any event, PCMV execution time turned out to be the dominant contribution to total update time, though not to the degree that QSTM dominates the preprocessor execution time.

Surprisingly, the computation of first-order stem associations in the full update turned out to be a relatively minor contribution to total full update time. This was due in part to an inner product computation program designed to be more efficient for the relatively low volume of messages that results in fewer stems being significantly associated. Retrievals did not appear to be affected much by this.

A greater problem than processing time was space. According to actual requirements at MAC HQ, every message requires about 8 disk blocks of storage in METER data base files. A projected total of 14,000 documents in data base of 14 daily updates would therefore fill up available space on the METER system pack assuming that about 20,000 free blocks would always have to be available for temporary storage to support update processing. MAC/IN originally hoped to be able to accommodate up to 21 daily updates altogether; and at least two analysts said they would like METER to retain messages for longer than two weeks. Unfortunately an extra 7 days with about 1,000 messages per day would result in a data base that would fill up almost a Bunker-Ramo 1536 disk pack just by itself.

The space problem was made worse by events in the Falkland Islands and in Lebanon leading to a major increase in message traffic at MAC HQ during the METER evaluation period. At times, daily traffic exceeded 1,500 messages, filling up disk space faster than expected. The data base window size had to be reduced first to 12 updates and then to 10 updates. On each occasion,

there were several days of down time because of METER software errors (since corrected) in code to reorganize data base statistics to reflect dropping of an update.

The frequent readjustment of data window size was not greatly emphasized in the METER System Manager's Guide at the Operator's Procedures Manual, but now seems to be important in order to provide analysts with as many messages as possible. This would involve fairly close monitoring of message traffic and resetting control file parameters as necessary. Toward the end of the evaluation period, daily message traffic decreased markedly resulting in a data base of only about 8000 messages over 10 updates.

6. METER Enhancements

Extended Operation of METER at MAC HQ on a full-time basis pointed out many possible improvements in the areas of system processing speed and facilities to support special needs of analysts at MAC. Where such improvements were relatively easy to implement, they were incorporated straightway into the operational system in order to encourage more use of METER during its evaluation period. Our philosophy here was to focus on evaluating the overall usefulness of a METER capability in a MAC environment and not particular system details that could readily be changed. The only modifications avoided were those involving many different modules and thus possibly affecting the reliability of METER.

The original METER effort had called for a separate enhancement phase to follow the evaluation phase. The work delay arising from the switch by MAC from SSB to MAXI analyst Support software, however, forced the enhancement phase to be telescoped. As it turned out, several major improvements to METER could not be accommodated under this new arrangement. We will note here first the changes actually made on the MAC operational system (the "Advanced Engineering Model") during evaluation and then those changes currently identified but not yet realized at MAC.

6.1 IMPLEMENTED CHANGES

The relatively slow speed of DEC PDP-11/45 processors prompted periodic examination of METER critical path processing to see where operations both before and after installation at MAC HQ could be significantly streamlined.

The five most important areas in this regard were:

- The processing of individual words in the innermost loop of the preprocessor.

- The production of inverted lists for stems, a sorting problem.

- The stem vector inner product computation.

- The indexing of messages by stems.

- The reduction of file I/O in retrieval processing.

Among the major changes actually implemented were the following:

- A denser packing of data in METER files to reduce the total number of blocks to be processed. This package included conversion of stems to a RADIX-50 storage format and compression of sparse vector data to put coordinate and value pairs into two bytes. The latter took advantage of the fact that the volume of message traffic at MAC is only about a thousand per day and allowed for greatly speeding up inner product computations and retrieval searches.

- A new assembly language version of the METER stopword recognition procedure, which is called in processing every word of an input text. This version was also motivated by the need for a more efficient way of filtering out special item designators in AUTOPIN messages having the form of several letters prefixed to a number.

- Elimination of lower case alphabetic character processing. At MAC messages were from AUTODIN, which is upper case only.

- Elimination of two-dimensional FORTRAN arrays in the METER "microconcordance" procedure for counting stems in a message. This apparently minor revision resulted in an overall speedup of about 10 percent in the message exception.

- Reduction by a half of console output during update processing. Communication with a slow 300 band terminal at MAC turned out to be a significant portion of update time.

- Implementation of functions for file directory operations. These allowed METER programs to take care of many things that an operator would otherwise be responsible for.

- More efficient code for message header source and date/time group processing. A change in the METER restriction form to allow users to prevent retrieval of messages from a given source.

- Precomputation of certain query expansions during the update process to save time in the retrieval process. This involved the addition of two new program modules in the full update sequence.

- A special I/O procedure to save time in reopening the evaluation data output file.

- Optimization of retrieval I/O initialization by eliminating the opening of unnecessary files.

- Adding tags to identify messages in METER displays and allowing users to specify a range of messages in the retrieval form for display.

These changes along with general tightening of code throughout the METER system resulted in a speed up of about fifty percent between the original METER "Advanced Development Model" and the "Advanced Engineering Model" at MAC HQ; additional capabilities were made available to users as well.

6.2 YET TO BE IMPLEMENTED CHANGES

Despite the considerable enhancements already made, current METER processing times are not entirely satisfactory and in fact significant improvements should be possible in the innermost loop of text processing, in producing inverted lists for stems, and in message indexing. These operations currently account for the largest individual contributions to the overall update time for METER.

- The METER QSTM program for text processing is implemented in FORTRAN and currently calls a special assembly language subroutine to read characters one at a time from an input stream of message text. This maximizes the amount of code in FORTRAN to increase portability, but it has the disadvantage of requiring many subroutine calls to process individual characters. The overhead for such calls becomes quite significant over a typical message 2500 characters in length. We have already experimented with an approach to reduce this overhead by having subroutines that return many characters at a time to QSTM. Although the new subroutines would require many times more assembly language code than before, they should reduce overhead by a factor of three or more in the innermost loop of QSTM and therefore allow text processing to be much faster than the current rate of about 3 seconds per message on the PDP-11/45 at MAC HQ. The implementation of these new subroutines would be straightforward, involving mostly code for buffering characters. They should also help to speed up processing in retrieval programs as well, though less dramatically than in the case of QSIM.

- We have implemented an improved version of the METER external sort utility that should speed up the sorting of large files by a factor of two. This will show up mainly in the stem list inversion carried out by the QNVRT program in the METER preprocessor. The improved sort utility was completed too late for installation at MAC HQ, but has been incorporated now into the METER core system for subsequent distribution.

The utility also has the advantage of being able to handle twice as many messages in any single input batch.

- After the preprocessor has run, message indexing currently accounts for over half of METER processing time in full updates at MAC. This can further be reduced in two ways:

    - An index stem lookup procedure taking better advantage of the fact that the stems for a message will come in alphabetic order. This stem lookup makes up the innermost loop of message indexing. An improved procedure has already been put in the METER core system for subsequent distribution, but this was not completed in time for use at MAC HQ.

    - Reorganization of innermost loop processing to consolidate operations for more efficiency. This has been done somewhat already in the METER Core system, but has not gone to MAC. Some additional enhancement remains to be carried out, and some assembly language code might be helpful here.

The speed up from these changes will amount only to about ten percent of message indexing time, but this would still be a significant savings in an overall full update.

On the retrieval end of METER, there are many possible enhancements. During the evaluation period at MAC, METER uses often offered reasonable suggestions on how to improve on the system from their standpoint, and more such suggestions will probably come with further use of METER. Enhancements already implemented at MAC were described in the preceding section; further user-requested changes include the following:

- A METER prompt for entry of query text was mentioned by several analysts. Apparently the MAXI convention of simply putting up "##" on a screen for text entry is often confusing to users. The addition of prompts was easy to make in the distribution system code at PAR, but it involved multiple modules and was not fully implemented at MAC.

- Most users wanted to be able to specify the number of messages to retrieve. To allow for this, we added a field to the restriction form and revised METER retrieval modules to expect a retrieval size specifier in query files. These changes were implemented and tested under simulation at PAR but was not installed at MAC for lack of time and because it would have invalidated METER user's manuals at MAC.

- Widespread user misinterpretation of the histogram in a retrieval report suggests that it should be eliminated at least at MAC. A possibility here would be to display two extra matching stems for a retrieval instead; currently only four top matching stems are given. This change will be easy to make.

Useful changes not specifically asked for by analysts would include the following:

- Merging of FMSIX batch index files during update into a single index file for message display. Users at MAC tend apparently to request many messages at a time; a single index file would eliminate the need here to open and reopen batch index files and therefore save time.

- Different analysts appear to vary greatly in their degree of interest in older messages. METER should probably incorporate a retrieval parameter to allow quicker retrievals when only a few of the most recent days are of interest, as is often the case for analysts on the watch.

- The writing of data to displays should be done in individual retrieval program modules rather than by a single DISPLAY program as in the case now. This will improve response time by eliminating the need to write out and then read back files of characters for display.

Although other enhancements of retrieval modules are possible, the ones described here constitute a reasonable set for the next version of METER. These would not involve a great deal of reprogramming but should bring about quite salient improvements in performance and capabilities as compared to the system that was running at MAC HQ.

APPENDIX A

Meter User's Guide

METER USER'S GUIDE

15 May 1982

DEVELOPED FOR MILITARY

AIRLIFT COMMAND

Under Contract

F30602-80-C-0232

Sponsored By

ROME AIR DEVELOPMENT CENTER

TABLE OF CONTENTS

1. PREFACE

1.1 To The Reader

This User's Guide is written for an intelligence analyst or other MAC HQ interactive user of the METER system who needs to recover information from accumulated message traffic arriving at MAC. The METER system is designed to support analysts in carrying out background research for briefings and reports, in making connections between messages that arrive at widely separated times, in developing scenarios for training, and in keeping track of what is generally going on in message traffic.

You need no prior experience in information retrieval techniques to use METER; METER is simple enough that, after reading this document, attending a training briefing, and acquiring some "hands on" experience, you should have no difficulty in fully exploiting its capabilities. Because METER at MAC is implemented as an application subsystem under MAXI, however, you must be able to use MAXI before you can use METER.

1.2 Related Documents

You may wish to consult the following documents and materials to find out more about METER at MAC:

- The METER System Manager's Guide describes the role played by the System Manager in setting policy in the use of METER at an installation.

- The METER Operator's Procedures Manual describes the role played by the METER Operator in executing various system functions in support of the MAC user group.

- METER Training Materials which further describe basic functions of the METER system.

For details related to terminal operation and the various MAXI subsystems, you should refer to the pertinent documentation for MAXI. Required documentation should be available through the system manager.

1.3 How the User's Guide Is Organized

The METER User's Guide contains these six sections...

1. Preface: an overview of the METER User's Guide.

2. Introduction: an overview of METER and its link to MAXI.

3. METER Functional Overview: an overview of how to use METER.

4. METER Menus and Functions: a more detailed description of METER functions.

5. Data Base Summarization: offline analysis aids available to the user.

6. METER Help Facilities and User's Error Messages: METER provisions for online help and definitions of user error messages.

2. INTRODUCTION

2.1 Why METER?

METER is an information retrieval system designed to simplify and enhance manual and burdensome automatic approaches to storing, retrieving and analyzing message traffic. This is primarily accomplished through automatic indexing of messages for storage, automatic updates to the data base, and automatic user query support. It was funded by RADC because it seemed a better way to manage some of the basic problems of information overflow.

2.2 METER and Its Link to MAXI

METER was developed for MAC because it promised to be an easy, natural, and actually useful message analysis and retrieval system. For example, if you want information on a subject, you can just ask for it in English. Or, if you want to find messages related to one you have on your terminal, just push "RETRIEVE" and METER will search its data base for such related messages. And if ever METER does not seem to be returning the kinds of messages you want, or intermixes messages you like with those you don't, you just have to tell it so, and the next messages it brings you will be more like those you want.

METER is integrated with MAXI to provide you with general facilities for managing message traffic. The MAXI user with METER access can simply press the lighted METER function key (FK) to gain access to METER capabilities.

METER and MAXI have access to the same messages. Messages received at MAC are routed to MAXI where they are directed to analyst profiles and subsequently sent to analyst queues for review and further action. In addition, a copy of each incoming message is sent to the MAXI/METER "gateway" and when scheduled, these messages are sent to METER for inclusion in the METER data base.

METER can be used "independently" of MAXI to provide a message analysis and retrieval capability and it can also be used in conjunction with several MAXI functions. For example, you are reviewing a MAXI message queue and find a message of particular interest. Suppose you are interested in determining if there are any preceding messages in the data base that are related to it. If the message is on your left screen, simply depress the METER FK to get into METER. The message text is preserved on your left screen. Now, depress the RETRIEVE FK and METER will automatically locate messages that are highly related to your message of interest.

In addition, METER utilizes several of the MAXI FK's, such as FORM COMPLETED, PRINT, PRIORITY PRINT, NEXT PAGE and PREVIOUS PAGE. This sharing of some of the basic Function Keys simplifies the MAXI/METER user's interface.

2.3 How METER Works

METER is a statistical textual analysis and retrieval system, designed to enable non-expert computer users to retrieve messages in which they are interested, and to organize large collections of text into information usable by analysts and decision makers. It differs from most currently available message retrieval systems in that, wherever possible, knowledge is built into the system or derived automatically from a message collection rather than demanded from a user.

METER does this by reading the messages with which it is continually supplied, and finding associations between the words (actually word roots or stems) in those messages. That is, if the words "president" and "reagan" are both found in the same messages quite frequently (co-occur), then they shall be said to have a high degree of association. METER statistically determines which stems to compute stem frequency and co-occurrence statistics on. These "major" stems are used to index or tag the messages in the data base for subsequent analysis and retrieval. Then, when a user asks to find messages about a topic like "president," METER knows to look also for messages with words highly associated to those of his query, like "reagan."

By this same principle of finding words which relate to one another, METER can find documents which relate to one another. So, if an analyst has one message, and he wishes to get another one like it, he can submit it to METER (just as he submitted the query "president"), and METER will find those documents in the data collection statistically most similar to his.

Likewise, METER can characterize the whole data base of messages it has, by printing out clusters (groups) of messages which are similar to each other, or by printing out words which associate highly.

The various METER statistical processes occur automatically and the statistical characterization of the data base varies dynamically with changes in the contents of the data base. Conceptually, the METER data base can be considered to store two versions of each message: the full text and message header as received from MAXI and the statistical abstraction of the original message.

Search requests (queries) are entered in natural language: sentences, phrases, or lists of words in any form. Variants of words, such as "decided" and "decision", are automatically recognized and need not concern the user. There is no retrieval vocabulary or formal query syntax to learn.

Retrieved messages are returned in order of estimated relevance to a query, and the retrieval listing represents each message in terms of the top four words responsible for its selection. In this way, METER helps a user to decide which messages are most worth looking at.

2.4 NEW USERS

This section describes how you can start using METER and provides an orientation to the OJ-389 workstation as it pertains to METER.

2.4.1 Obtaining Access to METER

First, You must have a valid MAXI user identification number and access to an OJ-389 workstation. You may then ask the METER system manager for a METER account; usage of METER will be somewhat controlled during its evaluation at MAC. If you invoke METER from MAXI without a METER account, METER will return you to MAXI and notify you of this on the standard error line.

2.4.2 METER OJ-389 Terminal Orientation

To use METER under MAXI, you must understand the following about the OJ-389 analyst workstation.

---- Basic editing and control of screen cursor motion.

---- Selection of MAXI menu options by FK, command line, or lightpen.

---- Explain (user HELP) capability.

---- Screen areas and their uses.

---- Paging of displays on a screen.

---- Standard MAXI facilities such as PRINT.

For information about these topics, see the pertinent MAXI documentation.

2.4.3 Logging On to METER

If you are a MAXI user with METER access you can easily logon to METER. First logon to MAXI. On the right pad of function keys you will find the METER function key lit. Depress the METER FK to gain access to METER.

3. METER Capabilities

In this section, general METER concepts, functions and ways of using the system are overviewed. The way in which you actually use the system will depend on your familiarity with the functions, your knowledge of the data base and your information needs. It is suggested that you read this section before reading the more detailed METER descriptions that follow in Section 4.

3.1 The METER Data Base

The METER data base is intended to support the anticipated volume of message traffic at MAC. Depending on operational requirements and data processing constraints this may mean about 30-45 days of messages. Therefore, for a given time interval, METER will have a copy of every message routed to MAXI from the CSP. When you perform retrievals you can search the entire data base or some predefined subset. The data base is updated automatically by METER at scheduled times or whenever required. The full data base update will generally be scheduled by your System Manager. The fast update procedure may be more useful in time-critical situations since it can input messages to METER with very little delay.

Each message in the METER data base is permanently given a number depending on when it was received. This number is preceded by an 'm' so that message number 400 would be referenced as 'm400'. Messages returned as part of a retrieval also are given a rank which can be a number from 1 to 14 or whatever the upper limit on retrieval size is. You should only be concerned

about the ranks when looking at the REPORT form.

Messages are indexed according to the root words that are most useful in "tagging" the message. METER automatically makes the determination of what root words to use to index messages most effectively. The root words are called stems in METER. Each stem is derived after a word has had its suffix removed and a regular ending added to it. This process is controlled by the METER stemmer which incorporates some 1000 English language suffix rules. For example, the words comparison, comparisons and compare would all be stemmed to the root word compare. You will notice that stems are also used to characterize the contents of messages in the REPORT form as well as to describe data base topics in the clusters produced offline.

3.2 Surveying the Data Base

You may wish to use the METER data base summaries to find out what general topics are covered in the message data base. You can obtain daily or updated data base summaries from the system manager. For example, the stem clusters provide you with a list of natural groupings of key words in the data base. Each grouping corresponds to a topic covered in the collection of messages. Reviewing these clusters or other summary reports may make asking your query a little easier or it may give you insight into what other queries to ask.

3.3 Using the Basic Query Capability

The basic query capability is concise and simple to use. You can simply enter your query on the left screen and depress the RETRIEVE function key. After about 20-30 seconds, the retrieval report will be displayed on the left screen. Queries can consist of sentences, phrases, or lists of words. Capitalization, punctuation, and spacing do not matter.

Another approach is to display a message on your left screen and use this as a query. The message may be displayed using the SHOW option available through the REPORT function or you may display the message text on your left screen while reviewing your message queue under MAXI.

3.4 Restricting Retrievals

You may wish to retrieve from some subset of the entire message collection. The RESTRICT function allows you to subset the data base on message header information (source of the message, the date time group, and the message classification) and/or on the existence of keywords or keyword combinations occurring in the message text. Once the RESTRICT function is invoked, all subsequent retrievals are constrained to those messages in the data base that meet the stated restrictions. The RESTRICT function can be removed by using the ERASE option provided in the RESTRICT form.

3.5 REPORTing the Retrieval and Analyzing the Results

The messages retrieved for your query are displayed on the REPORT form. The REPORT function is automatically invoked whenever the RETRIEVE function is invoked. In addition, you can directly invoke the REPORT function to redisplay the last retrieval. The REPORT form provides you with the retrieval summary as well as several options for further analysis. The SHOW option allows you to display messages at your terminal from the current retrieval. The CULL option allows you to evaluate the retrieval. When using CULL, each retrieved message listed in the retrieval summary can be marked to indicate whether you like the message or not. It is not necessary to evaluate each message, only those you wish to. When your finished with the REPORT form, the NEW QUERY option will give you the opportunity to enter another query, or the SHOW option will display the messages you wished to see, or the CULL option will perform another retrieval based on your evaluation of the current retrieval list.

3.6 COMMENTing on METER

The COMMENT function provides you with an opportunity to comment on any aspect of the METER system. Your comments will be stored in a special evaluation file for later review.

4. METER MENUS AND FUNCTIONS

4.1 METER Menus

A menu is something you can choose functions from. A menu can be visible to you in two ways: through selectively lit function keys and via display on the right screen of your terminal. In certain menus you have certain function keys lit, which means you can perform those functions. In other menus you have other function keys lit and so can perform other functions. Pushing a key invokes that function. It may also change the menu you are in.

Basic METER function keys are provided on the left bank of your terminal. On the right bank, you will find several MAXI function keys lit that are also used by METER. Similarly, METER functions and related MAXI functions are also displayed on the right screen of your terminal so that you may also invoke a METER function using the command line or light pen.

4.2 Functions

To use METER you will have to be aware of two basic menus. The primary METER menu contains the four basic METER functions: RETRIEVE, REPORT, RESTRICT

and COMMENT. The other METER menu contains the command FORM COMPLETED. When you are in either of these menus a function can be invoked by depressing the corresponding function key when it is lit. To guide you in using METER, function keys are lit only when they actually can be used.

The remainder of this section provides a complete description of METER functions. Note that each description defines the basic function, how to use it in the most direct and simple way, and also some other ways to use them. In addition, some functions put up forms on the left screen and these forms are also described.

4.2.1 RETRIEVE

The RETRIEVE function performs a retrieval using the query you have submitted. You can submit queries in several ways; it is important to note that a query can be sentences, phrases or lists of words that you enter from the keyboard or enter on your left screen by displaying the text of some message.

4.2.1.1 The Basic Retrieval Function

To perform the basic retrieval function, enter the text of your query on the left screen. This can be done by simply typing the text onto the screen using the keyboard. (Note that you may have to clear the left screen first.) Next, depress the RETRIEVE function key. After some 30 seconds, the retrieval will be displayed on the left screen. (See description of the REPORT function for a description of the REPORT display.)

4.2.1.2 MAXI Message Review and METER Retrievals

Messages displayed on the left screen can also be submitted as a query. There are several ways to do this. If you are in MAXI, reviewing messages in your MAXI queue, you may display the text of one on the left screen. To submit this message as a query, depress the METER function key on the right

pad of function keys. The message from your MAXI queue will still be on the left screen. The displayed message may have the MAXI header, source header, trailer, and even parts of the text that are not relevant to your query. To prune the message, before submitting it as a METER query, simply use your terminal editor to modify the message. This might mean deleting the header, some text and/or adding text to the original message text to make the appropriate query. When you have finished editing, depress the RETRIEVE function key. A retrieval will be performed and displayed on the REPORT form.

4.2.1.3 Using the REPORT Function to Specify a Query

Another approach to submitting messages as queries is to display a message on your left screen using the REPORT form. The REPORT form provides you with a SHOW option. Use this option to select a message to displayed. (This process is described in more detail in the description of REPORT, below.) Then depress the RETRIEVE FK to perform a retrieval.

4.2.2 REPORT

The REPORT function displays the results of the current retrieval in summary form. Each retrieved message is characterized by the four stems that best describe the similarity between the message and your query. Looking at the stems can often help you to decide if you should read the full text of the message. Another indicator of the relationship of the message to your query is the relevance histogram. This histogram indicates which messages you might want to look at first since it is the basis for rank ordering the messages identified on your REPORT form.

4.2.2.1 Automatic Invocation of the REPORT Function

Whenever you depress the RETRIEVE function key a retrieval is performed. This results in the REPORT function being automatically invoked by METER. You will not have to depress the REPORT function key to get the retrieval results.

4.2.2.2 Re-Displaying the REPORT Form

If you wish to review the current retrieval report simply depress the REPORT function key. The last retrieval you performed will be redisplayed on the left screen.

4.2.2.3 Using the REPORT Form

The REPORT form is used to display a retrieval and as a vehicle for selecting several related options. The form will be described below.

Line 1: The name of the form.

Line 2: The second line of the form, as shown in Figure 4-1, provides you with three choices. First, there is an option to show (SHOW) or display the text of a message in the retrieval list, below. Next, an option for leaving the REPORT form and entering a new query (NEW QUERY) is given. The final option allows you to evaluate the retrieval display shown below as part of the CULL option. Place an 'X' to indicate your choice. If no choice is made METER will assume you wish to enter a new query.

Line 3: The third line on the form allows you to select a message for display by simply entering its message number. Messages have a permanent METER data base number, such as m124. All METER message numbers begin with 'm'. So to select some message of interest directly from the data base, you must know its METER message number. You can also select a message from the retrieval list for display by specifying its relative retrieval number. This will typically be a number from 1 to 14.

RETRIEVAL REPORT FORM

MARK ONE OPTION WITH 'X': SHOW (.) NEW QUERY (.) CULL (.)

TO SHOW A MESSAGE, ENTER ITS METER MESSAGE NUMBER: (.....)

Msg	Y/N	Top Matching Stems				Estimated Relevance
1 -	(.)	SATELL	NUCLEAR	SOVIET	SPACE	IXXXXXXXXXX
2 -	(.)	SATELL	SOVIET	NUCLEAR	RADE	IXXXXXXXXX
3 -	(.)	SATELL	SOVIET	NUCLEAR	RADE	IXXXXXXXXX
4 -	(.)	SATELL	ORBIT	LAUNCH	EARTH	IXXXXXXXX
5 -	(.)	SATELL	NUCLEAR	SOVIET	ORBIT	IXXXXXXXX
6 -	(.)	SATELL	NUCLEAR	SOVIET	FAILSAFE	IXXXXXXXX
7 -	(.)	SATELL	NUCLEAR	SOVIET	FAILSAFE	IXXXXXXXX
8 -	(.)	SATELL	NUCLEAR	SOVIET	FAILSAFE	IXXXXXXXX
9 -	(.)	SATELL	SOVIET	RADE	MILE	IXXXXXXXX
10 -	(.)	SOVIET	SATELL	ATOM	SPACE	IXXXXXXXX
11 -	(.)	SATELL	SOVIET	RADE	MILE	IXXXXXXXX
12 -	(.)	SATELL	NUCLEAR	SOVIET	MILE	IXXXXXXX

Figure 4-1: The REPORT Form

Remaining Lines: The remainder of this form provides you with the retrieval summary. Each line represents a retrieved message. The line is divided into four parts:

1. "Msg" The message number corresponds to the ranking of each message in the retrieval. Message 1, has been determined to be the most relevant to your query.

2. "Y/N" This column is used in conjunction with the SHOW and CULL options described above. If you checked SHOW, above, and enter 'Y' next to some message number, that message will be displayed. (You do not have to mark 'N' to not see a message.) If you marked the CULL option, and enter 'Y' next to some message, that message will be considered by METER as the kind of message that is of general interest to you. If you mark 'N' while using CULL, then those messages will be considered irrelevant to your current interests.

3. "Top Matching Stems" The top four stems occurring in each message that are most relevant to your query are displayed. These stems provide you with information to determine which messages are most relevant to your query and worth displaying on your terminal.

4. "Estimated Relevance" The histogram provides a relative estimate of the association between your query and each message in the retrieval list.

4.2.2.4 When You Are Finished With the REPORT Form

After you have reviewed the REPORT form and selected required options, depress the FORM COMPLETED function key on the right function key pad. This

will submit the REPORT form for processing.

4.2.3 RESTRICT

The RESTRICT form can be used to subset the message data base. The RESTRICT form can also be viewed as means to set-up a message filter for limiting retrievals. You will be able to filter retrieved messages or similarly subset the message data base based on two types of message qualifiers: message header and keyword.

4.2.3.1 Restricting the Data Base on Message Headers

If you wish to limit retrievals based on the message header depress the RESTRICT FK. In the "Message Header Restriction" part of the form enter the message source, classification, and date time group that acceptable messages must satisfy.

4.2.3.2 Restricting the Data Base on Keywords in the Text

If you wish to limit retrievals based on keywords, enter the keywords that are required to be in the acceptable message under the form area called "Messages That Include The Keywords". Enter the keywords that should not be in acceptable messages under the form area "But Exclude The Keywords". It is possible to restrict messages based on the header and/or keyword restrictions. Note that whenever you use an "exclude" keyword you must use at least one

"include" keyword.

4.2.3.3 Removing Restrictions

To remove some or all of the restrictions, automatically, simply employ the ERASE option located at the top of the form. You may erase the header restrictions, keyword restrictions or both using this option.

NOTE THAT WHILE RESTRICTIONS ARE INVOKED ALL RETRIEVALS ARE LIMITED BY THE RESTRICTION PARAMETERS SET-UP ON THE RESTRICT FORM.

4.2.3.4 Description of the RESTRICT Form

The RESTRICT form shown in Figure 4-2, has three main parts: the Erase options, the Message Header Restrictions, and the Keyword Restrictions. The form will be described below.

Line 1: The name of the form.

Line 2: At this position you may elect to erase all restrictions (enter a

```
RETRIEVAL RESTRICTION FORM

ERASE: (.)   1= ALL RESTRICTIONS   2= ONLY MESSAGE HEADER   3= ONLY KEYWORD

-------------------------------------------------------------------------------
MESSAGE HEADER RESTRICTIONS:

SOURCE:              (........)

CLASSIFICATION :  (3)   1= UNCLAS   2= COLLAT   3= COMPART
DATE TIME GROUP:   FROM(010000Z JAN 81)       TO(312254Z DEC 88)
-------------------------------------------------------------------------------
MESSAGE KEYWORD RESTRICTIONS:

MESSAGES THAT INCLUDE THE KEYWORDS

   (...........)   (...........)   (...........)   (...........)

BUT EXCLUDE THE KEYWORDS

   (...........)   (...........)   (...........)   (...........)
```

Figure 4-2: The RESTRICT Form

1), only message header restrictions (enter a 2) or only keyword restrictions (enter a 3.)

Line 3: The name of the form subarea: Message Header Restrictions.

Line 4: Enter the name of the agency or unit which sent the message.

Line 5: Select the highest level of message classification you wish to see.

Line 6: Specify a time interval in which the message was sent. Use the space after "From" to enter the start of the time interval in Zulu time, and use the space after "To" to enter the end time for the time interval of interest.

Line 7: The name of the form subarea: Message Keyword Restrictions.

Line 8: The subarea section named: Messages that include the keywords.

Line 8: Enter up to four keywords that should be included in the desired messages.

Line 9: The subarea section named: But exclude the keywords.

Line 10: Enter up to four keywords that should not be included in desired messages.

4.2.3.5 When You Are Finished With the RESTRICT Form

After you have completed the RESTRICT form, depress FORM COMPLETED to submit the form for processing.

4.2.4 COMMENT

The COMMENT function can be used to write any comments you might have to the METER evaluation file. This file is used by METER evaluators to determine user assessments of METER functions. The COMMENT form is shown in Figure 4-3. After depressing the COMMENT FK, enter your comments and conclude by depressing the FORM COMPLETED FK.

5. DATA BASE SUMMARIES

METER can produce a variety of reports summarizing the content of a data base. These reports are based on the statistical content analysis performed by METER as an integral part of its data base update process. They will be generated and printed by the METER system manager for offline use. You should read them to become familiar with the kinds of messages currently available for retrieval.

The following METER summary reports are now available:

- INDEX STEMS

  This is an alphabetic listing of all stems chosen to index messages, plus a listing of the top 100 index stems in order of their statistical content measures. You can see how often each stem occurs in the current data base and get an idea of the relative importance of the stem. Examples of this report are given in Figure 5-1.

- ASSOCIATIONS

  This is an alphabetic listing of index stems and their associated stems. Highly associated stems tend to indicate major themes within a data base. An example of the associations report is given in Figure 5-2.

- CLUSTERS

  This is a reorganized presentation of stem associations, showing the stems that tend to occur together. Each cluster can be thought of as a major theme within a data base. Sample cluster output from an Associated Press message data base is given in Figure 5-3.

- CHANGES

  This compares the latest set of index stems with the preceding set and notes changes. Such changes in message traffic may indicate events of

Top 100 Stems for 4 measures

Var		nt		Ft		FT/nt	
Stem	Weight	Stem	Weight	Stem	Weight	Stem	Weight
YES	205	NEW	953	STATE	1836	CDY	62
ETHIOPIA	152	WASHINGT	890	NEW	1849	CLR	52
CUBA	143	STATE	887	WILL	1642	BOYLE	48
BUTT	141	YEAR	827	YEAR	1575	CHOP	41
FLU	135	WILL	813	OFFICE	1386	YABLONSK	41
PEND	131	PRESS	789	PRESIDE	1345	POLANSK	39
COSM	128	ASSOCE	746	WASHINGT	1339	CLOUD	38
GOFORTH	127	OFFICE	721	REPORT	1120	LOWELL	36
TOMAT	124	WRITE	701	NATE	1107	SLUG	36
TRUONG	124	PRESIDE	673	GOVERN	1084	BENZ	35
SOLDIER	120	REPORT	645	CART	1021	SOMALIA	35
SUGE	120	NATE	637	PRESS	970	HUMPHREY	34
ITEM	119	GOVERN	560	COMMIT	917	VIET	33
HUGH	118	WORK	528	ASSOCE	868	FRUIT	32
RETIRE	117	PLAN	491	WORK	835	ISRAEL	32
ANGOLE	113	FIRST	473	FEDER	819	TRUONG	32
CHUCK	112	ACT	472	AMER	800	CAMBODIA	31
LEND	112	ADD	472	WRITE	791	FRANC	31
TREAT	112	TUESDAY	471	PLAN	776	INM	31
UNCHANGE	112	WEEK	471	HOUSE	774	MUGABE	31
SUN	111	AMER	460	ACT	755	OFFSHORE	31
CAMBODIA	109	COMMIT	457	UNITE	733	ORANGE	31
KOREA	109	CALL	451	MILLION	718	VACCINE	31
SCOTT	109	CART	449	SEN	705	FEE	30
TAX	108	FEDER	444	DEPART	689	PORK	30
TOBACC	107	HOUSE	433	TUESDAY	684	SOMAL	30
GUN	106	UNITE	423	WEEK	680	ANGOLE	29
HUNG	106	TOLD	420	LEAD	665	EGG	29
ORANGE	106	LEAD	412	PERCENT	665	FOX	29
ABC	105	WEDNESDA	408	FIRST	637	SATELL	29
BANKRUPT	105	SERVE	407	ADMINIST	620	WEB	29
PARK	105	MILLION	397	WEDNESDA	615	AMIN	28
ERODE	104	DEPART	387	ADD	609	PANAME	28
JANUE	104	MADE	379	SERVE	608	PICKET	28
AFR	103	MAY	379	CALL	602	VANZETT	28
TRU	103	PART	378	LAW	599	FITZSIMM	27
FOX	102	ADMINIST	376	TOLD	587	NKOMO	27
LOB	101	PUBLE	369	CITE	569	PALESTIN	27
RHODESIA	101	MEMB	359	MEMB	563	PATRIOT	27
STANDARD	101	MONDAY	354	PUBLE	560	PEKING	27
MCDONALD	100	ASK	353	MAY	525	SNOW	27
PHIL	99	CITE	352	RIGHT	521	CANE	26
STRANGLE	99	GET	349	PART	519	DEFICIT	26
TON	99	PEOPLE	345	WEST	518	EGYPT	26
GRAHAM	98	FOUND	344	MONDAY	513	FARM	26

Figure 5-1: METER Index Stems

associations (scale factor: 10)

lower threshold = 0.80

stem	stem	measure	stem	measure	stem	measure	stem	measure	stem	measure	stem	measure
ALFR :	ATHERTON	0.807,										
ANDERSON:	GARE	0.826,										
ARM :	EGYPT	0.817,	SADAT	0.838,								
ASSET :	BARON	0.850,	CASIN	0.910,	DILIGE	0.814,	DUFFEY	0.814,	GOLDFARB	0.814,	HOLME	0.814,
	KICKBACK	0.810,	LUEK	0.814,	MATHESON	0.814,	RAHNEY	0.814,	SHANNON	0.850,	SHEETZ	0.814,
	SPICK	0.814,	TURNIP	0.814,								
ATHERTON:	ALFR	0.807,										
AVIV :	TEL	0.804,										
BARON :	ASSET	0.850,	CASIN	0.956,	DOSKIN	0.815,	FIDUCIAR	0.813,	FITZSIMM	0.819,	JACKIE	0.873,
	KICKBACK	0.956,	SHANNON	1.002,	TEAGUE	0.847,						
BENZ :	EXPOSE	0.810,	LEUKEMIA	0.855,	SCHMID	0.815,	STANDARD	0.801,				
BUTT :	FRANKFUR	0.819,	ITEM	0.827,	TOMAT	0.826,	UNCHANGE	0.872,				
CANE :	PANAME	0.729,	TREAT	0.832,								
CASIN :	ASSET	0.810,	BARON	0.956,	DOBKIN	0.872,	FIDUCIAR	0.870,	JACKIE	0.838,	KICKBACK	0.911,
	SHANNON	0.956,	TEAGUE	0.807,								
CHAT :	FIRESIDE	0.933,										
CHINATOW:	DALTON	0.807,	POLANSK	0.905,	RITTENBA	0.868,						
CLOUD :	RAIN	0.824,										
CONTAMIN:	DUTCH	0.800,										
COSM :	HAIR	0.800,										
CRATE :	DUBAWNT	0.859,	WILDLIVE	0.816,								
DALTON :	CHINATOW	0.807,	DIAGNOST	0.887,	POLANSK	0.818,	RITTENBA	0.833,	ROSEME	0.807,		
DEBR :	GROVE	0.810,	RADE	0.823,								
DIAGNOST:	DALTON	0.887,	POLANSK	0.879,	RITTENBA	0.958,						
DILIGE :	ASSET	0.814,	DUFFEY	1.002,	FIDUCIAR	0.862,	FITZSIMM	0.842,	GOLDFARB	1.002,	HOLME	1.002,
	JACKIE	0.977,	KICKBACK	0.956,	LUEK	1.002,	MATHESON	1.002,	RAHNEY	1.002,	SHEETZ	1.002,
	SPICK	1.002,	TEAGUE	0.932,	TURNIP	1.002,						

Figure 5-2: Stem Associations

[illegible]

[illegible]

[illegible] more than 1000 associations

19 clusters formed.

[illegible]

CLUSTER # 1
STEMS:

ASSET	BARON	CASIN	DILIGE	DOBKIN	DUFFEY
FIDUCIAR	FITZSIMM	GOLDFARB	HOLME	JACKIE	KICKBACK
LUEK	MATHESON	RAMNEY	SHANNON	SHEETZ	SPICK
TEAGUE	TURNIP				

CLUSTER # 2
STEMS:

CHINATOW	DALTON	DIAGNOST	NICHOLSO	POLANSK	RITTENBA
ROSEME	SANT				

CLUSTER # 3
STEMS:

BENZ	EXPOSE	LEUKEMIA	SCHMID	STANDARD

CLUSTER # 4
STEMS:

BUTT	FRANKFUR	ITEM	TOMAT	UNCHANGE

CLUSTER # 5
STEMS:

CONTAMIN	DUTCH	FRUIT	ORANGE

CLUSTER # 6
STEMS:

DEBR	GROVE	RADE	WARD

CLUSTER # 7
STEMS:

CANE	PANAME	TREAT

Figure 5-3: Stem Clusters

emerging importance. Sample output is given in Figure 5-4.

- TRENDS

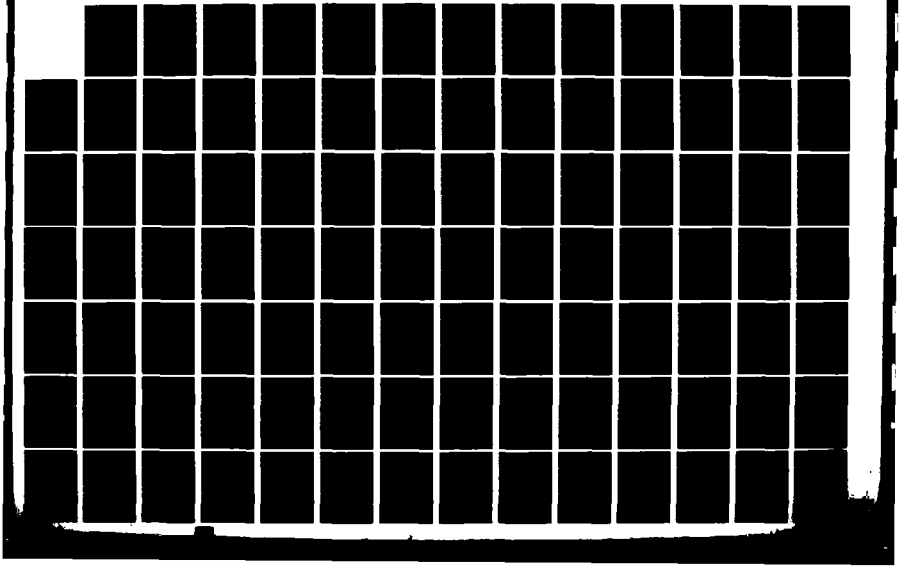
  This shows index stems with a significant increasing or decreasing rate of daily occurrence. Currently, it measures the sum of the differences between the number of occurrences for stems over 7 consecutive updates. Sample output is given in Figure 5-5.

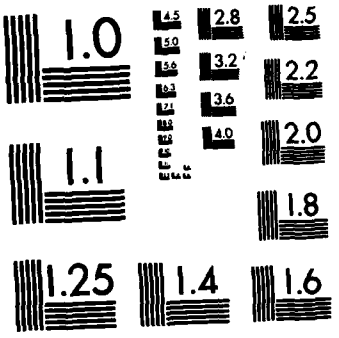
- ANOMALIES

  This shows index stems with an unexpected rate of occurrence in a data base with respect to recent history of occurrence. This may indicate unusual activity being reported in message traffic. Sample output is provided in Figure 5-6.

AD-A125 166 OPERATIONAL TEST AND EVALUATION OF THE METER ENGINEERING DEVELOPMENT MODEL(U) PAR TECHNOLOGY CORP NEW HARTFORD NY R J D'AMORE ET AL. NOV 82 2/4

UNCLASSIFIED RADC-TR-82-304 F30602-80-C-0232 F/G 14/2 NL





emerging importance. Sample output is given in Figure 5-4.

- TRENDS

  This shows index stems with a significant increasing or decreasing rate of daily occurrence. Currently, it measures the sum of the differences between the number of occurrences for stems over 7 consecutive updates. Sample output is given in Figure 5-5.

- ANOMALIES

  This shows index stems with an unexpected rate of occurrence in a data base with respect to recent history of occurrence. This may indicate unusual activity being reported in message traffic. Sample output is provided in Figure 5-6.

6. METER Help Facilities and User Error Messages

6.1 Help Facilities

METER provides you with limited online help through the MAXI EXPLAIN function key. Whenever you have the main METER menu in force, you can request short descriptions of METER functions. For example, to get a description of the RETRIEVE function, simply depress the EXPLAIN FK followed by the RETRIEVE FK. The EXPLAIN FK will be now be backlit and a description of the RETRIEVE function will appear on your right screen. As long as the EXPLAIN FK is backlit, you are operating METER in a "help" mode. To get back to normal METER operation, depress the EXPLAIN FK; it will no longer be lit. Then continue with your current task.

The following help text is provided in METER:

- METER. This brings up the METER system, allowing an analyst to retrieve messages with natural language queries taken from the left screen of a workstation. A query may be the text of a displayed message or text directly entered from the keyboard. Retrieved messages will be ranked by estimated relevance to a query. Four VFK's in the left bank will be defined for METER operations: RETRIEVE, REPORT, RESTRICT, COMMENT.

- RETRIEVE. This initiates a retrieval with query text taken from the left screen of a workstation. If the left screen displays a message text,

then METER will retrieve those messages most similar to it. An analyst may freely edit the left screen or enter text from the keyboard before invoking RETRIEVE. The REPORT function is automatically invoked to show results of a retrieval.

- REPORT. This puts up a METER retrieval report form on the left screen of a workstation. The form shows the ordering of retrieved messages for a query plotting their estimated relevance as a bar graph. It also gives the top four index word stems contributing to the estimated relevance of each message. The SHOW option on the report form allows messages to be displayed; the CULL option allows an analyst to mark retrieved messages as relevant or irrelevant in order to retry a retrieval with more information.

- RESTRICT. This puts up the METER restriction form allowing an analyst to restrict METER retrievals to those messages falling within a specified range of dates and times, coming from a specified source, having a specified maximum security classification level, or containing a specified combination of keywords. The form will show current restrictions in force.

- COMMENT. This puts up a METER comment form and allows an analyst to enter remarks about METER and its performance. These will be collected as part of the operational test and evaluation of METER at MAC.

6.2 Error Messages

METER uses a set of error messages for display at your terminal. These messages vary in their utility to you since in some cases they inform you of

some system state that you can change directly but in other cases inform you that there is a more serious problem that your system manager must address. The METER user's errors, enclosed in quotes, are as follows:

"QUERY MUST BE ENTERED BEFORE RETRIEVAL"

METER requires that you have submitted a query before attempting a retrieval. Check to see that you have either typed a query or displayed message text to the left screen before depressing the RETRIEVE FK.

"NO MAJOR STEMS FOUND IN QUERY"

METER messages are indexed on stems called "major" stems; these are the stems that are most useful in indexing messages. If your query does not contain a major stem a standard RETRIEVE-type retrieval cannot be performed. You may either re-phrase your query such that major stems are included or submit the original query using the RESTRICT form in a data base subsetting operation. Then either show messages directly or submit a query and use RETRIEVE.

"TOO MANY MESSAGES IN RESTRICTED COLLECTION"

When you restrict, i.e., subset the data base, METER keeps a list of messages included in the subset. The maximum size this list can be is determined by the amount of space available on the processor. If your subset is too long, you may have to adjust the keywords used on your RESTRICT form; possibly adding a keyword.

"NO POSITIVE KEYWORDS WERE ENTERED"

When doing a RESTRICT on keywords, METER requires that there be at least one keyword in the form area titled "Messages that include the keywords". The system does not allow for only keywords under the section titled "But exclude the keywords".

"ILLEGAL MESSAGE NUMBER ENTERED"

This error message will occur if you try to display a message that does not exist. This can occur if you specify an absolute message number; that is a message number preceded by an 'm' that does not exist in the data base or it can occur if you specify a relative message number from the REPORT form that is outside the allowable range.

"NO METER ACCOUNT"

To use METER, you must have a METER account. See your System Manager to for access privileges.

"METER ERROR"

The METER system is defined to fail gracefully, should a system error occur, other than those listed above. If this error message occurs you should see your System Manager since the error is probably software or hardware oriented.

APPENDIX B

Meter Operator's Procedure Manual

PAR Technology Corporation
New Hartford, New York 13413

July 31, 1982

METER Operator's Procedures Manual

Prepared under RADC Contract F30602-80-C-0232

ABSTRACT

This manual is written for the METER Operator of the MAC/IN PDP 11 (AN-GYQ 21(V)) computer system. It explains the role of the METER Operator, his responsibilities in keeping METER a usable system, and the tools available for such upkeep. The Operator must have hands on experience running the PDP 11 under RSX-11D/IAS.

This manual is a completely new version of the OPM, and replaces all earlier versions.

TABLE OF CONTENTS

1. THE OPERATION OF METER

1.1 Related Documents

Although the Operator will have the most day to day interaction with METER, detailed knowledge of each of the METER program's internal operations, or of its files is unnecessary. But it is important that he know how to keep METER operational (which is the purpose of this document), how to enact the decisions of the METER System Manager, and how METER appears to the user. For this reason it is highly recommended that the Operator read the METER System Manager's guide and, the METER User's Guide.

You may wish to consult the following documents and materials to find out more about METER at MAC:

- The METER System Manager's Guide describes the role played by the System Manager in setting policy in the use of METER at an installation.

- The METER/MAXI Interface Description
  - describes the relationship between the METER and MAXI systems.

- METER Training Materials which further describe basic functions of the METER system.

For details related to terminal operation and the various MAXI subsystems, you should refer to the pertinent documentation for MAXI. Required documentation should be available through the system manager.

1.2 Assumptions

- The "PDP-11/70" contains the MAXI system.
- The "PDP-11/45" contains the METER system.
- Text appearing between {}'s in examples are comments.
- All examples assume the operator has not logged into his terminal, and will always start with the command: "HELLO".
- Operators have knowledge of the IAS operating system, and at least one text editor. Examples showing the use of a text editor do not give the editor commands.

1.3 An Introduction to METER

METER is a statistical textual analysis and retrieval system which allows non-expert users to retrieve messages with a minimum of effort. To initiate a retrieval, the user simply provides METER with a query in the form of sentences, word lists, or message text. METER will then attempt to retrieve messages which are statistically similar.

METER receives messages from MAXI in batches. These messages are accumulated until there are enough to justify a database update; full updates will usually be once each day, but may be more often when messages are coming in during crisis situations. In a full update, METER analyzes all new messages and relates them to its previous messages. Once an update is complete, users may retrieve the newest messages by entering queries. At any point, METER can print topic summaries of messages in a collection from the last update.

At MAC, METER will run on its own processor, currently a DEC PDP-11/45. Users will talk to METER from Sperry-Univac 1652 (OJ-389) analyst work stations. The METER system manager and the METER system operator will manage METER from programming terminals such as the DEC VT52. There will be a special LA-120 terminal (TT5:) designated as the METER system operator's console terminal.

1.3.1 The Parts of METER

METER at MAC/IN consists of four major parts called the acceptor, the analyzer, the summarizer, and the extractor. These are also referred to as the "Q", "P", "R", and "S" subsystems respectively.

METER programs will be in separate directories for the message acceptor, message analyzer, message extractor, and the message summarizer. Other directories will contain METER databases, support libraries for METER programs, command files for the METER system operator, and various other files. These directories are described more fully in Appendix C.

The operator will normally be concerned with only with the METER [270,1] system operator's directory and the METER [277,*] user directories.

1.3.1.1 The message acceptor

This subsystem is responsible for passing batches of messages across the ICC from MAXI to METER. It also performs initial statistics gathering prior to a full update. In all, four programs make up the message acceptor subsystem.

1.3.1.2 The message analyzer

This subsystem consists of sixteen programs responsible for building the METER database. Statistics about the messages received from the acceptor subsystem are collected and used to build index tables which the retrieval system will use to access the messages.

1.3.1.3 The message extractor

This subsystem is responsible for retrieving messages based on queries given by the analyst. Retrieval programs interact with the user through the analyst work stations.

1.3.1.4 The message summarizer

The summarizer subsystem consists of a series of programs designed to provide users with information about the contents of the current database. These include: major stem clusters, associations, top stems, stem changes and trends occurring in the major stems since the last update.

1.4 METER Program Flow

Figure 1 shows METER system operation, omitting message extraction and assuming full updates only. Details of this operation are contained in the following sections, followed by an example of an actual running of the message acceptor, analyzer, and summarizer programs. Further information can be found in the METER System Manager's Guide and the METER Core System Description.

1.5 The Major Jobs of the METER Operator

The METER operator will be responsible for the following:

- Running database updates
- Printing database summaries
- Maintaining user directories
- Adjusting the database control parameters
- Correcting errors indicated by the METER system
- Installing and removing users
- Starting up METER when necessary

Each of the above will be explained in greater detail in later sections of this document.

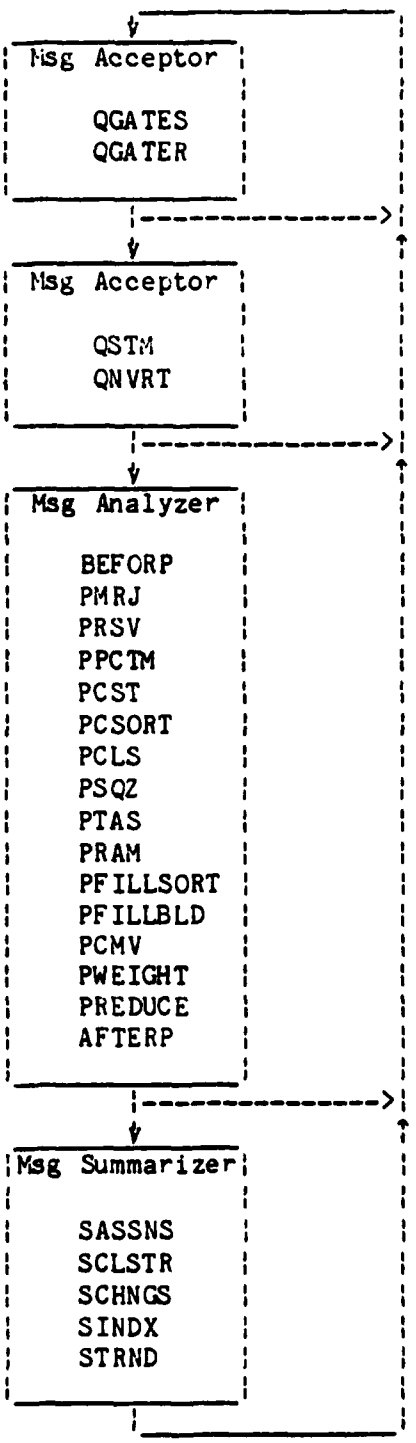


Figure 1. METER operational flowchart

2. INSTALLING NEW USERS

2.1 Overview

The unique ID number assigned to each MAXI user is also used by METER to put the user into his own METER work directory. For METER to do this it must read a table of user numbers in order to map into the correct directory. This table is set up by the METER operator.

2.2 Creating the User Table

2.2.1 Initial Table Creation

To create a table from scratch enter the following on the PDP-11/70 console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,1]
MCR> EDI USERS.TXT      { or EDT }
{ Enter 9 digit user ID numbers in a single column}
MCR> RUN MAKEUSERS<esc>
```

The MAKEUSERS program will map each 'user ID to a directory in group 277'. That is: [277,1] will be the first users directory, [277,2] is the second users directory, and so on up to [277,n] where n is the last user in the user table.

For each user entered into the table on the PDP-11/70, a directory must also be created on the PDP-11/45. To do this, enter the following:

```
MCR> HELLO [1,1]
MCR> INS [11,1]UFD
MCR> UFD SY:[277,1]/PRO=[RWED,RWED,RWED,RWED]  { First User in table }
MCR> UFD SY:[277,2]/PRO=[RWED,RWED,RWED,RWED]  { Second user }
          .
          .
          .
MCR> UFD SY:[277,n]/PRO=[RWED,RWED,RWED,RWED]  { Last user }
```

Please note that the directory numbers are in octal NOT decimal. If the UFD is not installed, and cannot be installed due to insufficient space, you may have to remove some non-essential program before continuing.

2.2.2 Maintaining Existing User Tables

If a user table already exists, and you wish to add or delete users; enter the following from the PDP-11/70:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,1]
MCR> RUN DUMPUSERS<esc>
Dumping METER users table
USER = 111111111              { 1st user's ID number }
USER = 222222222              { 2nd user's ID number }
     .
     .
     .
USER = nnnnnnnn               { Last user' s ID number}
     m                        { m = total users defined }

^C                            { Hit CONTROL C to get MCR }
MCR> EDI USERS.TXT
{ Add or delete users }
MCR> RUN MAKEUSERS<esc>
```

To delete a user, replace his ID number in the list by zero's. Never delete a user ID by using the editor - always replace. If this is not done, every user after the deleted user, will be mapped into the wrong directory. To add a new user, put his ID number at the end of the list, or at the first available slot.

3. MAINTAINING USER DIRECTORIES

3.1 Overview

Each user who has been entered into the METER user table is assigned his own directory (See the "ENTERING NEW USERS" section in this document). This directory holds temporary files generated by METER during retrievals. The METER operator will be responsible for maintaining these directories.

3.2 Initial User Directory Setup

Before the user can do retrievals, his directory must be initially loaded with two special files. To create these files enter the following from the METER console on the PDP-11/45:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[277,277]
MCR> PIP
PIP> [277,xxx]/DF
PIP> DATABASE.UIC=TI:
[270,5]
^Z                    {Enter "Control Z"}
PIP> EVALNS.DAT=TI:
^Z
PIP> ^Z
```

{ NOTE: xxx is the user directory number }

The user directory is now available for the user to do retrievals.

3.3 Purging User Directories

Doing retrievals in METER may generate a great deal of temporary files in each of the users directories. METER retrieval programs are capable of deleting some of these files but this may not be sufficient - particularly if there is a shortage of disk space. If space is needed, here are three ways to get some:

3.3.1 Have the User Purge His Own Directory

This is the easiest method of cleaning out a directory. Have the user do the following:

1. From the 1652 terminal have the user log in (If he has not already done so), and request the "RESTRICT" function.

2. Have him request option "1" for the "ERASE" option.

This option will purge most of the files from the users directory.

3.3.2 Purging the Users Directory for Him

In some cases, the analyst may not be available to purge his directory. It is possible to purge it for him - since he only needs the most recent files. To do this, enter the following from the METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[277,277]
MCR> PIP
PIP> [277,xxx]/DF
PIP> *.*/PU
PIP> ^Z
```

{ NOTE: xxx is the user directory number }

3.3.3 Purging All User Directories at One Time

If you would like to purge all user directories at one time, enter the following on the METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[277,277]
MCR> PIP
PIP> [277,*]*.*/PU   {This will do it!}
PIP> ^Z
```

3.3.4 One Last Method

One last way to clean up a user directory is to delete its contents and re-create its initial files. This step should only be done if the user is not using METER. From the METER console enter:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[277,277]
MCR> QUE [277,xxx]EVALNS.DAT  {Print out this file}
MCR> PIP
PIP> [277,xxx]/DF
PIP> *.*;*/DE          {Delete the directory contents}
PIP> DATABASE.UIC=TI:
[270,5]
^Z
PIP> EVALNS.DAT=TI:
^Z
PIP> ^Z
```

{ NOTE: xxx is the user directory number }

Please note that it is important that the EVALNS.DAT file be printed before doing the deletion.

3.4 Important Things to Remember

- All files in the users directory must be created under the [277,277] UIC or "Protection" violation errors will result. These errors may not necessarily be obvious when they occur.

- Each user directory must contain a DATABASE.UIC and EVALNS.DAT file before the user can use METER to do retrievals.

- Never purge or delete files in a users directory while retrievals are in progress. It may be best to notify users ahead of time.

4. STARTING UP METER

4.1 Overview

Before METER can do any work for MAC users, it must make itself known to MAXI. This is accomplished by running three command files which log METER into MAXI as well as install the necessary METER tasks on both systems.

4.2 Running the Start Up Command Files

METER MUST BE STARTED
BEFORE ANY MESSAGES ARE PROCESSED, OR RETRIEVALS INITIATED

To start up METER you will need access to terminals on both systems. Enter the following:

```
{On the PDP-11/45}

MCR> HELLO [1,1]

{ Ignore the next 4 lines if ICC is already up }

MCR> INS @[72,5]ICCINS
MCR> LOA DA
ICC ... Processor#> 2      {Reply 2 for the processor number}
MCR> NET /P1


MCR> @[270,20]ICCINS
..... command file output appears here .....
MCR> @[270,20]ICCLOG
..... command file output appears here .....


{On the PDP-11/70}

MCR> HELLO [1,1]
MCR> @[270,20]USSINS
..... command file output appears here .....
```

METER should now be fully operational.

5. METER DATABASE UPDATES

This section describes the steps involved in adding new messages to the METER database. Prior to performing any of the tasks in this section make sure that METER is installed and logged into MAXI or loss of messages may result. If you have any doubts, refer to the "STARTING UP METER" section in this document before continuing.

5.1 Getting messages into METER

5.1.1 Overview

MAXI stores message texts intended for METER in individual files on the PDP-11/70. Since the most of METER resides on the PDP-11/45, a special gateway has been developed to accomplish the task of moving the texts over. The gateway consists of two programs which transmit messages from the PDP-11/70 to the PDP-11/45.

The first program (called QGATES) is installed on the PDP-11/70 and is responsible for reading the individual message files stored in directory [270,100] by MAXI, and sending them across the ICC. As each file is transmitted, it is deleted.

The second program (called QGATER) is installed on the PDP-11/45 and is responsible for taking messages from the ICC and placing them in a METER message file called MESSAGES.DAT. This file is located in [270,5] - the METER database directory.

These programs will generally be scheduled to run every two or three hours until enough messages have been collected to perform a database update. When message traffic is heavy, these programs may be run more often. The following shows two examples for moving message texts to METER.

5.1.2 Manual Transmission of Messages to METER

If messages need to be sent to METER immediately, the following should be entered from the PDP-11/70 console:

```
MCR> HELLO [1,1]           { On the PDP-11/70 }
MCR> SET /UIC=[270,100]    { Gets you to the messages }
MCR> PIP /LI               { Displays message files to move }
MCR> RUN QGATES<esc>       { Moves the texts across ICC }
MCR> PIP /LI               { Confirms messages were moved }
```

The second PIP will indicate a successful transmission of texts if you receive a message indicating no files were found in the directory. After you gain more experience with the system, you may omit the PIP's as well as the "SET /UIC".

5.1.3 Automatic Transmission of Messages to METER

Messages may be sent to METER at pre-determined intervals by entering the following from the PDP-11/70 console:

```
MCR> HELLO [1,1]
MCR> RUN QGATES /RSI=2H<esc>    {Runs QGATES every two hours}
```

Once the command is entered QGATES will send messages to MAXI every two hours. The interval may be adjusted as needed. For further information refer to the IAS MCR User's Guide (Page 7-48).

A word of caution: If METER is taken down on the PDP-11/45 - for whatever reason, make sure that QGATES is disabled or MESSAGES MAY BE LOST. To disable QGATES, enter the following on the PDP-11/70 console:

```
MCR> HELLO [1,1]
MCR> REM QGATES
```

To restart METER, follow the instructions in the "STARTING UP METER" section in this document.

5.2 Preprocessing Messages prior to an Update

5.2.1 Overview

Before any update of the database can be performed, the message texts must be preprocessed. Preprocessing involves counting stems occurring in each message as well as recognizing and indexing the message header, body, and trailer. It is during this step that header information about a message's source, classification, and date/time are extracted and saved. In short, anything having to do with the text of a messages is performed in this step.

5.2.2 Preprocessing messages

Preprocessing the messages does not mean you are committed to doing a database update immediately. In fact, you may preprocess several batches of messages if you wish. The only requirement is that MESSAGES.DAT file be deleted, or renamed after preprocessing is completed; otherwise, duplicate messages will be entered into the database.

To preprocess a batch of messages, enter the following from the METER console on the PDP-11/45:

```
MCR> HELLO [1,1]
MCR> SET  /UIC=[270,5]
MCR> @[270,1]PREPROC
..... Command  file output appears here .....

MCR> PIP MESSAGES.DAT;/DE     {Do this only if instructed}
     -- or --
MCR> PIP MESSAGES.OLD=MESSAGES.DAT/RE
```

{NOTE: Ignore any "EOM before EOT" or "BOM before EOT" messages, this indicates a garbled message. }

If there were no fatal errors during the preprocess run, you may delete or rename the MESSAGES.DAT file as shown in the previous example.

5.3 Doing a Full Update

5.3.1 Overview

The "Full" update is the means by which new messages and statistics are added to the existing database. Only during the actual inclusion of new messages into the database will users be prevented from doing retrievals - about one to three minutes.

5.3.2 Running the FULL Update command file

The update process consists of sixteen programs run in sequence from a command file. There are given below three ways to perform a full update. Which one to use will depend on the System Manager. Use only one of them at a time.

All METER database files for the message analyzer are labeled with .P01 corresponding to the most recent and .P0k to the least recent. When room has to be made for incoming messages, the messages from the least recent update must be dropped.

5.3.2.1 Full Update with Purging.

Execute the following sequence to update a database in directory [270,5] with new messages while dropping off messages from the oldest update k. (The value of k is the same as the value for "THE NUMBER OF UPDATES IN THE METER WINDOW" stored in the METER control file.)

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> @[270,1]FULLUPD
;**********************
;* METER FULL UPDATE *
;**********************
;
IS THIS A RESTART ? [Y/N]: N    { Answer NO }
...... Several pages of output will appear here .....
```

5.3.2.2 Full Update with Purging from Offloaded Files.

If the files for the messages to be purged have been written out to a magnetic tape now on drive MM0:, do this:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> MOU MM0:UPDATE
MCR> PIP SY:=MM0:*.*
MCR> @[270,1]FULLUPD
;**********************
;* METER FULL UPDATE *
;**********************
;
IS THIS A RESTART ? [Y/N]: N    { Answer NO }
..... Several pages of output will appear here .....
MCR> DMO MM0:

{ See section 8.4.3.2 for steps on loading messages to tape }
```

The tape on drive MM0: now contains offloaded files for messages entering METER at the preceding update. This tape should be saved for the time when these messages will be purged.

5.3.2.3 Full Update with Purging Only.

This drops the messages from the k-th update from a database without adding new messages.

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> [270,1]UPDROP
*****************************************
* DROP ONE DAY FROM THE METER DATABASE *
*****************************************
..... Command File Output Appears Here .....
```

5.4 Doing a Short Update

5.4.1 Overview

Short updates are useful in situations where messages must be entered into the database, but there isn't enough time for a full update. The short update indexes the new messages using existing statistics. This means new messages can be made available for retrievals in a matter of minutes. When the next full update is run, new statistics will be generated for these messages, and added to the database.

5.4.2 Running the Short Update Command File

To run the short update command file enter the following from the METER console on the PDP-11/45:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> @[270,1]SHORTUPD
**********************
* METER SHORT UPDATE *
**********************

IS THIS A RESTART ? [Y/N]: N    { Answer NO }
..... Command File Output Appears Here .....
```

The new messages should now be ready for retrievals.

5.5 Predicting Update Times

To determine the approximate update time for a given number of messages, enter the following:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,1]
MCR> RUN ESTIME<esc>
Enter update Parameters in units of 1000      { Program Output }
number of messages in update:  xxx
total number of messages after update:  yyy
total number of index stems:  zzz


Expected Time for Update

stem inversion                    n.nn hours (= nnn.n minutes)
stem statistics                   n.nn hours (= nnn.n minutes)
index stem selection              n.nn hours (= nnn.n minutes)
stem association                  n.nn hours (= nnn.n minutes)
message indexing                  n.nn hours (= nnn.n minutes)

total time =nnn.n hours
(disregarding Message Acceptor processing)


{ Where xxx, yyy, zzz are input values in units of 1000, and
  nnn.n and n.nn are resulting output }
```

The following is the output from an actual run of ESTIME.

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,1]
MCR> RUN ESTIME<esc>


Enter update Parameters in units of 1000
number of messages in update:  .400          { 400 messages }
total number of messages after update:  5.5  { 5,500 after update }
total number of index stems:  3.2            { 3,200 stems total }


Expected Time for Update

stem inversion                    0.02 hours (=   1.6 minutes)
stem statistics                   0.00 hours (=   0.4 minutes)
index stem selection              0.06 hours (=   4.1 minutes)
stem association                  0.24 hours (=  14.7 minutes)
message indexing                  0.08 hours (=   5.3 minutes)

total time =  0.4 hours
(disregarding Message Acceptor processing)
```

The count of the number of messages in the update can be obtained from the run of QSTM in the preprocessing phase. The values for the total number of messages, and total number of index stems are contained in the control file.

The update times will not be exact. Control file parameter settings, system load, number of users, etc. will directly affect the actual time needed for a update.

6. GENERATING DATABASE SUMMARIES

6.1 Overview

Although knowledge of the contents of a METER database is not necessary for performing retrievals, it is helpful if the analyst has some idea of the types of messages he can expect to find. This information can be provided in the form of database summaries.

Summaries are generated after updates and involve running a series of programs to access the database statistics to produce reports which analysts will be able to use. These summaries may also be generated on request as well.

6.2 Running the Summary Generation Command File

To generate summaries enter the following on the PDP-11/45 METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> @[270,1]SUMMARY


Summaries Available:
        0) Quit, Return to MCR mode
        1) Stem Clusters
        2) Stem Associations
        3) Top Stems
        4) Stem Changes Since Last Update
        5) Stem Trends

Select Option [D R:0-5 D:0]:___ {Enter your choice}
```

6.2.1 If You Selected Main Menu Option: 0

If you have completed generation of database summaries, or if you have invoked this command file by accident; entering zero will return you to the "MCR>" prompt.

EXAMPLE:

```
0) Quit, Return to MCR mode
1) Stem Clusters
2) Stem Associations
3) Top Stems
4) Stem Changes Since Last Update
5) Stem Trends

Select Option [D R:0-5 D:0]:0
..... Some Command File Output Appears Here .....
MCR>
```

6.2.2 If You Selected Main Menu Option: 1 - Clustering

The following sub-menu will be displayed:

```
Select Clustering Method
        0) None, Return to Main Menu
        1) Mutual Nearest Neighbor
        2) Minimal Spanning Tree
        3) Tukey Gapping

Method [D R:0-3 D:0]:___    {Enter one of the above}
```

6.2.2.1 If You Selected Clustering Option: 0 - None

You will be returned to the main menu without any processing.

6.2.2.2 If You Selected Clustering Option: 1 - Mutual

The following will be output:

Enter upper limit for mutual neighbor: 0.__

Enter upper limit on number of associations
for stem to be clustered without restriction: __

{The above values will be supplied by the METER operator.}
{ Below is the resulting output }

n mutual pairs found.

m of these rejected.

There were xxx clusters formed.
Largest had yyy items.

Print stem clusters.

Enter minimum cluster size: __

{ n, m, xxx, and yyy outputs will vary with input values.}

6.2.2.3 If You Selected Clustering Option: 2 - Minimal

The following will be output:

Enter lower threshold for associations
(give TWO digits): 0.__

Enter upper limit on number of associations
for stem to be clustered without restriction: __

{The above values will be supplied by the METER operator.}
{ Below is the resulting output }

nnnnn stem pairs below threshold.

There were xxx clusters formed.
Largest had yyy items.

Print stem clusters.

Enter minimum cluster size: __

{ nnnnn, xxx, and yyy outputs will vary with input values.}

6.2.2.4 If you Selected Clustering Option: 3 - Tukey

The following will be output:

Enter Lower threshold for associations: 0.__

Enter upper limit on number of associations
for item to be clustered without restriction: __

{The above values will be supplied by the METER operator.}
{ Below is the resulting output }

mm percent of associations over threshold accepted.

nnn pairs with both items over limit on number of associations.

There were xxx clusters formed.
Largest had yyy items.

Print stem clusters.

Enter minimum cluster size: __

{ mm, nnn, xxx, and yyy outputs will vary with input values.}

6.2.3 Supplying Input to Clustering Programs

The size and number of clusters is determined by the input values supplied to the clustering programs. Since the data base is constantly changing, entering the same values on two different days may yield completely different clusters.

To generate clusters; the operator will enter a series of values to the same clustering program, and select the output that appears to contain the most information. Since the clustering programs run quickly, this should not take much time.

6.2.4 If You Selected Main Menu Option: 2 - Associations

The following to be output:

```
Enter minimum association to be printed.
(Specify TWO digits):  0.__

{The above value will be entered by the METER operator.}
{Below is the resulting output}

Note: only stems with associations
      above 0.xx (yyy) will be output

Row   100: nn1
Row   200: nn2
Row   300: nn3
      .
      .
      .
Row  zzzz: nnn          {Row = Stem number}

There are mmm zero rows.

There are nnnn stems with associations.
The average number of associations per index stem is  mm.mm

{xx, yyy, nn1-nnn, zzzz, mmm, nnnn, and mm.mm will vary with input}
```

6.2.5 Supplying Input for Stem Associations

The size of the stem associations report will be determined by the value entered. For more associations, enter a smaller value such as: .2, .4, .6 etc.. For less associations, enter larger values such as: .7, .8, .9 etc.. In other words, the smaller the value the larger the report.

6.2.6 If You Selected Main Menu Option: 1, 2, 3, 4, or 5

Selecting options 1 or 2 required you to supply input to various programs. Options 3, 4, and 5 do not require input. In all cases however, the following will always be displayed:

```
Do you wish the report to be printed on your terminal or the printer?

      0) Neither, Return me to the previous menu.
      1) Print the report on my terminal.
      2) Print the report on the printer.

Select Option [D R:0-2 D:0]:___      { Enter your choice }
```

6.2.6.1 If you Selected Print Option: 0 - Neither

Entering zero returns you to the previous menu without printing any reports.

6.2.6.2 If you Selected Print Option: 1 - Display to Terminal

Entering 1 will cause the report to be printed immediately following the carriage return. After the report is finished printing, you will be returned to the main menu.

6.2.6.3 If You Selected Print Option: 2 - Display to Printer

You will see the message: "The report is on the printer." prior to the main menu being redisplayed. This allows you to continue doing work without having to wait for the report to finish printing.

6.3 Command File Messages

During the execution of the command file you may notice messages from the ".AT" processor indicating programs are "DELAYING" or "WAITING". These will occur in the middle of the outputs shown above. Messages of this type are normal and should be ignored.

7. EDITING THE CONTROL FILE

7.1 Overview

The METER control file contains information that is common to all subsystems. By modifying its contents, you can modify the way METER process or retrieves messages.

7.2 Displaying Control File Contents

To display the current contents of the control file, enter the following on the PDP-11/45 METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> RUN [270,2]DMPCON<esc>
***** Program Completed *****

{ To Type the contents on your terminal enter: }

^C                                          { Enter Control-C to get MCR}
MCR> PIP TI:=CONTROL.TXT

{ or, to display the contents on the printer: }

^C                                          { Enter Control-C to get MCR}
MCR> QUE CONTROL.TXT
```

7.3 Editing the Control File

To edit the control file, enter the following on the PDP-11/45 console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> EDI CONTROL.TXT                  { or EDT, TECO etc..}
{... Make necessary changes....  (See Sample Control file) }
MCR> PIP CONTROL.TXT/PU
MCR> PIP TI:=CONTROL.TXT          { or MCR> QUE CONTROL.TXT}
```

It is a good idea to make a hard copy of the control file after you make any changes.

7.4 Creating a New Control File

To install the new control file, enter the following:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> RUN [270,2]MAKCON<esc>
***** Program Completed *****
```

7.5 A Sample Control File

The following is a sample control file. Areas which may or may not be edited are indicated.

```
!METER system control file as of 14-JAN-82       10:44:43
!Database directory
[27C,5]                                                    {Don't Modify}
!Base number for labeling messages
0                                                          {Don't Modify}
!Minimum association threshold
0.20                                                       {May be Modified}
!Association weight in queries
0.50                                                       {May be Modified}
!Context weight in queries
0.33                                                       {May be Modified}
!Message Acceptor Lock
unlocked                                                   {Don't Modify}
!Message Analyzer Lock
unlocked                                                   {Don't Modify}
!Number of updates in the METER window
14                                                         {May be Modified}
!Maximum number of index words
3500                                                       {May be Modified}
!Number of index words in current update
0                                                          {Don't Modify}
!Number of index words for retrievals
3186                                                       {Don't Modify}
!Number of message batches since the last full database update.
0                                                          {Don't Modify}
!Number of message batches currently being processed.
0                                                          {Don't Modify}
!Number of full updates processed by METER so far...
4                                                          {Don't Modify}
!Lower message count index threshold
10                                                         {May be Modified}
!Upper message percentage index threshold
50                                                         {May be Modified}
!Minimum message count for association
10                                                         {May be Modified}
!Maximum message percentage for association
50                                                         {May be Modified}
!Daily word co-occurrence threshold
10                                                         {May be Modified}
!Date/Time of the last message batch - MM/DD/YY/Time
!       Where: Time = minutes after midnight.
3                                                          {Don't Modify}
18                                                         {Don't Modify}
82                                                         {Don't Modify}
```

```
627                                                    {Don't Modify}

!Base year for encoding header date/time
82                                                     {May be Modified}
!Number of messages processed by the message acceptor
0                                                      {Don't Modify}
!Offset into micro-concordance file as of
!   the last short update
0                                                      {Don't Modify}
!Number of messages entered into the database
!    by short update
0                                                      {Don't Modify}
!Number of messages in each of the METER full updates
! (Starting with the most recent update.)
0                                                      {Don't Modify}
428                                                    {Don't Modify}
450                                                    {Don't Modify}
397                                                    {Don't Modify}
449                                                    {Don't Modify}
```

Changes to the control file parameters will be determined by the METER system manager.

8. ERROR CORRECTION

This section is reserved for people with an understanding of METER internals (METER programmers). Use of corrective procedures contained within this section by individuals without such knowledge may result in serious damage to the METER database. This section does contain however, many tips on avoiding problems with METER.

Operators should refer to Appendix E for information regarding METER operation.

8.1 Overview

This section covers the steps to correcting the errors that may occur under the METER system.

8.2 Start Up Errors

8.2.1 How to Avoid Them

- Make sure MAXI is running
- Make sure the three command files are run.
- If METER is taken down or disabled, either restart it, or "remove" QGATES and METERM from the PDP-11/70. This is doubly important if QGATES has been set to run automatically.

8.2.2 How to Fix Them

8.2.2.1 Can't Install Tasks

This may occur if there is insufficient room in the task tables to install METER tasks. Since the probability of program development on an active MAXI/METER system is slight; you may consider deleting unneeded tasks. This may include: "...F4P", "...SRD", "...EDT",

EXAMPLE:

QGATES cannot be installed so we elect to remove the "SRD" program to make room. So we type:

```
MCR> HELLO [1,1]
MCR> REM ...SRD
MCR> INS [27C,10]QGATES/UIC=[1,4]
```

Appendix D contains a table of installed METER tasks and their UIC's which they are to be installed under. Refer to this if you ever have to manually install a task.

8.2.2.2 Can't Run Retrievals

METER may not retrieve for the following reasons:

- METER was not started.
- Insufficient space for METER tasks during the start up procedure.
- Missing files in the users or database directory.

- Insufficient disk space.
- METER did not find any messages that related to the analyst's query.
- Program bugs.
- Very slow response time due to insufficient memory to run retrieval tasks. The task will eventually run when current tasks terminate.

8.2.2.2.1 Finding the cause

Causes for retrievals not running fall into two areas: Global and local. Global causes affect all users while local causes may effect only one or two users.

8.2.2.2.1.1 Local Causes

This is by far the easiest to correct. If only one or two users are experiencing retrieval difficulties the most likely area for problems will be in the individual user directories. Verify that the user directory is correctly set up by entering the following:

```
MCR> HELLO [1,1]    {On the PDP-11/45}
MCR> SET /UIC=[277,277]
MCR> PIP
PIP> [277,xxx]/DF
PIP> /FU
..... PIP Output Appears Here .....
PIP> TI:=DATABASE.UIC
..... PIP Output Appears Here .....
PIP> ^Z

{xxx is the users directory number}
```

The "PIP /FU" output should be examined to verify that the owner of the files is [277,277]. If they are not enter the following:

```
MCR> HELLO [1,1]    {On the PDP-11/45}
MCR> SET /UIC=[277,277]
MCR> QUE [277,xxx]EVALNS.DAT  {Save the output}
MCR> PIP
PIP> [277,xxx]/DF
PIP> *.*;*/DE
PIP> DATABASE.UIC=TI:
[270,5]
^Z
PIP> EVALNS.DAT=TI:
^Z
PIP> ^Z
```

{xxx is the users directory number}

This will establish an initial user directory with files having the correct ownership.

The "PIP TI:=DATABASE.UIC" should display "[270,5]" as the database location. If it does not, enter the following:

```
MCR> HELLO [1,1]    {On the PDP-11/45}
MCR> SET /UIC=[277,277]
MCR> PIP
PIP> [277,xxx]/DF
PIP> DATABASE.UIC;*/DE
PIP> DATABASE.UIC=TI:
[270,5]
^Z
PIP> ^Z
```

{xxx is the users directory number}

8.2.2.2.1.2 Global Causes

Global causes are more difficult to correct since they tend to indicate errors in the software instead of errors in setting up METER. To begin isolating the problem, enter the following in the PDP-11/45 METER console:

```
MCR> HELLO [1,1]
MCR> PIP [277,*]/FU
..... PIP Output Appears Here .....

MCR> PIP [270,1]COMMENTS.TXT/FU
..... PIP Output Appears Here .....

MCR> PIP [270,1]ERROR.RPT/FU
..... PIP Output Appears Here .....
```

Examine the outputs for the following:

- Ownership for all files in the 277 group must be [277,277]

- Protection for ALL files must allow Read, Write, Delete, and Extend access for GROUP and OWNER. The COMMENTS.TXT file must also allow WORLD the same access.

- All database UIC's must point to [270,5].

For steps to correct any errors you may find, refer to section 1.2.2.2.1.1 - "Local Causes".

8.3 Control File Errors

Control file errors are caused by:

1. A missing, locked, or invalid CONTROL.DAT file in [270,5].

2. Missing or invalid DATABASE.UIC file in the user's directory.

8.3.1 How to Avoid Them

- Make sure a CONTROL.DAT file exists in the database directory.
- Verify that the control file contains the correct values for the last update.
- Make sure the control file is not locked.
- Make sure each user has in his work directory a DATABASE.UIC file which points to the database directory.

8.3.2 How to Fix Them

8.3.2.1 Missing or Invalid CONTROL.DAT File

If this file does not exist, no retrievals or updates will take place. To create this file, enter the following on the PDP-11/45 METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> PIP CONTROL.*/LI     {See what "CONTROL" files exist}
..... PIP output appears here...


{ Examine the PIP output for a CONTROL.TXT file.  If one
  exists, enter the following:  }


MCR> PIP TI:=CONTROL.TXT          {If you are on a hard-copy terminal}
      -or-
MCR> QUE CONTROL.TXT              {If you are on a video terminal}


{ Compare this control file with the control file from the last
update.  If  these  are  the  same  then enter the following: }


MCR> RUN [270,2]MAKCON<esc>
***** Program Completed *****       {You now have a CONTROL.DAT file}
```

If they are different then you will have to edit the file to look like the last update. Perform the steps in section 7.3 and 7.4 of this document.

If there are no "CONTROL" files, you will have to create one using the sample CONTROL.TXT file in [270,2]. To do this enter the following:

```
MCR> PIP *.*=[270,2]CONTROL.TXT
MCR> QUE CONTROL.TXT              {or PIP TI:=CONTROL.TXT}
```

Perform steps 7.3 and 7.4 as indicated above.

8.3.2.2 Locked Control File

Should you get a message indicating the control file is "LOCKED", enter the following to unlock it:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> RUN [270,2]DMPCON<esc>
***** Program Completed *****
^C
MCR> RUN [270,2]MAKCON<esc>
***** Program Completed *****
^C
MCR> PIP CONTROL.*/PU
```

8.4 Disk Space Problems

8.4.1 How to Avoid Them

- Make sure that the update window is not too large - ie: Don't set a window size of twenty days if the analysts are only interested in messages that are no older then ten.

- Require the analysts purge their directories regularly.

- Make sure that the MESSAGES.DAT file is deleted after each successful update.

- Clean out temporary files from the database directory after generating summaries.

8.4.2 Making more room

8.4.2.1 Cleaning Up User Directories

Refer to section 3.3 for user directory purging.

8.4.2.2 Cleaning Up the Database Directory

To clean up the database directory, enter the following from th METER operator console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> PIP *.S01;*/DE
MCR> PIP *.T01;*/DE
MCR> PIP CLSTRALG.DAT;*/DE
```

8.4.2.3 Dumping the Error Message File

If the METER error message file becomes too large, enter the following on the METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,1]
MCR> PIP TI:=ERROR.RPT
..... PIP output appears here .....
MCR> PIP ERROR.RPT;*/DE
MCR> PIP ERROR.RPT=TI:
^Z
MCR> PIP ERRORS.RPT/PR:0

{ Save the Output }
```

8.4.2.4 Dumping the Comments File

If the comments file becomes too large, enter the following on the METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,1]
MCR> PIP TI:=COMMENTS.TXT
..... PIP output appears here .....
MCR> PIP COMMENTS.TXT;*=TI:
^Z
MCR> PIP COMMENTS.TXT/PR:0

{ Save the Output }
```

8.4.3 No Room for an Update

8.4.3.1 Dropping a Day

Perform the full update in section 5.3.2.4.

8.4.3.2 Off Loading Messages

To off load a day of messages, mount a scratch tape, and enter the following from the METER console on the PDP-11/45:

```
MCR> HELLO [1,1]
MCR> INI MM:xxxxxxx      { Assign a unique label }
MCR> MOU MM:xxxxxxx
MCR> PIP MM:*.PON=FDIP.POk
MCR> PIP MM:*.PON=FDIPX.POk
MCR> PIP FDIP.POk;/DE
MCR> PIP FDIPX.POk;/DE
```

{ xxxxxxx - A tape ID such as: APR1482 for April 14, 1982 }

{ k is any day greater then 1 and less then n; where n is the largest day }

8.5 Message Transmission Errors

8.5.1 How to avoid Them

- MAXI must be installed on both systems.
- METER MUST BE INSTALLED ON BOTH SYSTEMS OR MESSAGES MAY BE LOST.
- If either system is stopped, crashes, or is brought down, make sure METER and MAXI are started before attempting to send messages across ICC.
- After preprocessing messages, make sure the MESSAGES.DAT file is deleted or duplicate messages will result (Updates will also take much longer).
- If METER is to be brought down on the PDP-11/45 to free the system for other work, make sure to remove METER tasks from the PDP-11/70 as well.

8.5.2 QGATES

8.5.2.1 Won't Run

8.5.2.1.1 No room to install the QGATES task

This is possible on the PDP-11/70. To correct this you will have to "REMOVE" an un-needed task from the system. Possible programs for removal could be: ...F4P, ...EDT, ...EDI, ...SRD, ...TKB, ...BOO, etc. To remove a task follow the steps in section 8.2.2.1. Before removing any tasks check with the system manager. Please note: QGATES must be installed with the UIC of [1,4] so that it can delete the messages after reading them.

8.5.2.2 Won't Read Messages

8.5.2.2.1 Verifying Messages Exist

If QGATES is installed to run at scheduled intervals there may not be any messages in the queue at the time it starts to run. To verify that messages exist enter the following on the PDP-11/70 console:

```
MCR> HELLO [1,1]
MCR> PIP DU1:[270,100]/LI
..... PIP output appears here .....
```

If files are not displayed then there are no messages to send.

8.5.2.3 Can't Delete Messages

There may be several reasons why QGATES may not delete messages. The most common has to do with improper installation. This is because QGATES, unlike other METER tasks, is designed to be installed under different options.

8.5.2.3.1 QGATES is Installed Under the Wrong UIC

To insure that the QGATES task is correctly installed enter the following from the PDP-11/70 console:

```
MCR> HELLO [1,1]
MCR> REM QGATES
MCR> INS [270,10]QGATES/UIC=[1,4]

{ If the MESSAGES were transmitted then enter: }

MCR> PIP DU1:[270,100]/DE

{ If messages were not transmitted enter: }

MCR> RUN QGATES
```

Depending on the situation, duplicate message texts may be transmitted. This will not cause an error in METER, but will cause the next update to take longer.

8.5.3 QGATER

Correcting errors on this side of ICC will be more difficult since there is a greater chance for losing messages. You can minimize the chances for an error by putting a copy of the messages into a separate directory.

8.5.3.1 Won't Run

8.5.3.1.1 Verify that QGATER is Correctly Installed

To insure that QGATER is correctly installed you will have to manually re-install it. This requires you to remove it first. If QGATES is installed for automatic message transmission on the PDP-11/70 you should remove it before proceeding with the following steps.

To re-install QGATER, enter the following from the PDP-11/45 console:

```
MCR> HELLO[1,1]
MCR> REM QGATER
MCR> INS [270,10]QGATER/UIC=[270,5]
```

Please notice that QGATER is installed under the [270,5] UIC. If it is installed under any other UIC, protection errors will result. When protection errors occur, they are not obvious.

Once QGATER is installed, re-install QGATES on the PDP-11/70, and enter the following:

```
{ On a PDP-11/70 VIDEO terminal (VT52) }

MCR> HELLO [1,1]
MCR> DEM
..... The DEM program will activate .....


{On a PDP-11/45 VIDEO terminal (VT52) }

MCR> HELLO [1,1]
MCR> DEM
..... The DEM program will activate .....
```

For the next step you may need someone to help you. The reason for running the DEM programs is to allow you to follow the execution of the QGATE programs. When these program are running, they will be visible on the two VT52 terminal screens. In this way you can tell where the QGATE programs are failing.

To start the trace you and an assistant should be within view of the two terminals. Have another assistant enter the command to start QGATES from a third terminal. When this is done you should see "QGATES" appear on the screen of the terminal connected to the PDP-11/70. A few moments later you should see "QGATER" appear on the screen of the terminal connected to the PDP-11/45. If things are working, they should remain on the screen until all messages are passed over. If one or the other terminates early, that is where the problem is.

To start the trace, enter the following on the PDP-11/70 console:

```
MCR> HELLO [1,1]
MCR> RUN QGATES<esc>      { Make sure this was re-installed.}

{You should observe the activities on the screens }
```

If the failure occurred in the QGATES program, perform the steps in section 8.5.2. If the failure occurred in QGATER, continue with this section.

To remove the DEM screens to free the terminals, enter a ^Z on each terminal. This should return them to the MCR mode.

8.5.3.2 Won't Write Messages

8.5.3.2.1 Verifying Messages Exist

In the case of automatic message transmission; sometimes there are no messages to send when the QGATES program starts to run. Refer to section 8.5.2.1.1 to determine if this was the case.

8.5.3.2.2 QGATER is Installed Under the Wrong UIC

To insure that QGATER is correctly installed, enter the following from the PDP-11/45 console:

```
MCR> HELLO [1,1]
MCR> REM QGATER
MCR> INS [270,10]QGATER/UIC=[270,5]
```

Try sending messages across. If this fails, continue to the next step.

8.5.4 Lots of Duplicate Messages

Duplicate messages are caused by writing into a MESSAGES.DAT file that has already been preprocessed. This is not generally an error, but will cause the update programs to run longer. To avoid this problem make sure the MESSAGES.DAT file is re-named or deleted after it has been pre-processed.

8.5.5 Messages Never Got to METER

This can be caused by:

- Improperly installed QGATER, or QGATES programs.
- A system failure on the PDP-11/45 with QGATES still installed on the PDP-11/70. (If METER is disabled on the PDP-11/45 the same effect will result.)
- No messages being sent to METER by MAXI.
- No one starting up METER, or requesting QGATES start up.

By following the steps in section 8.5.2 through 8.5.4 you should have eliminated most of the above causes. If problems still exist, check your scheduling of the QGATE programs. If all else fails, call P.A.R..

8.6 Pre-Processing Errors

8.6.1 How to Avoid Them

- Make sure QGATE programs are properly installed for message collection.
- Always delete the MESSAGES.DAT file after preprocessing.
- Make sure the control file is not locked.
- Make sure there is a MESSAGES.DAT file to preprocess.

8.6.2 How to Fix Them

8.6.2.1 QSTM

8.6.2.1.1 Control File is Locked

If the control file is locked due to a previous error, perform the steps in section 8.3.2.2 before running the preprocessing command file.

8.6.2.1.2 No MESSAGES.DAT File

If no messages were transmitted since the last update you will not have a file to preprocess. If QGATES was run in a directory containing messages, and the file was still not created; refer to section 8.5 for corrective steps.

8.6.2.1.3 EOM & BOT Errors

These messages indicate METER could not locate the standard start or end of a message. This is not a serious error. If the most of the messages have this error, it indicates that the pattern recognizer in QSTM needs to be modified. Print copies of the header and trailer for a few of these messages, and call P.A.R. for the necessary corrections. You can still do updates, but when these messages are retrieved they may contain more information then the analyst may want to see.

8.7 Update Errors

8.7.1 How to Avoid Them

- Run DMPCON after each successful update. This will save an image of a correct control in case the next update fails.
- Make sure there were no errors during preprocessing before starting a "Full" or "Short" update.
- If the control file was locked due to a previous error, unlock it before continuing.
- Make sure the control file is set with the necessary parameters.
- Make sure you preprocessed messages before running the "Full" or "Short" update command file.

8.7.2 How to Fix Them

If the messages were transmitted and preprocessed without error, you should not have much problem running the update command files. If an error is encountered in a update program, it will exit - leaving the control file locked to prevent the remaining update programs from running. To restart the update: correct the error, and enter the following from the METER console on the PDP-11/45:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> RUN [270,2]MAKCON<esc>      { This unlocks the control file }
***** Program Completed *****
^C
MCR> @[270,1]FULLUPDS
;**********************
;*  FULL METER UPDATE *
;**********************
;
IS THIS A RESTART? [Y/N]: Y     { Reply YES to restart }
;
;-----------------------------------------------------
; IF YOU HAVE NOT RUN MAKCON, EXIT THIS COMMAND FILE
;-----------------------------------------------------
;
; SELECT ONE OF THE FOLLOWING:
;
;       0) EXIT COMMAND FILE
;
;       1) BEFORP      5) PCST       9) PTAS        13) PCMV
;       2) PMRJ        6) PCSORT    10) PRAM        14) PWEIGHT
;       3) PRSV        7) PCLS      11) PFILLSORT   15) PREDUCE
;       4) PPCTM       8) PSQZ      12) PFILLBLD    16) AFTERP
;
ENTER NUMBER OF THE PROGRAM TO BE RESTARTED [D R:0-16 D:0]: ____
..... Rest of the Command File Output Appears Here .....
```

Error correction in a update command file is not as easy as in previous command files. When errors occur, the reason may not be obvious. The majority of the errors which you may see will be missing files, or a locked control file. These missing files are usually tables used by METER for message analysis. The following is a list of files which must be present in the database directory for an update to run successfully:

```
CONTROL.DAT
DATABASE.UIC              { Must point to [270,5] }
MESSAGES.DAT     { Only if you want to preprocess messages. }
NEWSOURCE.TBL
SOURCE.TBL
STOPWORD.TBL
SUFFIX.TBL
```

Of course, the fact the the files exist does not insure a successful update. The files must contain information which is in a form that METER can use. If the files become corrupted or lost you can PIP copies over from directory [270,2].

If the restart still fails you can delete all "P00" files and start the update from the beginning. To do this, enter the following from the METER console on the PDP-11/45:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> PIP *.P00;/DE
MCR> RUN [270,2]MAKCON<exe>
***** Program Completed *****
^C
MCR> @[270,1]FULLUPD    { or SHORTUPD if you were running that}

Is this a restart? (Y/N): N        { Answer NO }

..... Command file Output Appears Here .....
```

The MAKCON program will re-create the CONTROL.DAT file that existed prior to running the full update. If you run DMPCON and then makcon, the control file will be at the state it was when the failure occurred.

9. BUILDING METER SOFTWARE

9.1 Creating METER Directories

To create the necessary METER directories, enter the following on the PDP-11/45 console:

```
MCR> HELLO [1,1]
MCR> INS [11,1]UFD   {If it isn't currently installed}
MCR> UFD SY:[270,1]/PRO=[RWED,RWED,,RWED,RWED]
MCR> UFD SY:[270,2]
MCR> UFD SY:[270,3]
MCR> UFD SY:[270,4]
MCR> UFD SY:[270,5]/PRO=[RWED,RWED,RWED,RWED]
MCR> UFD SY:[270,10]
MCR> UFD SY:[270,12]
MCR> UFD SY:[270,14]
MCR> UFD SY:[270,16]
MCR> UFD SY:[270,20]
MCR> UFD SY:[270,270]
```

9.2 Loading METER from Tape

Mount the distribution tape and enter the following from the PDP-11/45 console:

```
MCR> HELLO [1,1]
MCR> LOA MM
MCR> INS [11,1]FLX     (If not currently installed)
MCR> MOU MM:/CHA=[FOR]
MCR> FLX SY:/RS/UIC=MM:[270,*]*.*/DO
```

9.3 Compiling, Building Libraries, and Linking

Enter the following from the PDP-11/45 console:

```
MCR> HELLO [1,1]
MCR> INS [11,1]F4P
MCR> INS [11,1]LBR
MCR> INS [11,1]TKB
MCR> INS [11,1]MAC
MCR> SET /UIC=[270,1]
MCR> @COMPILE
MCR> SET /UIC=[270,2]
MCR> @COMPILE
MCR> SET /UIC=[270,3]
MCR> @COMPILE
MCR> SET /UIC=[270,4]
MCR> @COMPILE
MCR> SET /UIC=[270,10]
MCR> @COMPILE
MCR> SET /UIC=[270,12]
MCR> @COMPILE
MCR> SET /UIC=[270,14]
MCR> @COMPILE
MCR> SET /UIC=[270,16]
MCR> @COMPILE
MCR> SET /UIC=[270,20]
MCR> @COMPILE
MCR> SET /UIC=[270,3]
MCR> @STORE
MCR> SET /UIC=[270,4]
MCR> @STORE
MCR> SET /UIC=[270,20]
MCR> @STORE
MCR> SET /UIC=[270,1]
MCR> @LINK
MCR> SET /UIC=[270,2]
MCR> @LINK
MCR> SET /UIC=[270,10]
MCR> @LINK
MCR> SET /UIC=[270,12]
MCR> @LINK
MCR> SET /UIC=[270,14]
MCR> @LINK
MCR> SET /UIC=[270,16]
MCR> @LINK
MCR> SET /UIC=[270,20]
MCR> @LINK
MCR> SET /UIC=[270,270]
MCR> PIP *.*=[270,10]*.TSK/RE
MCR> PIP *.*=[270,12]*.TSK/RE
```

9.4 Initial Database Creation

9.4.1 Editing Message Sources / Creating the Source Table

One function of METER allows the analyst to limit retrievals to messages with certain characteristics. This includes: classification, date-time group, and source. For METER to limit retrievals to a specific source, a table must be created. METER allows up to 127 sources to be entered into the table. Each source can contain up to eight characters.

To create the source table, enter the following on the PDP-11/45 METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,2]
MCR> RUN DMPSOURCE<esc>
..... DMPSOURCE Output Appears Here .....
^C
MCR> EDI SOURCE.TXT
{ Edit the file to insert source names }
MCR> RUN MAKSOURCE<esc>
..... MAKSOURCE Output Appears Here .....
^C
MCR> SET /UIC=[270,5]
MCR> PIP SOURCE.TBL=[270,2]SOURCE.TBL
```

{ A source must end with an asterisk(*) when it is less than 8 characters in length.}

You have the option to re-edit the source.txt file at any time. Each time the MAKSOURCE program is run it will re-map the sources in the database to reflect the changes you made.

If you just want a listing of the current sources, run DMPSOURCE and print out the SOURCE.TXT file.

9.4.2 Editing the Stopword Table

To edit the stopword table enter the following:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,2]
MCR> EDI STOPWORD.TXT
..... Edit the Table .....
```

If you have previously created a stopword table, and installed it in the database directory, enter the following:

```
MCR> RUN [270,2]STOPBUILD<esc>
.... STOPBUILD Output Appears Here .....
^C
MCR> SET /UIC=[270,5]
MCR> PIP *.*=[270,2]STOPWORD.TBL
```

Otherwise, proceed to the next section.

9.4.3 Creating Initial Database Files

Enter the following from the PDP-11/45 console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,2]
MCR> RUN SUFBUILD<esc>
..... SUFBUILD Output Appears Here .....
^C
MCR> RUN STOPBUILD<esc>
..... STOPBUILD Output Appears Here .....
^C
MCR> SET /UIC=[270,5]
MCR> @[270,1]INIDBS
MCR> PIP DATABASE.UIC=TI:
[270,5]
^Z
MCR> EDI CONTROL.TXT
{ Set up the control file for your message collection needs }
MCR> RUN [270,2]MAKCON<esc>
***** Program Completed *****
^C
MCR> SET /UIC=[270,1]
MCR> PIP COMMENTS.TXT=TI:
^Z
MCR> PIP ERRORS.RPT=TI:
^Z
MCR> PIP COMMENTS.TXT/PR:0
MCR> PIP ERRORS.RPT/PR:0
```

Proceed to "ENTERING NEW USERS", "STARTING UP METER", and "DOING AN UPDATE"

10. METER DIRECTORIES

In the MAC/IN implementation of METER, the number "270" will be the METER group number for RXS UIC's. METER file directories will be set up as follows (with scope of interest given in parentheses).

[270,1] (METER system programmer, operator)
This is for METER program development and maintenance; command files for setting up METER will also be here.

[270,2] (METER system manager)
This directory contains the master METER users file, various utilities, and files required for compiling METER programs.

[270,3] (METER system programmer)
For program development and maintenance. This will contain libraries for linking METER programs.

[270,4] (METER system programmer)
For program development and maintenance. This will contain libraries for linking METER programs.

[270,5] (METER system manager, operator, and users)
Contains all current files for the principal METER database.

[270,6] (METER system manager, programmer)
For development of test databases.

[270,7] (METER system manager, programmer)
For development of test databases.

[270,10] (METER system operator, programmer)
Contains all source files for the METER message acceptor.

[270,12] (METER system operator, programmer)
Contains all source files for the METER message analyzer.

[270,14] (METER system operator, programmer)
Contains all source files for the METER message extractor.

[270,16] (Meter system manager, operator, programmer)
Contains all source files and executable programs for the METER message summarizer.

[270,20] (METER system programmer)
For program development and maintenance. This will contain libraries for linking METER programs.

[270,270] (METER system users, operator) This will contain all installed METER task images for current operation.

[277,*] (METER system users, manager, operator)
For files generated by METER users during retrievals. These are controlled by the METER system manager. Each directory corresponds to an RSX/IAS UIC assigned to each user.

Other [270,*] file directories will be established as needed to support METER system development.

11. SAMPLE UPDATE

Here is the output from a sample run of a full update. For clarity, all command file messages related to the "AT" processor "Delaying" or "Continuing" have been eliminated. Long lists of information displayed by some of the programs have been reduced to save space.

(Notice the recovery from the incorrect message by CSTM.)

```
MCR>@[270,1]PREPROC
>;*******************************
>;* PRE-PROCESS METER MESSAGES *
>;*******************************
>;
>RUN [270,2]DMPCON
***** Program Completed *****
>;
>RUN [270,270]QSTM

QSTM: Message Text Reduction
----------------------------

        Current Processing Status:

        Message Number  Elapsed Time (in seconds)
        --------------  ------------------------
              20                 46.1
              40                104.0
              60                144.8
              80                221.3
             100                275.7
             120                342.4
             140                404.2
             160                464.4
             180                668.4
             200                732.2
Message 218: EOM before EOT
             220                781.1
             240                818.3
             260                853.5
             280                887.5
             300                945.7
             320                974.7
             340               1000.7
             360               1079.5
             380               1114.6
             400               1143.8
             420               1183.2
             440               1216.6

        QSTM Summary
        ------------
        Number of messages processed:                    448
        Number of messages with format errors:             1

        Total number of words processed:              155407
        Average number of words/message:                 346
        Number of stopwords:                           77626
```

```
        Stopword percentage of total words:                    49

        Total number of raw stems produced:                 77781
        Average number of raw stems/message:                  173

        Message 119 had the most (353) raw stems.

        Elapsed time: 20 minutes 59.8 seconds
        Average processing time/message: 2.81 seconds


>RUN [270,270]QNVRT

QNVRT: Message-Stem File Inversion
----------------------------------
        merge 17 files with pattern = 9  8  0

>;
>;***** PREPROCESSING COMPLETED *****
>;
>; (IF THERE WERE NO ERRORS YOU MAY DELETE, OR RENAME THE MESSAGES.DAT FILE)
>;
>@<EOF>
MCR>SET /UIC=[270,5]
MCR>@[270,1]FULLUPD
>;**********************
>;* METER FULL UPDATE *
>;**********************
>;
>RUN [270,2]DMPCON
***** PROGRAM COMPLETED *****

>RUN [270,270]BEFORP

BEFORP: FULL UPDATE PREPARATION


>RUN [270,270]PMRJ

PMRJ: MERGE INVERTED FILES
        merge 1 files with pattern = 1  0  0


>RUN [270,270]PRSV

PRSV: GET STEM VECTORS
        Number of stem occurrences:     52320
        Number of unique stems:          7524
        MAX NT=248 FOR NEW      (4639)
```

```
        MAX FT=438 FOR NEW      (4639)
        MAX JT=944 FOR NEW      (4639)
        Total elapsed time: 1 minutes, 42.30 seconds


>RUN [270,270]PPCTM

PPCTM: CONTENT MEASURE UPDATE
        Total Number of Stems:          7524
        Number Dropped:                    0
        Number Added:                   7524


>RUN [270,270]PCST

PCST: Index Stem Selection
        Allowed number of index stems:  3500
        Number Actually chosen:         1261

        Average Content Measure:        0.196

        Number of Associated Stems:     1261
        Number Over Association Limit:     0


>RUN [270,270]PCSORT

PCSORT: Index Stem Sort
        merge 1 files with pattern = 1  0  0


>RUN [270,270]PCLS

PCLS: Index Stem Correlation
        Number of stems processed:      1261


>RUN [270,270]PSQZ

PSQZ: Stem Vector Compression
        Number Compressed Records:      7518

>PIP FCSV.P00;1/UP=FCSV.P00
>PIP FCSV.P00;/DE
>RUN [270,270]PTAS

PTAS: Inner Product Computation
        Number of rows to compute:      1261
        Current Row:
```

```
Row 1
Row 2
Row 3
Row 5
Row 8
Row 13
Row 21
Row 34
Row 55
Row 89
Row 144
Row 233
Row 377
Row 610
Row 987

Inner Product Threshold:                10
IP Percentage Above Threshold:          13
Number computed:                    106374
Average value:                          21

Largest IP value was 458 for stem numbers 164 and 871
(CART     and PRESIDE )

Total elapsed time: 15 minutes, 52.40 seconds

>RUN [270,270]PRAM

PRAM: Association Measure Computation

        Row Item        # of Associated Stems
        ------------    ---------------------
              1                   123
             40                    38
             79                    23
            118                   276
            157                   120
            196                   118
            235                   157
              .                     .
              .                     .
              .                     .
           1132                     4
           1171                    14
           1210                     3
           1249                     0
        Number Rows processed:                  1261
```

```
        Minimum Association Threshold:          0.20

        Percentage Above Threshold:               47
        Number of inner products:              50569
        Average association:                    0.30

        Average number Per Item:                  80

        Maximum association was 1.00 for stem numbers 259 and 350
        (CRATE     and DUBAWNT )

        Total elapsed time: 8 minutes, 10.47 seconds


>RUN [270,270]PFILLSORT

PFILLSORT: Fill Association Matrix
        merge 12 files with pattern = 9  3  0


>RUN [270,270]PFILLBLD

PFillEld: Build Association Matrix
        Number of associations:                        101138
        Total elapsed time: 4 minutes, 27.92 seconds


>RUN [270,270]PCMV

PCMV: Message Indexing

        Day 0

                Message 119 - 279 Index Stems
                0.27 Secs/Msg

        Number of Days Processed:                           1
        Number of Messages:                               448
        Number with no Index Stems:                         0
        Index Stems Per Message:                        81.82

        Message 119 had the most index stems (279)

        Processing time in minutes:                      1.97
        Time Per Message in Seconds:                     0.26


>RUN [270,270]PREDUCE
```

```
PREDUCE: Reducing Stem Associations

        Row    1
        Row    40
        Row    79
        Row    118
        Row    157
        Row    196
        Row    235
                .
                .
                .
        Row    1132
        Row    1171
        Row    1210
        Row    1249

        Elapsed Time = 1.19 Secs.


>RUN [270,270]PWEIGH

PWEIGHT: Weighting Stems
        Elapsed Time = 1.06 Secs.

>REM METERD
>PIP FFF.FFF/RE=FCSV.P00;1
>PIP GGG.GGG/RE=GCMV.P00;1
>RUN [270,270]AFTERP

AFTERP: Complete a Full Update

>PIP GGG.GGG/UP=GCMV.P01
>PIP GCMV.P01;/DE
>PIP GCMV.P01;1/RE=GGG.GGG
>PIP FCSV.P00;1/RE=FFF.FFF
>INS [270,20]METERD/UIC=[277,277]
>;
>;
>;          ******* FULL UPDATE COMPLETEED *******
>;
>;
>; (Dont Forget to run DMPCON if this Update was successful)
>@ <EOF>
MCR> RUN [270,2]DMPCON
***** Program Completed *****
```

12. TABLE OF INSTALLED METER PROGRAMS AND FILES

12.1 Installed METER Programs

Program	System	UIC
QGATES	11/70	[1,4]
METERM	11/70	[277,277]
QGATER	11/45	[270,5]
METERD	11/45	[277,277]
METERZ	11/45	[277,277]
BUILD	11/45	[277,277]
COLLECT	11/45	[277,277]
COMMENT	11/45	[277,277]
CULL	11/45	[277,277]
DISPLAY	11/45	[277,277]
GET	11/45	[277,277]
ORDER	11/45	[277,277]
REPORT	11/45	[277,277]
SHOW	11/45	[277,277]

12.2 Important METER Files

File Name	System	Directory	Owner	Protection
USERS.DAT	11/70	[270,1]	[270,1]	
COMMENTS.TXT	11/45	[277,1]	[270,1]	[RWED,RWED,RWED,RWED]
ERROR.RPT	11/45	[277,1]	[270,1]	[RWED,RWED,RWED,RWED]
DATABASE.UIC	11/45	[277,*]	[277,277]	[xxxx,RWED,RWED,xxxx]
EVALNS.DAT	11/45	[277,*]	[277,277]	[xxxx,RWED,RWED,xxxx]
All Other files in 277,* user directories	11/45	[277,*]	[277,277]	[xxxx,RWED,RWED,xxxx]

* - Any legal directory number under group 277

xxxx - Can have any protection setting

13. ADDITIONAL OPERATOR INSTRUCTIONS

This section consists of a revised list of METER operating instructions. It is intended to replace Section 8 error correction procedures. Unless specifically instructed to perform a task in section 8, you should only refer to this section.

13.1 Revised Operating Procedures for METER

1. Under no circumstances should you halt an update before it finishes. Let the command file run to completion.

2. If an update is interrupted by a system crash, power failure, or other external causes, unlock the control file and restart it from where it was interrupted. To do this enter the following on the PDP-11/45 METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> PIP *.P00;*/LI              { Check for locked files and
                                     delete any found }

MCR> RUN [270,2]DMPCON<esc>      { Do not edit the control file }
***** Program Completed *****
^C
MCR> RUN [270,2]MAKCON<esc>
***** Program Completed *****
^C
MCR> @[270,1]FULLUPD             { Answer YES to the restart
                                     question }
NOTE:  You should restart an interrupted update even though
       the message analyzer programs have been locked.
```

3. If a METER update locks because of a -26 error (No Such File), it is probably due to a sporadic UIC problem. In this case, reset your terminal UIC to [270,5], make sure that all files in [270,5] are owned by [270,5], unlock the control file, and restart the update from where the -26 error occurred. To do this enter the following from the PDP-11/45 METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]            { Reset the UIC }
MCR> PIP /FU                     { Check file ownerships }
MCR> PIP [270,5]/NV=xxx          { For each file xxx not
                                   owned by [270,5] }

MCR> HELLO [1,1]
MCR> PIP [270,5]xxx/PU           { for all files xxx with
                                   ownership to be changed }

MCR> SET /UIC=[270,5]
MCR> RUN [270,2]DMPCON<esc>      { Do not edit the control file }
***** Program Completed ****
^C
MCR> RUN [270,2]MAKCON<esc>
***** Program Completed *****
^C
MCR> @[270,1]FULLUPD              { Answer YES to the restart
                                    question }
```

4. The error message "FCSV.P00 LOCKED FROM READ/WRITE ACCESS" from PIP indicates the FCSV file was accidentally deleted. Verify that it is gone and recreate it. To do this enter the following from the PDP-11/45 METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> PIP FCSV.P00/LI              { Look for it }
MCR> RUN [270,1]ALLOC<esc>        { To recreate it. This will
                                    prompt for a file name and
                                    block count -- 1200 for
                                    FCSV }

MCR> PIP FCSV.P00;/PR/SY:RWE/OW:RWE/GR:RWE
MCR> RUN [270,2]DMPCON<esc>       { Do not edit the control file }
***** Program Completed ****
^C
MCR> RUN [270,2]MAKCON<esc>
***** Program Completed *****
^C
MCR> @[270,1]FULLUPD               { Answer YES to the restart
                                     question }
```

If you are ever in doubt about FCSV.P00, Delete and recreate as above. To delete enter the following from the PDP-11/45 METER console:

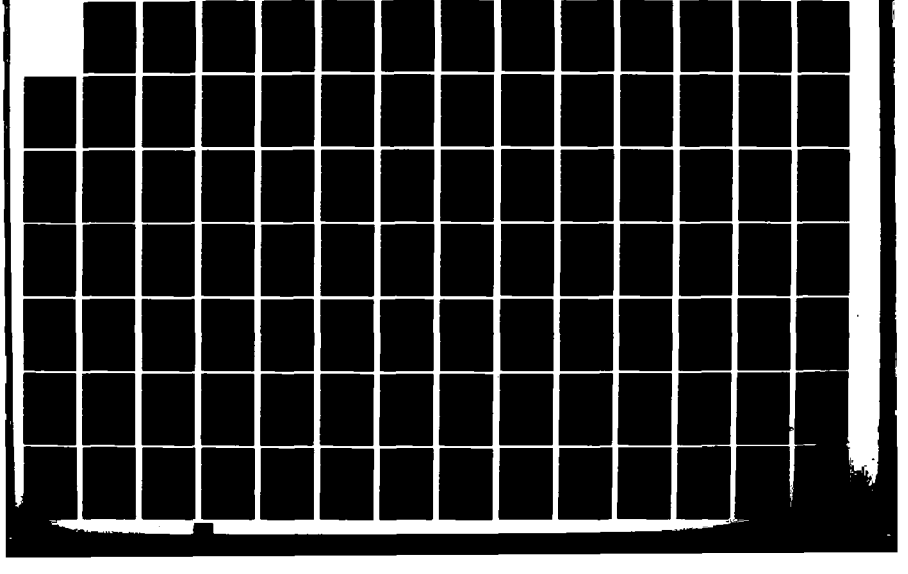
```
MCR> HELLO [1,1]
MCR> PIP [270,5]FCSV.P00/PR:0
MCR> PIP [270,5]FCSV.P00/DE
```

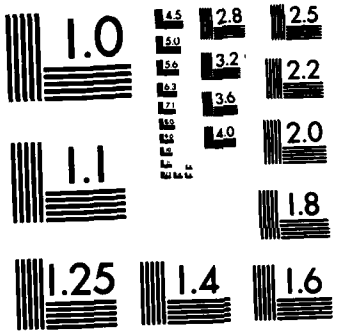
ll else fails, begin again from your MESSAGES.DAT input file and run re-processor and update command files over. You should first reset control file to what it was before you ran the pre-processor the last . To do this enter the following from the PDP-11/45 METER console:

```
MCR> HELLO [1,1]
MCR> SET /UIC=[270,5]
MCR> PIP CONTROL.TXT/LI          { To find the version of
                                   control text in effect
                                   at the start of the last
                                   run of the preprocessor.
                                   Check the times associated
                                   with the files with the time
                                   of the last invocation of
                                   the preprocessor. }

MCR> PIP CONTROL.TXT;/DE         { To get rid of all versions
                                   comming after the one you
                                   want to restart from.  You
                                   will have to do this once
                                   for each version to be
                                   deleted.

MCR> RUN [270,2]MAKCON<esc>      { NOT PRECEDED BY MAKCON }
***** Program Completed *****
^C
MCR> PIP [270,5]*.P00;*/DE       { Delete left files from the
                                   failed update. }

MCR> { Recover the MESSAGES.DAT file from wherever you stashed it.
       If you can't find it, forget about this update. }

MCR> @[270,1]PREPRCC
MCR> @[270,1]FULLUPD             { This is not a restart.
                                   Answer NO the restart
                                         question. }
```

AD-A125 166 OPERATIONAL TEST AND EVALUATION OF THE METER ENGINEERING DEVELOPMENT MODEL(U) PAR TECHNOLOGY CORP NEW HARTFORD NY R J D'AMORE ET AL. NOV 82 3/4

UNCLASSIFIED RADC-TR-82-304 F30602-80-C-0232 F/G 14/2 NL





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

6. Miscellaneous

- Avoid running UPDROP and SHORTUPD.

- After an update, run the clustering with the minimal spanning tree algorithm and the index stems report for summaries. It is not necessary to run these at all if you do not wish to.

APPENDIX C

Meter System Managers Guide

PAR Technology Corporation

4 August 1982

METER System Manager's Guide

Prepared under RADC Contract F30602-80-C-0232.

This Guide describes how to run the METER system for best results under the changing conditions of an operational military intelligence center. It applies to the Advanced Engineering Model version of METER delivered for operational test and evaluation at Military Airlift Command HQ I&W Center, Scott AFB, IL.

1. PREFACE

This section presents the assumptions here about the intended reader of this Guide and gives an overview of it. A list of related METER documentation is provided for further reference.

1.1 Assumptions about the Reader

This document is intended for a data processing manager responsible for the overall operation of METER in an intelligence center as a means of deriving timely information for analysts from accumulated message traffic. The METER System Manager need not be an expert on running the AN/GYQ-21(V) (PDP-11) computing system under RSX-11D/IAS; this function will typically be delegated to one or more specially trained METER System Operators. Some familiarity with information retrieval techniques would be helpful, but not absolutely necessary.

1.2 The Role of the METER System Manager

The METER System Manager will establish policy with respect to METER user access, resource allocation, and information products in order to permit intelligence analysts to exploit online collections of message text to the maximum degree. This document describes these problems in detail and provides the necessary background for effective decision-making about METER operations.

1.3 Related Documentation

The following documents describe the different aspects of the METER system in detail:

- The METER User's Guide is a reference for the intelligence analyst using METER, describing the implementation of METER as seen by the METER user at an OJ-389 analyst workstation. It is written for the nonexpert user

and assumes no prior experience with computers.

- The METER Operator's Procedures Manual explains how to set up and run the components of the METER system. It consists of step-by-step procedures to be carried out at the METER Operator's terminal.

- The METER Core System Description is aimed at the METER systems programmer. It describes the various file structures defined by METER and the different subsystems, utilities, and libraries constituting METER.

- The MAXI-METER Implementation Description documents the interface between METER and MAXI as implemented in the Advanced Engineering Model. It gives a general overview of MAXI from the perspective of METER operation.

- The METER System Configuration Management Manual describes software configuration control for the Advanced Engineering Model System.

- The METER Test Plan outlines the approach to be taken by PAR in testing and evaluating METER at MAC.

The METER System Manager will be responsible for retaining and making available copies of METER documents as required. Any updates of documentation will be directed to METER System Managers.

1.4 The Organization of This Guide

The METER System Manager's Guide has eight main sections:

1.4.1 Preface: an introduction to the System Manager's Guide.

1.1 Assumptions about the Reader.
1.2 The Role of the METER System Manager.
1.3 Related Documentation.
1.4 The Organization of This Guide.

1.4.2 Overview: a summary of METER operation and methods for controlling it.

2.1 Operational Summary.
2.2 Control Options.
2.2.1 Scheduling.
2.2.2 Data Base Size.
2.2.3 Stopwords.
2.2.4 Indexing.
2.2.5 Associations.

1.4.3 Supporting METER Users: getting them on the METER system and helping them out.

3.1 Establishing a New User.
3.2 Training Users.
3.3 User File Requirements.
3.4 Security
3.5 Help for the User.

1.4.4 Resource Management: space and time computational requirements of METER.

4.1 Hardware Requirements.
4.2 File Management.
4.2.1 Naming Conventions.
4.2.2 File Offloading.
4.2.3 Update Purging.
4.2.4 Required Contiguous Allocations.
4.2.4 File Directories.

1.4.5 Subsystem Scheduling: when to run updates for best results.

5.1 Processing Time Estimation.
5.2 Scheduling Criteria.
5.2.1 Message Acceptor.
5.2.2 Message Analyzer.
5.2.3 Message Summarizer.

1.4.6 Operational Parameters: how to adjust METER to meet operational requirements.

6.1 Stopwords.
6.2 Data Base Window.
6.3 Index Stem Limits.
6.4 Association Threshold.
6.5 Query Weighting.
6.6 Base Year and Span of Coverage.
6.7 Message Sources.

1.4.7 Content Summaries: how to produce reports for users.

7.1 Stem Associations.
7.2 Stem Clusters.
7.3 Stem Trends.

1.4.8 Performance Analysis: tools for assessing effectiveness of a particular METER configuration.

1.4.8.1 Instrumentation.
1.4.8.2 Data Base Summaries.
1.4.8.3 Operator Error Logging.

An Appendix to these sections explains how to change system parameters in the METER data base control file. This duplicates material in the Operator's Procedures Manual. Included also is a glossary of standard METER terminology taken from the METER Core System Description.

2. OVERVIEW

This section provides an overview of how METER works and how its operation can be be controlled. Section 6 of this Guide will go into detail on procedures for actually adjusting METER operating characteristics.

2.1 Operational Summary

METER is an information system based on dynamic collections of message text. It is designed to process a thousand incoming messages per day with an update time of under two hours on a minicomputer system with a large disk drive; it will provide users with the capability of entering natural language queries against collected messages.

The version of METER at the MAC HQ I&W Center is an Advanced Engineering Model with special interfacing to MAXI software running on the AN/GYQ-21(V) Analyst Support Processor at MAC/IN and accessible throughout MAC/IN from OJ-389 analyst workstations. METER has been installed at MAC for operational test and evaluation in its present configuration; MAC is the first operational site to receive METER.

Currently, METER is organized into four separate subsystems:

The Acceptor Subsystem (Q)

takes messages from the MAXI Message Support Subsystem and prepares them for subsequent analysis in a METER data base update. Parts of the message acceptor will typically be scheduled by RSX/IAS to execute periodically. The METER input message stream from MAXI can be turned on or off by a logical switch built into the OSIPRE preprocessor task for MAXI MSS.

The Analyzer Subsystem (P)

computes statistics for a collection of incoming messages, indexes them for retrieval, and then updates a METER data base. The output of the Analyzer includes an association matrix relating the co-occurrence of index word stems in messages; this will be used like a thesaurus at retrieval time to expand a user's query.

The Extractor Subsystem (R)

retrieves messages according to their estimated relevance to a user's query. It includes an automatic query expansion facility to improve the likelihood of a user retrieving relevant messages. The MAXI interface to this subsystem is described by the METER User's Guide. The R subsystem will be invoked by VFK's from an an analyst workstation and will accept natural language text for a queries.

The Summarizer Subsystem (S)

produces optional statistical summaries of a data base, showing lists of important word stems ordered by various criteria, stem associations, stem clusters for a data base, significant changes in index stems since the previous update, and significant trends in index stems. This information will be helpful for users in getting acquainted with a data base and for the System Manager in assessing the performance of METER.

The Acceptor and Analysis subsystems together constitute the METER update process.

2.2 Control Options

When running METER, there are five basic questions from the perspective of system management:

- When should the the various subsystems run?
- How many messages should be retained for retrievals?
- What words should be processed?
- How much indexing of messages should there be?
- To what degree should computed associations be used?

2.2.1 Scheduling.

The Acceptor and Analyzer subsystems typically will be scheduled separately. The Acceptor should run fairly frequently because messages will be coming all the time and should be processed as soon as possible if the computational resources are available. The Analyzer should run at least once per day and more often if timely information is critical. The Analyzer, however, will require at least one or two hours to complete a full update under typical conditions at MAC HQ and will load down a processor significantly.

There is an alternate option of a short update, which incorporates the latest batch of messages, but uses previously derived index stems and associations. The short update is useful when a small number of messages have to be brought quickly into a data base; it takes only a few minutes to run because it does not recompute statistics. Short updates can be run repeatedly, but when these account for a non-negligible portion of a target data base, a full update must be run to revise data base statistics to reflect changing content.

The Extractor and Summarizer subsystems will always run on demand. The Extractor is invoked whenever a user retrieval is initiated and can run at any time except for a minute or so at the completion of a short or full update

·hen new data base tables are switched in. The summarizer normally would run once immediately after a full update.

2.2.2 Data Base Size.

The size of a METER data base is measured in the total number of messages stored for retrieval and the number of (full) update cycles that these represent. In general, retrievals by users will become slower with increasing data base size. METER is currently compiled to handle up to 100,000 messages; a System Manager must make sure that this limit is not exceeded by controlling the number of (full) update batches to be accommodated by METER. When this number is reached, METER will automatically purge the messages in the least recent update to make room for new messages.

2.2.3 Stopwords.

METER can be d·:ected to ignore specified words when encountered in input text. The words to be ignored are kept in the METER stopword table; these would typically be grammatical function words like "the" or special message marker tokens. A good stopword table is important when running METER on a minicomputer because it reduces the volume of data to be processed, saving both time and space, and it greatly improves message indexing by filtering out words with little information content. Every table must, however, be tailored to a particular application through trial and error.

2.2.4 Indexing.

The content of a message is indicated in METER by the index stems that it contains. A larger number of index stems means it is more likely that the content of a message can be represented adequately, but this results in longer

processing time in updates and in retrievals. METER allows for an absolute limit of 3500 index stems, but a lower number of index stems can be selected to speed up METER operation.

2.2.5 Associations.

Measures of association between pairs of index stems are an important statistic in METER. They are used in expanding queries to make retrievals more comprehensive and in constructing data base summaries, but they are costly to compute, accounting for one-third to one-half of processing time in the Analyzer subsystem. METER allows the specification of criteria for choosing stems to compute associations for; such criteria will be critical for speed of update because association computation time is proportional to the square of the number of stems to be associated.

3. SUPPORTING METER USERS

The METER System Manager will determine who is to have access to METER and will be responsible for training METER users and providing any necessary systems support. This section will describe the establishment of new METER users, user training, user file requirements under METER, security, and help for users.

3.1 Establishing a New User

Any MAXI user may run METER, but first the user's MAXI 9-digit analyst identification number must be entered in a METER users list for the assignment of retrieval file directories. This list is created by editing a file called USERS.TXT in the METER directory [270,1] on the Analyst Support Processor system pack and then running the program MAKEUSERS; the list may be dumped for examination by running the program DUMPUSERS. See the Operator's Procedures Manual for more details on this.

3.2 Training Users

The Advanced Engineering Model of METER will be a subsystem of MAXI, available as an option in the top-level MAXI escape menu. An analyst must be able to log into MAXI before running METER. Analyst training should therefore cover the following areas:

- Basic MAXI concepts: subject areas and profiling, user queues, message review, and work files.

- Analyst workstations: forms and menus, VFK's, editing text on the screen, paging, and special terminal functions like hardcopy output and escaping.

- Basic METER concepts: index stems, associations, estimated relevance, ranked retrievals, restrictions on retrievals, and culling.

- METER menus and forms implemented under MAXI.

- METER offline aids.

Users at MAC should be aware that METER is undergoing operational test and evaluation and is subject to change. METER usage will be monitored in detail, and some users will be interviewed as part of the evaluation.

The METER User's Guide describes the menus and VFK's seen by the user. METER is fully integrated as a MAXI subsystem, incorporating the MAXI conventions for terminal communication and allowing tasks to be invoked by typing their names at the keyboard, using a lightpen on a right-screen menu, or hitting a VFK on the left pad of a terminal.

3.3 User File Requirements

Each METER user must have an RSX-11/IAS file directory on the METER host system for temporary files created in query generation and the retrieval process and for saving METER usage information as part of its instrumented operation. The setting up of such directories is described in the METER Operator's Procedures Manual.

At MAC HQ, METER user directories will be of the form [277,xxx], where xxx is a METER user identification number in the range 1 to 377 (octal). The METER identification number will correspond to the position of an analyst's 9-digit MAXI identification number on the METER users list.

METER files will be invisible to users, but the System Manager will be responsible for insuring that they are periodically purged to avoid overflowing disk storage. See the METER Operator's Procedures Manual for more details on this.

3.4 Security

The METER System makes no provision for the security of sensitive information in messages; this will be the responsibility of MAXI. METER is assumed to run on a processor in SYSTEM HIGH mode with controlled access to terminals. A METER user may choose to view messages only of a certain classification by employing the METER RESTRICT facility, but this is entirely at the option of the user.

Files produced by a user in a METER directory will be unprotected and may be read or deleted without restriction. The user may clean up such files with the RESTRICT facility, but they cannot be saved in any way within METER. Most user files in any event will become obsolete after completion of a full data base update. Text displayed by METER may be copied to MAXI work files for long-term storage, however.

3.5 Help for the User

Online help for the use of METER commands will be available as part of the standard MAXI EXPLAIN facility. Other information will be incorporated directly in METER forms as appropriate. Information about a data base is available in the form of content summaries produced by the METER Summarizer subsystem; these should be made available to users requesting them.

The Summarizer will provide the following reports:

Indexing: Program SINDX will produce an alphabetic listing of index stems with associated content statistics and a second listing showing the top 100 index stems ranked by four different statistics. This shows what words METER will recognize in queries and will generally be helpful in identifying words that should go on a stopword list. The statistically ranked listing will usually be more helpful for users.

Associations: Program SASSN will produce a listing of entries in the current association matrix that exceed a specified threshold value. These are helpful in becoming familiar with a message collection and in understanding query expansion. See Section 7 more details.

Clusters: The METER programs SPANS, MUTUAL, and GAPS implement different clustering algorithms for identifying closely related groups of word stems in a collection of messages. The programs SCLSTR and PRNSCLSTR will produce a listing of stem clusters sorted in descending order of size. SPANS applies a minimal spanning tree hierarchical clustering algorithm to stem associations; MUTUAL clusters mutual nearest neighbors; and GAPS clusters according to statistically significant gaps in association values. These programs tend to produce similar results, but perform best under different circumstances. Clusters will be a good way to get an overview of a message collection. See Section 7 for more details.

Trends: The program STRNDS produces an analysis of stem change trends in the data as a function of time, using normalized statistics for messages entering a collection over the 7 most recent updates. This will show stems with significant increasing or decreasing frequency as well as stems with anomalous statistics in the most recent update relative to immediately preceding updates. A

second program SCHNGS produces a listing of changes in index stems in the most recent update. Both programs provide indicators of things that an analyst might want to look at.

Summarizer reports can be run once after each full update, but may be omitted at the METER system manager's option. They will take about 5 to 10 minutes to execute, which is negligible compared to times required in message analysis. Analysts should review these reports when coming on watch in order to get an idea of what is going on. These reports are also helpful for system performance assessment (see Section 8).

4. RESOURCE MANAGEMENT

This section deals with METER hardware processor and memory requirements and file management on secondary storage.

4.1 Hardware Requirements

The minimum hardware required is a DEC PDP-11/45 with a floating-point processor option and 124K bytes of memory; at least 64K bytes more memory will be needed to avoid degraded performance when retrievals by two or more users have to be supported concurrently with full updates. The system should have at least a single large disk drive with at least 88M bytes of storage at a 1M byte/second data transfer rate plus two magnetic tape drives for backing up the system and offloading files not required for immediate use.

Extra processors, main memory, and disk drives will speed up METER significantly. The various METER subsystems can be split up to run on different machines; for example, all message extractor programs including the user interface can run on a separate dedicated processor if many users are to be supported. Running METER on a PDP-11/70 instead of an -11/45 should make processing twice as fast because of the additional mass bus on the -11/70 for supporting direct memory access by I/O devices.

The most critical space requirement is for the METER data base files on disk: message text, indexes, statistics, and intermediate computations. These will amount to about 8 blocks per message or about 80,000 blocks for a data base of 10,000 messages. Another 20,000 blocks of disk storage will be needed for temporary files when running a full update. The number of index words will also affect this total. The amount of text normally kept by METER should be about 2,000 bytes per message, but may range from 1,000 to 4,000 bytes. Note that METER must currently store message text already stored by MAXI, although there is some compression of text in METER by eliminating a

large part of message headers.

Dynamic memory requirements will depend on the degree of concurrency in running METER subsystems.

- The Acceptor consists of two main tasks running in a sequence, with a maximum memory requirement of 52K bytes, plus two background data acquisition tasks of about 24K bytes each. The data acquisition tasks will run on two separate processors, since incoming messages typically will not be received on the processor on which the main METER system is resident.

- The Analyzer is a sequence of sixteen tasks (see the Operator's Procedures Manual), requiring a maximum of 54K bytes of memory for any one task; most should take about 30K bytes.

- The Extractor comprises ten independent tasks plus three monitor tasks for MAXI ICC communication. These are all set up as single-user tasks so that only one invocation of each can be running at any one time. They require about 40K bytes of main memory each, with a maximum of 50K bytes. Total resource requirements will depend on the degree of concurrency to be supported. METER Extractor modules have been designed according to MAXI conventions to be non-interactive so that none will ever occupy main memory while waiting for a user response; they simply read or write a display and and exit immediately.

On a PDP-11/45, main memory will be a severe restriction. Of the maximum of 124K words available, about a half will be taken up by the executive of the RSX-11/IAS operating system and by various device handlers, including the DA pseudo-device for the ICC interprocessor link. The MCR indirect command processor used to sequence METER programs during updates will take another 4K words when running. This means that only two METER processes can be resident

in main memory at any time. When running an update process, only one user can be served by the PDP-11/45 and may be locked out for extended periods because of fragmentation of free memory.

4.2 File Management

METER requires fast secondary storage for saving message text, index files, data base statistics, system programs and user files, and intermediate computations. The data transfer speed of such secondary storage will be the major hardware factor in overall METER performance; i.e. METER is I/O-bound.

All METER files are in RSX/IAS FILES-11 format. To save space and speed up processing time, most files are organized as sequential fixed-length records requiring special procedures for access from FORTRAN. In some cases, files have to be allocated contiguously on a device to make it possible to meet time-critical processing needs (see below).

To avoid overflowing available secondary storage, METER is set up to delete most working files automatically when they are no longer needed. Operator intervention is required as a precaution, though, for deleting certain files; periodic scanning of secondary storage is needed to avoid excessive accumulation of these files.

METER user directories, designated [277,***] should also be periodically purged. Because internal message numbering and index words change with each update, accumulated non-text files (i.e. without a ".TXT" suffix) would be obsolete anyway.

4.2.1 Naming Conventions.

On RSX/IAS systems, files produced by the METER Message Acceptor, Analyzer, and Summarizer have special suffixes identifying their source: ".Q**" for the Acceptor, ".P**" for the Analyzer, and ".S**" for the Summarizer. All these files will reside in a particular data base directory. The ".P**" files also have special names, starting with "F" if they contain information pertaining only to messages from a single update or starting with "G" if they contain information over an entire data base of messages. Naming conventions are fully described in the METER Core System Description.

The ".P**" suffixes have special significance. The ".P01" files will be those generated as of the most recent full update; ".P02", ".P03", ".P04", and so forth will correspond to files from less and less recent updates. The suffix ".P00" is kept for the files of the next update before it is complete. The suffix ".P0N" is for all files reloaded from offline storage so that they can be used in purging old messages from a system. The suffix ".P0S" is for special short update files.

4.2.2 File Offloading.

Many METER files are saved only for use in purging messages from a given update. If messages stay in the system for a long time such as more than a month, you may save secondary storage by putting files onto magnetic tape until they are actually needed. The following files can be moved offline in this way

```
FDIP.P02, FDIP.P03, ...
FDIPX.P02, FDIPX.P03, ...
```

The files *.P0k are needed only when the next update must purge the messages for day k; they should then be reloaded from tape under the file type *.P0N.

A full description of file lifetimes and permissible offloading is given in the METER Operator's Procedures Manual and also the METER Core System Description.

A complete data base may be saved on tape for future reference by copying out the directory containing the data base files. It can be reloaded in a different directory and accessed by editing a user's DATABASE.UIC file to specify the new directory.

4.2.3 Update Purging.

METER has no provision for indefinite archival; this function must be served by MAXI. Accordingly, every message will eventually be purged by METER. This may be postponed by increasing window sizes and by update consolidation, but there will be a point when METER can accept no more new messages. The current limit on data base size is 100,000 messages, which would more than saturate available secondary storage on a typical AN/GYQ-21(V) configuration.

For users who require message storage of indefinite duration, it is possible to set up separate METER data bases for them with their own update schedules. Access to other data bases is possible by changing the DATABASE.UIC file in user's directory to indicate the new target data base. Strict file storage management will be essential for maintaining multiple data bases, however. Only a single data bases is currently planned in the initial MAXI/METER configuration.

4.2.4 Required Contiguous Allocations.

The METER message analyzer and message extractor subsystems are limited in speed by I/O. Because most of this involves sequential access to disk files, it is possible to achieve major improvements in processing time by allocating such files contiguously to cut down on disk head movements. Even faster operation will be possible with multiple disk controller and disk drive configurations; this would allow METER to allocate files so that those simultaneously accessed by any given program can be put on separate devices.

Even with only one disk drive, however, contiguous files are advantageous because they allow for fast multi-block disk operations, implemented on RSX/IAS systems to support task image loading. This allows for speedup even though a program might have to access more than one file at once. Multi-block operations can read or write several disk blocks with only a single disk head movement, thus saving significant time where processing is I/O-bound.

The following files should be contiguous to meet minimum METER performance specifications:

FCSV.P00	required in update, about 200 blocks for every 1000 index words
GCMV.P01	required in retrievals, about half a block for every message in a data base

These files should be pre-allocated on a disk and protected from deletion before starting up METER to avoid problems of storage fragmentation cutting down on available contiguous disk space. The files could be n-block piece-wise contiguous also, but this would requiring recompilation of referencing programs to make sure that they are accessing n blocks at a time. See the METER Core System Description for more details.

4.2.5 File Directories.

In the MAC/IN implementation of METER, the number "270" is the METER group number for non-user RXS UIC's. METER file directories are set up as described below. The letters after the UIC's refer to the scope of each directory. The codes are:

SM	System Manager
SO	System Operator
SP	System Programmer
USER	Analyst Users

[270,1] SM SO SP
The METER users list, command files, and maintenance software.

[270,2] SM SO SP
Various utilities and included files required for compiling METER programs.

[270,3] SP
For program development and maintenance. This contains libraries for linking METER programs.

[270,4] SP
For program development and maintenance. This contains libraries for linking METER programs.

[270,5] SM, SO, USERS
Contains all current files for the principal METER data base.

[270,6] SM, SP

For alternate data bases.

[270,7] SM, SP

For alternate data bases.

[270,10] SO, SP

Contains all source files for the METER message acceptor.

[270,12] SO, SP

Contains all source files for the METER message analyzer.

[270,14] SO, SP

Contains all source files for the METER message extractor.

[270,16] SM, SO, SP

Contains all source files and executable programs for the METER message summarizer.

[270,20] SP

For program development and maintenance. This will contain non-core libraries for linking METER programs.

[270,270] SO

This will contain all METER task images for current operation.

5. SUBSYSTEM SCHEDULING

This section deals with the question of when to run METER programs. Because full updates are computationally costly, they must be carefully scheduled to provide users with timely information when it is needed.

5.1 Processing Time Estimation

To make decisions about scheduling, the System Manager must be aware of how much time various METER processes will take to complete and how this can be modified by altering various system parameters.

Retrieval time will be directly proportional to the number of messages searched. With 3,000 index stems, this will amount to about 1 second per 1,000 messages in a typical data base, assuming an average of about sixty index stems found in each message. Retrieval time will be almost entirely a function of how fast a disk can transfer data to METER retrieval programs; it will deteriorate when other processes, including other METER users, are contending for access to the disk. When retrieval times are too slow, then the number of messages to be searched must be reduced, or the number of simultaneous users must be limited.

For data base updates, the problem of estimating time is more complicated. The following describes a first-order estimation procedure for a full METER update under various conditions. The actual time values here are based on a PDP-11/45 processor with RP04 (or equivalent) disks for storage; the times assume no other processes running and thus might be much longer on on a heavily loaded processor. Operations on a DEC PDP-11/70 with a mass memory bus should be about twice as fast as on a PDP-11/45. Because METER is I/O-bound during update processing, the actual speed of the host processor CPU will make only a relatively minor contribution to overall update times.

Details on the functions of individual programs and variables mentioned below can be found in the Core System Description. Total execution time for update will be defined as follows:

```
TOTAL TIME =   STEM INVERSION TIME
             + STATISTICS COLLECTION TIME
             + INDEX STEM SELECTION TIME
             + ASSOCIATION COMPUTATION
             + MESSAGE INDEXING
```

This includes only the processing in METER during message analysis. Processing in the METER message acceptor will require additional time of about 3.0 seconds per message on a PDP-11/45, but usually will run as a background task.

Let

```
K = number of messages in this update,
M = total number of messages in data base after update,
N = total number of index stems to be selected,
```

With all figures in thousands, then the time in hours for

```
Stem Inversion       = C1*K*(LOG2 (K) + 10)
Stem Statistics      = C2*K
Index Stem Selection = C3*K*(LOG2 (N) + 10) + C3S
Stem Association     = C4*K*N*N
Message Indexing     = C5*M*(LOG2 (N) + 10),
```

Where the coefficient values, based upon a PDP-11/45 implementation of METER as of April, 1982, at MAC HQ, are as follows

```
C1 = .02
C2 = .04
C3 = .008, C3S = .05
C4 = .12
C5 = .003
```

For a typical MAC configuration, where K = .75, M = 15, and N = 3.5, with all figures in thousands, total message analysis time is about 1.9 hours. A special METER program called ESTIME has been provided to compute estimated from given values of K, M, and N obtained by prompting the user; see the Operator's Procedures Manual for more details.

The short update procedure will usually take under ten minutes. Use this when recomputation of statistics is not critical.

5.2 Scheduling Criteria

Effective scheduling of METER operation depends on a number of factors: the volume of message traffic, the criticality of timely information, the work schedule of users, and the amount of processing required by other systems. Eac. METER subsystem has to be scheduled differently.

5.2.1 Message Acceptor

In the MAC implementation of METER, messages will be coming to METER from the MAXI MSS subsystem on a continuous basis. The current scheme is to tap incoming messages before they are matched against profiles in MSS. There will be a software switch in MAXI to turn the message to METER on or off; if the switch is on, then all messages sent by CSP to MAXI will also go to METER.

Messages will be transferred to METER as individual RSX-11/IAS files in a designated directory on the system disk of the MAXI host processor. The METER QGATES program running on that processor will periodically read these files and will ship the messages via ICC to the METER QGATER program running on the METER host processor. QGATER will create a packed text file of messages for the Acceptor subsystem to process.

The design of the Acceptor subsystem makes it more efficient to process messages in batches instead of singly. At the same time, it is advantageous to run the Acceptor whenever there is free time so as to get a leg up on a data base update. The best strategy to take here will depend on the volume of message traffic.

If traffic is heavy (e.g. more than a hundred messages per hour), then the Acceptor should be scheduled automatically to run about once every hour. More frequent scheduling is possible during a crisis, but this would start putting a heavy load on a processor. If traffic is light, it may be possible to run the Acceptor once per shift or even once right before each full update of a data base.

Note that QGATES will have to be scheduled separately from the main Acceptor subsystem since it is on a different processor. In this case, QGATES scheduling will depend on what is happening on the MAXI host processor. QGATES should run as often as possible without overloading the processor, but it will be more efficient if it can process at least 5 to 10 messages at a time.

5.2.2 Message Analyzer

A full update consists of bringing a new set of messages into a data base, optionally dropping off an old set of messages, and recomputing statistics. Ideally, updates should be continuous, but since this involves a great deal of startup overhead even for a few new messages, it is more efficient to accumulate messages in batches. The primary restriction on METER update cycles is that one cannot begin until the previous one is done. Because full updates will take at least an hour and possibly over two hours for large data bases with extensive indexing, there is a practical limit on the number of updates per day; more than four per day will probably overload a small machine to the disadvantage of system users.

In addition to a full update, there is a short update option that may be run when the number of new messages entering is much smaller than the size of the current data base. This allows the new messages to be entered without going through the recomputation of statistics or the reindexing of old messages, thus eliminating about ninety percent of the time required for a full update. A short update, however, is only a temporary measure to make messages available for retrieval as soon as possible. Eventually, all messages entered through short updates must be reentered through the full update process so as to bring data base statistics and indexing back in line with the actual contents of a data base.

A full update should be run at least every 24 hours in routine situations, and at least every 12 hours when the volume of incoming messages reaches several thousand per day. Short updates can be run once every 2 hours, or more frequently in crises.

When a data base contains fewer than ten thousand messages, it may generally be more efficient to run full updates as often as possible instead of short updates. A full update, however, will be advantageous only when the number of incoming messages in the latest unprocessed batch is more than several hundred.

5.2.3 Message Summarizer

Summarizer programs need to be run only once after each full update, since METER statistics will remain unchanged until the next full update. The choice of which programs to run will depend on what users want. It will usually be advantageous to produce reports on index stems and on stem clusters for offline use just before a change of watch; these should each take under five minutes to do. Other reports are more specialized in nature and require more time to produce.

6. OPERATIONAL PARAMETERS

The principal parameters controlling METER operation are maintained for each METER data base in the METER control file (CONTROL.DAT) and various tables. The contents of the control file may be inspected by running program DMPCON in the METER utilities directory [270,2] to produce a printable file called CONTROL.TXT showing all current values in the control file along with descriptive labels. These values may be changed by editing the CONTROL.TXT file and then running the utilities program MAKCON to translate any modifications back into the control file. Similar procedures apply to table modification; for more details on control file and table manipulation, see the Operator's Procedures Manual. The Appendix contains a short description of control file modification.

Scheduling of updates is is not a control file function, although the file does keep track of when the most recent update occurred. Ordinarily, updates should be initiated by the METER System Operator according to current information needs and the volume of message traffic. The METER System Manager has ultimate responsibility for deciding when any update is to occur.

6.1 Stopwords

The METER stopword table provided in the distribution system consists mostly of English grammatical words not likely to be of use for indicating the content of messages. This table will probably have to be adjusted for a particular target data base. For example, a common grammatical word like "BEGIN" may have to be dropped from a stopword list because of the Israeli Prime Minister's name, or a nonsense string WXYZ may have to be added because of its frequent use in delimiting sections of certain classes of messages. The METER stopword subroutine can be set up to recognize item list tokens of the form XXX001, XXX002, ..., by giving it a pattern in the form XXX###.

Changes to the stopword table can be made by editing the file STOPWORD.TXT in directory [270,2] to add or drop words and then running the METER program STOPBUILD. The output of STOPBUILD is a file STOPWORD.TBL, which should be copied into a data base directory. The new table will apply only to new incoming messages. See the Operator's Procedures Manual for more details. It should be noted that METER automatically drops words of fewer than three characters in length; this rule can be changed only through recompilation of programs.

The addition of stopwords will increase lookup time in the Message Acceptor, but it will save even more time in the Message Analyzer when high frequency words with low content can be eliminated. On the whole, it pays to have as many stopwords as possible up to the limit for the table. Indexing can be improved with proper stopword list tuning; a good practice is to periodically review index stems selected by METER to identify new stopword candidates.

6.2 Data Base Window

The maximum window size parameter in the control file specifies the number of updates that may occur before automatic purging of the messages from the oldest update; the current window size shows how much of the window is taken up by updates.

If updates are frequent, then the maximum window size should be extended. A large window size, however, will result in slower METER operation for keyword restriction of retrievals and for displaying retrieved messages and will make message indexing take longer.

The best approach is probably to set update scheduling so that messages of interest can be accommodated within a fixed window size. If message traffic fluctuates greatly, however, it may be necessary to reset the window

size periodically to maintain a minimum number of messages in a data base or to avoid overflowing disk space.

6.3 Index Stem Limits

METER indexes messages with word stems selected by a statistical content measure of how important a word is in distinguishing between messages in a data base. The maximum number of stems for indexing is 3500, but for faster updates, this number can be reduced in the control file for a data base. METER is initially set so that no stem will ever be selected for indexing if it shows up in fewer than 5 messages or in more than a third of all messages in a data base. These limits are defined in the "Lower index message count threshold" and "Upper index message percentage threshold" (over an entire message collection) control parameters.

Reducing the number of index stems will degrade the precision of METER retrievals. The optimum number of index stems will depend on the diversity of subject matter in a message collection, but the best policy is to have as many as possible without making update times excessively long. A good rule of thumb is to have about 50 to 60 index stems on the average in a message. If there are many messages without index stems despite having a large number of index stems, it might be helpful to extend the stopword list for a database instead of adding more index stems.

6.4 Association Threshold

METER will compute association statistics only for index stems. Furthermore, it rules out index stems that occur in too few or too many messages; the thresholds are set by the control parameters "Minimum message count for association" and "Maximum message percentage for association" (over an entire message collection).

Associations are retained only if they exceed a specified threshold. Daily contributions must meet the "Daily co-occurrence threshold"; associations themselves, the "Minimum association threshold." These are also defined control file parameters, but probably should normally be left at their defined values.

The "Minimum message count for association" should be at least 10; otherwise the sample size for co-occurrence statistics is to small to make any inferences from. The "Maximum message percentage for association" should be under 50; as this number increases update time goes up as the square without really giving much more useful information.

Note that only index stems can be associated. In general, the criteria for stem to be associated should be much more stringent than for stem to be selected for indexing. Restrictions on associations will significantly speed up update processing.

6.5 Query Weighting

- The "Association weight" parameter determines how much statistical associations are to contribute to a user's expanded query. This is expressed as a fraction between 0.0 and 1.0; a value of 0.0 means to disregard associations entirely, while 1.0 means to give associated words equal weight with the original words of a user's query. Experimentation has shown that a value of about 0.5 seems to be best; as the "Association weight" increases, the potential coverage of a query increases, but this is offset by additional noise. If it appears that certain important messages are not being retrieved, the weight can be incremented a little.

- The "Context weight" parameter determines how much non-index words should contextually contribute to an expanded query. This is currently unimplemented in the MAXI version of METER.

6.6 Base Year and Span of Coverage

Because of space limitations, METER can encode dates only within a range of four years starting from a specific base year. The "Base year" control parameter can be adjusted to cover different ranges, but for all intents, this value may be regarded as a constant at MAC. It is unlikely that METER will run for four years without major changes in the system, and in any event, the expected volume of message traffic will preclude retaining more than several weeks of messages at a time.

6.7 Message Sources

METER maintains a table of up to 127 message sources that may be specified by a user in restricting messages to be retrieved; this is stored as a file SOURCE.TBL associated with each METER data base. The table is created using the special METER program MAKSOURCE, which takes its input from a file SOURCE.TXT created with a text editor; the table may be dumped for inspection with the program DMPSOURCE.

A source name lookup is carried out for every incoming message. Since the overhead for this is reflected directly in each data base update, it is advantageous to keep the size of the table to under 32 sources to reduce search times. Patterns in the source name table will be 8 characters long; every 64 names therefore implies another disk block access, although a binary search algorithm helps out somewhat.

A source name must match patterns exactly up to the length of the pattern. A pattern of fewer than 8 characters can be specified by terminating it with the special character '*'. For example, the pattern 'ABCDEFGH' will match the source names 'ABCDEFGH' and 'ABCDEFGHIJKL', but not 'ABCDEFG'; the pattern 'ABCD* ' will match 'ABCDE' and 'ABCDEFGHIJKL', but not 'ABC'.

7. CONTENT SUMMARIES

The reports produced by the METER Message Summarizer Subsystem require no operator intervention except in the case of stem association, stem cluster, and stem trend programs. These ask the operator to supply various parameters to establish the amount of detail wanted; in general, more detail implies a longer report. The responsibility of the METER System Manager is to select the parameters that will produce the kinds of reports that METER users need.

7.1 Stem Associations

Currently the stem associations report program asks for a single parameter, the minimum association value threshold. Only those associations exceeding this threshold will be be put into the report; lowering the threshold will result in more associations.

Because stem associations are close to being raw data, useful reports should probably be fairly short. The threshold should typically be set to at least 0.50 and perhaps higher in smaller or more homogeneous collections of messages. Short summaries of a few pages in length can be obtained with a high threshold of about 0.90. Some trial and error will be necessary, however, to adjust reports to a desired target size.

Association values in METER range theoretically from 0.00 to 1.00, but in practice, values below about 0.20 will be discarded by METER because they contain too much noise and would take up excessive amounts of secondary storage. Values of over 0.90 will be fairly common in message collections, although they often reflect only recurrent multi-word phrases in message text; for example, BUENAS AIRES.

The most interesting association values occur around 0.80. This reflects a highly significant degree of word stem co-occurrence in a large message

collection while reflecting some information content. Stems with higher associations would be good candidates for a stopword list.

7.2 Stem Clusters

Clustering algorithms are hit-or-miss by nature. There is no standard procedure for using them; successful application of them inevitably requires having a good feel for the data to be clustered and also a bit of luck. METER supplies three different clustering algorithms in the Message Summarizer Subsystem because it is impossible to guarantee that a single algorithm will always give good results.

- The Minimal Spanning Tree algorithm in effect clusters all stems that are associated with values higher than a given lower threshold. This is fairly fast and should work fairly well when stem association values are distributed unimodally about some central value. The METER implementation of this algorithm asks the operator to supply the lower threshold to be used. A value of about 0.60 or 0.70 usually is satisfactory.

- The Mutual Nearest Neighbor algorithm ignores associations values almost entirely, focusing instead on ranking stems according to how highly they associate; only the stems that both are highly ranked relative to each other will be clustered. This method is more robust in that it does not depend on the associations of a stem taking on a certain distribution in order to work well. It is somewhat limited, however, that the determination of nearest neighbors takes time and considerable table space. The METER implementation of this algorithm asks the operator to supply a neighborhood size; if two stems are related to each other with ranks adding up to more than the neighborhood size, they are not clustered. A value of about 5 or 6 usually is satisfactory; higher values may result in program failure due to table overflow, depending on

the total number of associated stems.

- The Tukey Gapping algorithm clusters stems according to associations values that fall into distinct clumps. It looks for the first significant gap is association values for each stem and clusters that stem with those having values above the gap. This algorithm is fairly robust, although it does work better when clumping is prominent. It requires a great deal of computation, but often produces the best looking clusters. The METER implementation of this algorithm asks the operator to supply a lower threshold on association values to look for gaps in. This should be set lower than a threshold for the Minimal Spanning Tree algorithm above; a value of about 0.40 seems to work out well.

In addition to asking for the values described above, all three algorithms ask for a "FAT stem" definition value. A FAT stem in METER is defined as a stem that is associated with more than a certain number of other stems, as specified by the definition value. FAT stems are treated in two special ways by METER clustering: first, no FAT stem is added to a cluster unless it is associated with a non-FAT stem according to the criteria established by one of the METER clustering algorithms; and second, FAT stems may be shared by more than one cluster without the clusters having to collapse into a single one.

A fairly low FAT stem definition value prevents clusters from getting too large and thus smearing out significant relationships between clustered stems. For most purposes, a value of about 20 to 30 works out all right; setting an extremely high value like 10000 that exceeds the maximum number of index stems in METER in effect turns off the FAT stem clustering feature entirely.

METER users will probably want to request a stem cluster summary by specifying the number of clusters in the output and the maximum size of clusters. There is no way of setting these directly, but a close approximation of them can be obtained by setting various the various clustering parameters already described. To get more clusters, the

association threshold should be lowered for the Minimal Spanning Tree and Tukey Gapping algorithms and the neighborhood size increased for the Mutual Nearest Neighbor algorithm. Increasing the FAT stem definition value will tend to give larger clusters. Everything will of course depend on the message collection; clusters cannot be found if they simply are not there.

The cluster report formatting program allows clusters of certain size to be suppressed by prompting the operator to specify that size. In general, users will not find clusters of size 2 or even size 3 very informative. The output will in any event be sorted by descending cluster size and will note the minimum cluster size printed.

7.3 Stem Trends

The METER stem trends program produces two separate reports: one of stems that are unexpectedly frequent or infrequent in an update with respect to recent history and one of stems that show a significant trend of increasing or decreasing normalized frequency over recent data base updates. The first report is based on a standard non-parametric statistical analysis of frequency distributions and requires no operator input; the second, however, must have some measure of what constitutes a significant trend.

The trends program will ask for a single value between 0.00 and 1.00 as a threshold for the net change in normalized frequency required over a series of updates before a trend is recognized. This threshold will be a percentage of the maximum normalized frequency of a stem over a trend period; a value of about 0.20 should work out here, but some experimental calibration will probably be necessary for best results. Raising the threshold will reduce the number of stems for which a trend is seen. Note that this threshold applies to both increasing (positive) and decreasing (negative) trends. Currently the trends program works with normalized frequencies over the 7 most recent updates; this is a compile-time parameter.

8. PERFORMANCE ANALYSIS

The METER system manager should closely follow the performance of the system to insure that METER updates are keeping up with incoming message traffic and that retrievals are acceptably fast and relevant. Deterioration in performance may call for remedial action such as revision of update schedules, redefinition of system parameters or tables, reorganization of data, software modifications, or even acquisition of additional hardware. It is important to diagnose the source of the problem correctly.

First of all, the trouble be external to METER entirely if speed is the problem. The host MAXI system may be heavily loaded or there may be system software anomalies somewhere. In this case, the only solution might be to postpone full updates and to limit access to METER according to some priority scheme. Such severe action might be necessary during a crisis.

Hardware problems can also arise to interrupt or slow down METER processing. The interruptions can in most cases be handled by correcting the problem and restarting, but slowdowns resulting from recoverable hardware errors may be hard to pin down. Lack of adequate free disk space may lead to a slowdown before space actually runs out.

If METER operating characteristics are the problem, then a change in parameters may be appropriate. There is, however, a trade-off between speed and coverage: fast operation means less coverage of the content of messages. Here, speed is easy to measure, but determining whether coverage is adequate for expected user queries takes close examination of what METER is doing.

If the problem is an error with METER software itself, then a trouble report should be filed to describe the nature of the problem so that METER system programmers can track down its source. This typically will not be

obvious in METER software. See the METER System Configuration Management Guide for a sample error reporting form and for a description of error correction procedures.

8.1 Instrumentation

All transactions at analyst work stations are monitored and logged in the file "EVALNS.DAT" maintained in each METER user directory. This provides information about the adequacy of message indexing for queries, the extent that METER statistics actually aid users, the time required to execute commands, and any user difficulties. See the METER Test Plan for more details. Programs for profiling performance instrumentation may be run by the METER System Manager from the METER console terminal. See the METER Operator's Procedures Manual for instructions on how to do this. Instrumentation output from retrieval modules may be turned off by installing special versions of the programs for users.

8.2 Data Base Summaries

Output from the METER message summarizer is extremely helpful in diagnosing METER problems. If index stems seem odd or if associations seem counterintuitive, then something may be wrong. The data base should be checked first in this case to see if it reflects unusual events. If the data base and its control parameters show nothing special then expert system programming help should be called in.

8.3 Operator Error Logging

A METER Log Book will be kept near the METER operator's terminal for recording unusual conditions in METER and reporting problems. Error messages from the message extractor subsystem will be routed to the METER system

operator's terminal. Hardcopy from updates and other procedures at the operator's terminal should also be retained for reference. The most serious problem to look for is running out of space. See the Operator's Procedures Manual for instructions on removing unnecessary files and for performing an update to purge the oldest messages in the data base.

Control File Manipulation

This appendix describes the control file, and how to make changes to it. Included is a sample METER control file.

Two METER operator programs allow reading and writing of the control file "CONTROL.DAT" in a data base directory.

DMPCON dumps a data base control file along with descriptive labels for each value defined in the file. Output is to a printable file called "CONTROL.TXT".

MAKCON converts a text file "CONTROL.TXT" into binary and writes this into a control file "CONTROL.DAT". MAKCON automatically resets all METER system locks when it is run.

The programs may be used to check the status of a system and to modify its operating characteristics. You must be in the target data base directory to run DMPCON and MAKCON.

To Print a Control File

For a data base in directory [270,5]:

```
MCR> HEL [270,5]
MCR> RUN [270,2]DMPCON<esc>
MCR> QUE CONTROL.TXT/DE
```

To Modify a Control File

This is similar to the command sequence in 2.3:

```
MCR> HEL [270,5]
MCR> RUN [270,2]DMPCON<esc>
MCR> EDI CONTROL.TXT
*

        (edit parameters)

*
*EDX
MCR> RUN [270,2]MAKCON<esc>
```

Below is a copy of a sample control file as formatted in CONTROL.TXT:

```
!METER system control file as of 9-APR-82
!Data base directory
[270,5]
!Base message number for labeling messages
0
!Minimum association threshold
0.20
!Association measure weight
0.50
!Context weight
0.33
!Message Acceptor Lock
unlocked
!Message Analyzer Lock
unlocked
!Number of updates in the METER window
14
!Maximum number of index words
3500
!Number of index words in current update
0
!Number of index words for retrieval
3194
!Number of message acceptor updates since the last full database update.
0
!Number of message acceptor updates currently being processed.
0
!Number of full updates processed by METER so far...
4
!Lower index message count threshold
10
!Upper index message percentage threshold
```

```
50
!Minimum message count for association
10
!Maximum message percentage for association
50
!Daily word co-occurrence threshold
10
!Date/time of the last message acceptor update (MM/DD/YY/Time)
!   where time = minutes after midnight
4
9
82
1194
!Year of the last full update
82
!Number of messages processed by the message acceptor
0
!Offset into micro-concordance file as of the last short update
0
!Number of messages entered into the database by the short update
0
!Number of messages in each of the METER full updates
! (Starting with the most recent update)
0
428
450
397
448
```

APPENDIX D

Meter Evaluation Questionnaire

THE METER SYSTEM EVALUATION

METER EVALUATION QUESTIONNAIRE

28 July 1982

Prepared Under Contract

F30602-80-C-0232

Submitted by

PAR TECHNOLOGY CORPORATION

1. Introduction

The primary purpose of this questionnaire is to provide the PAR Technology project team as well as evaluators from MAC and RADC with information related to the utility and/or potential utility of the METER system at the Military Air Command. More explicitly, the questionnaire may be perceived as supporting the METER evaluation by collecting information that can be used to assess the requirements for automated message analysis tools at MAC, the utility of METER, the joint efficiency/functionality of METER and MAXI, and as a side issue, the effectiveness of the METER training materials.

Your inputs to this questionnaire are important to the evaluation of METER. Most of the questions can be answered quite directly and do not require lengthy replies. Therefore, the questionnaire should not take too long to complete. Several questions, provide an opportunity for more lengthy responses and you may wish to write several sentences to complete them. If you wish to make any additional comments to a particular question or comments related to the evaluation in general, please use additional paper and attatch to the questionnaire.

Thank you for your cooperation.

2. General Questions

NAME/RANK_____

GROUP/ORGANIZATION_____ for example, INA

GEOGRAPHIC AREA or SUBJECT ASSIGNED TO YOU_____

DATE_____

1. How long have you been in your current role at MAC?

2. What general information needs do you have?

3. What information sources do you normally employ?

4. What job constraints are most imposing?

5. About how many new messages do you receive or do you review each day?

_____ messages per day.

6. Do you generally review messages at a MAXI terminal? YES ( ) NO ( )

7. Do you generally require messages of interest to be hardcopied?

   YES ( ) NO ( )

8. With respect to the messages you have access to, what time frame are you most interested? For example, the last 24 hours, the last 1-3 days.

For the next question, please fill in the blank.

9. If you are to rely on automatic message handling/analysis systems to provide timely support, then at a minimum, messages received as far back as _____ days should be available online.

10. Are you required to perform background searches covering large volumes of messages?

    YES ( ) NO ( )

11. If you do perform background searches, how far back in time would you normally be required to search as a minimum.

12. Do you prepare briefings? YES ( ) NO ( )

13. If you do prepare briefings please indicate how often you do and what type they are? (for example, the weekly commanders briefing)

14. How often do you use MAXI?

15. What do you use MAXI primarily for?

To review messages in message queues. ( )

To build/send messages. ( )

To write reports/briefings/etc.... ( )

Other uses. Please explain below.

16. Do you use MAXI to retrieve messages related to some information need you have.

YES ( ) NO ( )

17. Does MAXI provide you with a sufficiently useful message retrieval and/or analysis capability?

18. What additional features or enhancements would you like to see on MAXI that would provide increased analysis capabilities?

19. What do you expect METER to do for you?

3. QUESTIONS ON METER TRAINING AND SUPPORT DOCUMENTATION

1. Have you had a METER training session? YES ( ) NO ( )

IF YOU ANSWERED NO TO QUESTION 1 PLEASE GO TO QUESTION 5.

2. If you had a METER training session what organinzation conducted it?

   PAR TECHNOLOGY ( ) MAC ( )

3. How would you rate the training session you received?

   not very useful ( ) slightly useful ( ) moderately useful ( ) very useful ( )

4. Please enter below, in your own words, what you felt were some positive and/or negative aspects of the METER training session you had.

5. How much of the METER User's Guide have you read?

All or nearly all of it ( )

More than half ( )

About half of it ( )

Very little or none at all ( )

6. In your own words, how would you rate the User's Guide?

7. How often do you refer to the User's Guide before or during METER useage?

Nearly every time ( )

About half the time ( )

Occassionally, but not often ( )

Very seldom, if ever ( )

4. Questions On METER Performance

1. Are METER commands hard to use or remember?

2. Was the terminal hard to use?

3. Did you use the on-line or off-line HELP facilities?

4. What is your opinion of the METER forms (RESTRICT, REPORT and COMMENT)?

5. Did you understand METER output?

6. In the REPORT form, for your query, the relevance of each of the retrieved messages is indicated through the bar chart or relevance histogram. Also, the the content of the message is briefly stated using the four top matching stems.

   6a. Did you find the retrieval summary easy to understand?

   6b. Was the bar chart useful for deciding which message to look at?

   6c. Did the top matching stems provide you with a reasonable overview of what was in the retrieval?

   6d. Would you suggest any modifications in the REPORT form?

The following questions relate to the MAXI/METER interface. They are essentially directed towards the efficiency and utility of information flow between MAXI and METER.

7. Have you taken a message, briefing paper, or other piece of text from a MAXI message queue or work file and used it directly as a METER query?

7a. If you answered NO to question 7, are you aware of the key sequence that would accomplish it?

7b. If you answered YES to question 7, do you think transferring text from MAXI to METER is useful and important? (Why?)

7c. If you answered YES to question 7, do you think the transfer of text from MAXI to METER is efficient and easily initiated by the user?

8. Have you taken a message retrieved by METER and stored it directly in a MAXI work file?

8a. If you answered NO to question 8, are you aware of the key sequence that would accomplish it?

8b. If you answered YES to question 8, do you think transferring text from METER to MAXI is useful and important?

8c. If you answered YES to question 8, do you think the transfer of text from METER to MAXI is efficient and easily initiated by the user?

9. The METER system allows you to submit any piece of text as a query; for example, a message stored in a MAXI message queue could be used as a METER query after it had been passed to METER. Do you feel this free, unformatted text input is simple to use and effective?

10. The following table provides a list of the basic METER capabilities. Please indicate the degree to which you think each of the listed features is useful. The following scale is provided to assist in the ratings.

Not Useful	Neutral (not sure)	Somewhat Useful	Very Useful	Extremely Useful
1	2	3	4	5

Function	Rating from the scale above
1. On-line HELP facilities	_____
2. METER retrieval function (using natural text as a query; no formal query language is required)	_____
3. METER message header restriction (a RESTRICT function capability)	_____
4. METER keyword restriction (a RESTRICT function capability)	_____
5. METER SHOW function (a REPORT function capability)	_____
6. METER CULL function (a REPORT function capability)	_____
7. METER/MAXI information exchange (text transferal between the two systems)	_____
8. METER off-line statistical data base summaries	_____

11. Are you generally retrieving messages pertinent to your query?

12. Was response time too slow?

13. What would make the system more useful to you?

14. What capabilities should be eliminated?

15. Would you use METER in a crisis?

16. Based on your overall assessment of METER/MAXI would you like METER maintained at MAC in its present form or possibly upgraded? Please discuss this below.

APPENDIX E

Statistical Text Processing Techniques

E. STATISTICAL TEXT PROCESSING TECHNIQUES

This appendix contains technical notes describing the basic ideas underlying the design of METER. These are in nine sections:

1. FOUNDATIONS - an approach to characterizing text, stemming

2. SEGMENTATION AND SCALING - how to count word stems

3. STEM SELECTION - measuring the importance of word stems

4. WORD STEM ASSOCIATIONS - significant co-occurrence of word stems

5. WORD STEM CLUSTERS - patterns derived from significant co-occurrence

6. DOCUMENT REPRESENTATION - internal vector forms for processing

7. DOCUMENT SIMILARITY MEASURE - comparing documents by content

8. DOCUMENT PATTERN MATCHING - for retrieval

9. MODELS OF USERS - statistical characterization of interest

1. FOUNDATIONS

In statistical text processing, we attempt to characterize the content of text according to fairly small sets of numbers. This means that a certain amount of information must be lost, but in dealing with large numbers of documents, it may be impractical to analyze each in full, and in fact, most of these will probably be of no interest in any given situation. A statistical approach contrasts with the approach of trying to read text as a person might, which requires a system with a great deal of intelligence and world knowledge. A statistical approach can also incorporate intelligence and world knowledge, but because it needs this in far less detail to be effective, it tends to be more robust and easier to implement. For large collections of documents with heterogeneous content, a statistical approach is best.

In statistical analysis of text, the basic data consist of counts of the occurrences of words. Since this may involve tens of thousands of different words in a language like English, however, it is usually more reasonable to work with the stems of words instead. The stem of a word is obtained by stripping off its grammatical and morphological prefixes and suffixes; for example, the words EMITTING, EMISSION, and EMITS can be reduced to the single stem EMIT. It is possible to develop fairly comprehensive and accurate procedures for stemming words; the METER system incorporates a table-driven stemming procedure, which removes suffixes according to English spelling rules.

The METER stemmer is fairly sophisticated, having a procedure-based component for inflectional endings and a pattern-based component for other endings. Together these recognize about two thousand different cases of word endings. We are continually adding more cases to the stemmer as we have the opportunity to process different kinds of text and eventually expect to have about another several hundred cases. The development of stemmers is laborious, but given the noise inherent in statistical results, it does not

seem unreasonable to make an effort on the front end to get stems as good as possible.

Stemming has the advantages of greatly reducing the amount of storage required to process text and of yielding higher counts of occurrences for deriving statistics from. There are also disadvantages in that distinctions between word variants are lost and in that some degree of incorrect stemming is unavoidable and even highly likely for proper names, where standard spelling rules often fail to apply. On the whole, though, stemming fits in well with a statistical approach because information loss is inherent, and a small percentage of error in stemming can be tolerated.

From the standpoint of linguistic theory, an approach based on stem counts is a zero-th-order approximation to language. Meaning in language, however, is not only in the occurrence of words but also in the various relationships between words in text. Ideally, our basic data should also include counts of pairs, triples, and so forth, of co-occurring words, but this extension quickly becomes impractical. In a typical collection of with about ten thousand distinct stems, for example, we would have to store on the order of a hundred million items of data just to handle counts of distinct stem pairs. Dealing with stem triples would be altogether prohibitive.

Even statistics of only stem pairs is so costly from a computational standpoint that we need to be selective about what stems to work with. Fortunately, this is fairly straightforward to do. First of all, we can choose stems according to the requirements of a particular application. Furthermore, we can apply relatively simple statistical criteria to determine whether associations for a given stem are worth computing and whether associations once computed are worth retaining. With careful stem selection and with well-designed computational software, we can at reasonable cost derive intuitively sensible stem-pair data to support various kinds of content analysis.

In this paper we will examine in detail what we can actually do with data on word stem occurrences and stem pairs for large collections of documents. We will discuss methods for deriving such data from source text, and we will apply such data to applications such as document indexing, classification, clustering, discrimination, importance rating, and retrieval. The techniques involved here are fairly elementary, but call for careful attention to statistical assumptions and computational complexity.

2. SEGMENTATION AND SCALING

To statistics of text, we must first decide how to count word stem occurrences. This may appear to be a trivial matter at first, but it has serious ramifications for subsequent analysis because it will affect the range of numbers that we will have to deal with and the shape of their statistical distributions. A scheme for counting stems must take two key factors into account:

1. The length of documents varies greatly in terms of the number of words in them. The longest document in a large collection may be ten times or more longer than the shortest. If we are simply counting word stem occurrences in documents of different lengths, then the significance of these counts will be inconsistent.

2. Even in documents of the same length, we have the problem of how to compare counts for different stems: if one stem occurs four times more frequently than another, then should the statistical weight associated with the first be four times greater? Most statistical content analysis techniques will make these counts their starting point.

One approach to the first problem is to normalize stem counts according to the length of the document. For example, the computation of the Dennis measure for the discrimination value of stems for documents divides individual stem counts by the total stem count for a document, thereby transforming them into a percentage value. Alternatively, with a vector-space model where documents are interpreted as vectors and distance between vectors is defined as an inner product, we might normalize with the square-root of the sum of squares of stem counts. The second problem of stem count scaling can be handled by taking a non-linear function of stem counts to be our basic data instead of raw stem counts. Both of the problems listed above could be

addressed at the same time by breaking text into approximately equal segments for counting stems.

Our preference here is to avoid normalization because we can process text faster if we can work strictly with integer values or with scaled fixed-point values stored as integers. Floating-point numbers typically have twice as much storage as integers do, making data files larger and requiring extra computation to preserve the unnecessary precision. With 16-bit data, we have 4 decimal digits of accuracy, which is about as much precision as we can hope for in most aspects of statistical text processing. The storage of fractional normalized values also can lead to the accumulation of round-off errors in a system like METER where documents are processed in batches for the update of a collection.

Segmentation of text and functional transformation of raw counts do seem to helpful, and we like a combination of the two in most applications. First, documents are segmented according to the number of stems contained in them, ignoring "stopwords" that are defined beforehand as denoting no content of interest. Segmentation could also be put on a linguistic basis by parsing source text, but would be much slower. We currently use a purely heuristic segmentation scheme based on the "pyramid" model of text, where the initial portions of a document have greater significance in identifying its content. The scheme assigns the k-th stem in a document to segment n if $F_{n-1} < k < F_n$, where F_i is a Fibonacci sequence.

The length of such segments increases as a second-order polynomial function as we go farther from the beginning of a document. These segments may in turn be combined in various ways to obtain coarser divisions of the document as needed. The cost of segmentation is to increase the amount of storage for basic data by a factor equal to about the average number of the number of segments per document. A scheme with polynomially increasing segment size, however, will tend to keep the number of segments down while

allowing for short segments where most helpful.

Other segmentation schemes can be devised as appropriate for particular collections of text, though no one can really give any guidelines for doing this. Some possibilities are to segment according to stopword counts where fractional counts might be assigned to certain words, according to a distance from some established reference point within a document, or according to text features like the start of paragraphs, sections, and chapters. If documents in a collection all are fairly short or all have about the same length, segmentation might be dispensed with.

With respect to stem counts within segments, we currently take the ceiling function of their logarithms to the base 2 as our basic data (actually $\lceil \log_2 1+\text{count} \rceil$ so that counts of 1 go to a non-zero value). This scheme fits in with intuition based on information theory and also has some practical advantages: (1) high frequency stems are not overemphasized, (2) basic data consists of smaller numbers making overflow less likely in integer computations, (3) it ameliorates the problem of unequal document lengths, assuming that longer documents imply greater repetition of stems. Another possibility here is to have the ceiling of the square root of counts as our data transformation, which would fit in with a Fibonacci segmentation scheme.

3. STEM SELECTION

There are four major reasons in statistical content analysis for selecting subsets of stems from the overall set of stems in a collection of documents:

- Filtering out word stems with denotational content from purely grammatical words like "a," "the," and "of" or words like "system," which in a given context may have little content. These are commonly called "stopwords"; they are statistically useful to note in stylistic analysis of text, but not in content analysis.

- Getting highly discriminatory words to break up a collection of documents in as many different subsets as possible according to content.

- Deriving index words or groups of index word to characterize the information content of documents either for retrieval or for classification.

- Obtaining a class of stems for which a computation such as deriving association measures is statistically significant.

A fifth application, the identification of misspelled words, is also possible, but not discussed here because spelling correction in documents is best done when they are originally generated.

3.1 Filtering by Semantic Content

Many efforts have been made to recognize purely grammatical words by their statistics of occurrence in a collection of documents. These have generally been only marginally successful, the main difficulty being that documents vary enormously in their distribution of grammatical as well as

content words. A statistical procedure that works well for one collection may work poorly for another.

Given the relatively small number of purely grammatical words, it seems more practical simply to compile prior lists of the stopwords to filter out in a given application. Typically only several hundred words would be involved, even when extended to include low-content words known to occur frequently in a collection. Our approach here is to implement an efficient table-driven string matching procedure for recognizing stopwords. The table for this procedure is produced from a word list maintained in a text file that may be edited as required to tailor it to an application. The approach is not statistical, but works out well in practice. There is a danger that designated stopwords may turn out by chance to have major significance, but this can be minimized by periodic review of stopword lists. The computational advantage of stopword filtering is in any event compelling; it will eliminate about a half of all word occurrences from further consideration when processing documents written in fairly standard English.

3.2 Discriminatory Word Stems

Discriminating between subsets of documents within a larger collection is a fundamental part of document classification as well as a basis for retrieval of documents. Certain word stems are especially important for such discrimination from a statistical standpoint; these include stems that occur in only a few documents, stems that occur in fairly many documents but with a high variance, or stems with domains of occurrence co-extensive with document subsets of prior interest. Many statistical approaches have been developed for identifying such stems.

Our experience with such measures, however, is that they tend to be quite similar. On the several document collections we have worked with, the Spearman product-moment correlation between different measures of discrimination has always been above 0.5. If we choose stems according to

ranking by a measure of discrimination, the top N stems for any two schemes will differ by only about 10 percent, and so it is hard to say which selection is really better. The best strategy is probably to adopt the measure that takes the least computation.

The most simple measure is n_t , the count of documents that a stem t occurs in. For selection of stems, we simply put an upper percentage bound on the count to get those that mark a fairly small subset of documents and also a lower absolute bound to make sure that selected stems have at least some significance for statistical analysis. This helps greatly to keep the number of stems manageable for analysis because, according to Zipf's "Law", the number of stems that occur in a given number of documents tends to drop off inversely with the number of documents.

A second simple measure is the so-called inverse document-frequency defined as $\lfloor \log_2 N \rfloor - \lfloor \log_2 n_t \rfloor + 1$, where n_t is as above and N is the total number of documents in a collection. The inverse document-frequency measures the same thing as the simple n_t does, but tends to be better to work with because the measures for two stems will be of the same scale even for several orders of magnitude difference in frequency of occurrence. We could also argue for a logarithmic measure on an information theoretic standpoint.

An altogether different but also useful measure is the normalized variance of the frequency for a stem, conveniently computed as

$$\frac{\sum_d f_{t,d}^2}{\sum_d f_{t,d}} - \frac{\sum_d f_{t,d}}{n_t}$$

where $f_{t,d}$ is the frequency that stem t occurs in document d and n_t is as above. The variance computation usually is over an entire collection using N instead of n_t, but because most stems will occur in few messages, this

variance is deceptive in its implications. For example, consider the case of the N-variance with N=1000, $f_{t,d} = f_{t,d} = 1$, and increasing n_t.

n_t	1	10	20	30	40
N-Var	999	99	49	32	24

which plots almost in a straight line as long as the $f_{t,d}$ values are very much smaller than N as in fact they almost always are. N-variance on the whole will be of little interest over the range of frequencies for most stems of a document collection.

The Dennis measure, computed as $(\sum_d f_{t,d})$xN-variance looks better in theory, but has the same practical problem, as a few hand calculations will readily show. For fairly low stem frequencies, it merely changes the slope of the near linear plot of N-variance and thus offers little improvement. The n_t-variances provide much better data for further statistical analysis, although we have to be careful that n_t is large enough to make the n_t-variance significant.

3.3 Information Content of Stems

Semantic content is an absolute concept, the property of a particular word stem. Discrimination for a stem is always with respect to a given collection of documents and can be independent of semantic content; for example, the word "the" can be highly discriminating for terse telegraphic documents versus documents written in fairly formal English. Information content on the other hand measures conformance to expectations. Such expectations may be based in part on the particular documents in a collection, but in general, they involve the purpose and context in which documents are collected. For example, documents containing important information for some situation may in fact constitute a large portion of a collection and may be missed for that very reason.

Given a word stem, we have two basic ways of ascertaining information content. First, we can compare the distribution of occurrence for a given stem in a document collection against some hypothetical distribution of random occurrence and determine its departure from this norm. Second, we can look at how the statistics of a stem changes if a document collection happens to be continually updated. Both approaches pick out stems that stand out in some statistical sense.

For distributions to compare against, we have two fairly obvious possibilities: a uniform distribution representing a null hypothesis of arbitrary scattering or a Poisson distribution with a null hypothesis assuming occurrences patterned like radioactive decay events. In either case, for large document collections and stems of sufficient frequency, we can determine the derivation from a null hypothesis and its statistical significance. Stems with the greatest deriviation would be the most unexpected and in this sense carry the most information.

Trends of stem occurrence provide an information measure of an altogether different sort. By looking at previously collected documents, we can estimate the expected frequency of occurrence in a new set of documents of the same type. A stem with frequency falling more than several interquartile widths outside the expected interquartile range would be highly significant. Similarly, we can observe the changes in frequency and other statistics for a stem in samples of documents over a period of time; stems with definite rising or falling trends in frequency normalized for sizes of samples would be significant, since the null hypothesis would be no observable trend.

3.4 A Filter for Statistical Computations

Certain computations like the derivation of pairwise stem associations are extremely time-consuming. In a practical system, we would like to know beforehand which stems are worth computing statistics for so that we do not waste scarce machine resources. This can be done simply by restricting to

stems prefiltered according to other criteria and by specifying at least a lower limit on the frequency of a selected stem to assure that its statistics will at least have a sound basis, but from a practical standpoint, we can be even more restrictive.

For example, we may question the usefulness of computing associations for stems that occur in most of the documents of a collection. Although such associations will certainly be significant, they fail to tell us much than what we might know already. We are also uninterested in stems associated with most other stems, and so we will want to put an upper bound on the frequency of stems as a percentage of all documents in a collection. We will find this convenient in other statistical computations because it means less work.

4. WORD STEM ASSOCIATIONS

As a major part of content analysis, we compute associations for selected stem pairs within a document collection. Since this involves a number of values proportional to the square of the number of selected stems, we typically set a practical limit of 4,096 on this number, resulting in about 8.5 million associations to compute when they are symmetric. This is still a heavy computational load, however; it typically accounts for up to half of total processing time and may take about an hour to complete on a small processor like a DEC PDP-11/45. On the other hand, pairwise stem associations provide some extremely useful information for characterizing documents because they allow us to look at the statistical relationships between stems as well as the properties of single stems.

Tight time and space constraints favor simple calculations for associations. To get the best possible coverage of a document collection, it is more important to calculate as many associations as possible than to get precise associations for only a few stem pairs. Our current scheme is based solely on co-occurrence of stems within the same document segment, where segments in general may be allowed to overlap. The kind of segmentation depends on the application and the time frame in which computation is to be carried out. Finer grain segmentation requires retention of more data and therefore takes more processing. For time-critical applications, we may simply make segments equivalent to documents.

For each stem selected for association, we derive a vector of its frequency over each of the segments defined for a document collection. The association value computed for stems x and y is a normalized inner product.

$$A(x,y) = \frac{V_x \cdot V_y}{\| V_x \| \, \| V_y \|}$$

where Vx and Vy are the vectors derived respectively for x and y. This produces a N x N symmetric association matrix for N selected stems.

Symmetric associations are advantageous in having only half the matrix entries of asymmetric associations, but for some applications, we may want to pay more attention to the relative importance of stems being associated. For example, with two stems differing greatly in the sizes of their domains of occurrence in a collection, it may be useful to distinguish associations from a smaller to larger domain and vice versa. One possible approach is to multiply each now and each column of the association matrix by a weighting factor for the stem corresponding to the row or column.

$$A' = W \cdot A \cdot W^t$$

where W is an N x N diagonal weighting matrix measuring the domain of occurrence of each associated stem. The weights can be set to the interval [0,1] to preserve normalization of association values.

An association matrix may be derived as a composite result of various data. In the simplest case, we could just compute inner products within different subsets of stems; these inner products can readily be merged and then normalized to get a combined association matrix. The computation of inner products for subsets could be through entirely different segmentation schemes and in fact need not even be over the same document collection.

For example, we could choose to compute associations separately for high frequency and for low frequency stems, greatly reducing the number of inner products to be executed. We might also have a coarser segmentation with the high frequency stems to reduce computation further. Another possibility is to have extremely fine segmentation for static information like geographic associations, which typically can be computed once and for all. Associations

derived from the text of an atlas, for example, could be the basis for standard geographic "world knowledge" to be incorporated into different applications.

The quality of associations we have been able to obtain with the various techniques described here has been generally good with all the document collections we have worked with. A visual examination of derived association matrices will show that most of the associated stem pairs make sense without much explanation. Furthermore, where the connection between associated stems does seem obscure, this is usually a reflection of unexpected subject matter in a document collection and thus provides the kind of information we would want in a content analysis. For example, in a wire-service collection, an unexpected connection between the stems POPE and DETROIT turned out to arise from news items about the recall of an automobile presented to the Vatican as a gift.

In working with association matrices, we have found it advantageous to set a lower threshold on the entries to be retained because the lowest association values reflect mostly noise. We currently use a cutoff of 0.20, which has the effect of eliminating most computed associations from further consideration, but low association values do not seem to be especially useful in proportion to the space they take up. With most associations eliminated, we can store a matrix in sparse form on secondary storage and speed up applications referencing it.

5. WORD STEM CLUSTERS

Having computed stem associations, the logical next step is to cluster associated stems to see their global relationships more clearly. The stem associations derived by METER are fairly intelligible from a semantic as well as a statistical standpoint, but it is unclear where they come from and how they are related in text. The clustering of associated stems is in effect a reconstruction of the significant patterns of stems occurring within documents of a collection.

Clustering is more of an art than a science. As rule, application of different clustering algorithms will produce different sets of clusters from the same data, and even manual clustering of the same data by different persons will normally produce different results. Nevertheless, from various experiments we have run, the kinds of clusters obtained either automatically or manually seem to be significantly more similar to each other than to clusters obtained by random grouping.

Our approach to clustering to provide a choice of algorithms to try on a collection of documents with the hope that one of them or possibly a combination of them will yield information useful for a given application. Currently, we are working with three different algorithms:

- The minimal spanning tree algorithm is a hierarchical clustering scheme that puts a pair of stems into the same cluster if their measure of association exceeds a given threshold. The threshold can be set higher or lower to obtain a desired level of clustering.

- The mutual nearest neighbor algorithm clusters stems that are mutually close. This ignores actual association values other than using them to rank stems according to their closeness of association to a given stem.

Two stems are put into the same cluster if the sum of their respective ranks to each other is below a "neighborhood" threshold that may be specified at run time.

- The Tukey gapping algorithm clusters stems up to the first significant gap in associations between a given stem to other stems. This assumes that the gaps between association values for related stems have a normal distribution.

We are also experimenting with a version of the interactive Isodata algorithm, adapted to work with an association matrix as input instead of a set of points in n-dimensional space. So far, however, it does not seem to perform as well as our three other algorithms. The best results seem to come from the Tukey gapping algorithm although in some cases the nearest neighbor and spanning tree algorithms seem better.

These algorithms are all fairly standard techniques except that in content analysis they must be applied to thousands of stems, which as data have high intrinsic dimensionality (measured by the number of documents in a collection). In most clustering applications, the dimensionality is around a hundred at most, and so it is feasible to work directly with vectors. With stems, we use the vectors only to compute the associations from which clusters will be derived. Clustering starting from associations is fairly fast, requiring only a few minutes even on a mini-computer.

Normally, the algorithms as described produce disjoint clusters, but disjointness may not always be appropriate for content analysis of text. Important patterns in text, for example, will generally have stems in common; and so overlapping clusters would be more appropriate for capturing content. The easiest way to get overlapping clusters is to separate stems into two categories according to the degree that they associate with other stems. Highly associated stems in a document collection would be those that we would

want to put into more than one cluster; otherwise they would tend to pull stems into a small number of clusters. Our approach would be to cluster less associated stems as before, but to allow highly associated stems only to be added to existing clusters.

A set of stem clusters produced for a document collection in effect identifies major topics arising in a collection. By measuring the degree that a cluster pertains to a given document, we can determine how much that document is about the topic associated with the cluster. With a set of clusters, we could characterize the content of a document in the same way that we would do with a set of index stems. This is particularly useful in getting summaries of documents because the number of clusters will be much smaller than the number of stems and represent more general content of documents.

6. DOCUMENT REPRESENTATION

To work with documents, we can represent them in two different ways. The first is to choose a set of N index stems according to some criteria and to characterize a document according to the frequency that each index stem occurs in the document. This yields a vector of N integer components that we may now process numerically. The second way is similar, but starts from a set of index stem clusters. The idea here is to compute a measure of pertinence for a document with each cluster in a set to get a numerical vector, usually of lower dimensionality than that for index stems. We can think of each cluster as representing a particular topic covered by the documents of a collection.

The key to deriving pertinence values in the second approach is that each cluster of stems can also be treated as a numerical vector. In this case, each vector component would correspond to the degree to which a certain stem belongs to a cluster, where the degree of belonging could be the number of links for a stem with other stems in the cluster. Given the vector X_k for the kth cluster we can measure the pertinence p_k^D of the cluster to a given document D as a function f of the cluster vector and a corresponding vector V_D of stem frequencies for the document.

$$P_K^D = f\ (X_K,\ V_D)$$

We may then represent document D as a vector $(P_1^D,\ P_2^D,\ \ldots,\ P_M^D)$, where M is the number of clusters. This will be called a topic cluster vector to distinguish it from an index stem vector for the same document.

There are various choices for the function f above. The simplest is to make it the normalized inner product of its vector arguments. This turns out to work fairly well for ranking clusters according to their importance to a

given document and has in fact been employed in several document classification and retrieval systems. The scheme has a deficiency, however, in that there is no good way of comparing the relative importance of a single cluster for two different documents.

Suppose that we have a cluster X and a document D such that the importance of X for D is a positive value. If we then construct a new document DE by appending a section E that is completed unrelated to X, the the normalization of the vector V_{DE} in the inner product computation results in reducing the overall importance of cluster X. This kind of situation is unsatisfactory when we want it possible for the content of a document to encompass two or more independent topics represented by distinct clusters. For this purpose, we want a function f that is less sensitive to the length of a document.

Another problem with the normalized inner product is that its value may be fairly high even when a document contains only a few of the stems in a cluster. This property is a manifestation of the triangle inequality that must hold in order for a metric to be definable in a vector space. Because of the inherent relatedness of stems in a cluster based on associations, however, we would want the opposite property. That is, we would want function f to take a high value for a cluster only when almost all of its stems are present in a document and want its value to drop rather sharply when even only a few stems fail to show up.

The above requirements suggest that we in effect turn the inner product approach inside out and also change the normalization. This can be accomplished by computing a penalty function f^- for the non-occurrence of cluster stems instead of a net relevance function for the occurrence of stems.

$$f^- = [\ \Sigma w_s^2\ ]^{1/2} \quad , \text{ for } s \text{ in } X_k \text{ and in } D$$

where w_s are weights assigned to stems s in cluster X_k. The value w_t will measure both the degree to which a stem t belongs to cluster X_k and the likelihood that stem t may not occur in a document. The latter consideration is important because we do not want to penalize greatly for the non-occurrence of a stem unlikely to occur.

We can normalize the function f^- to the interval [0,1] by dividing the weights for each stem s in cluster X_k by Σw_s^2, the sum of the squared weights for all stems in the cluster. We may then define our cluster pertinence function as follows

$$f(X_k, V_D) = 1 - f^-(X_k, V_D)$$

There are a number of important differences for analysis between the index stem vector and topic cluster vector representations of a document. We have already noted that index stem vectors will generally have higher dimensionality than topic cluster vectors. This means that index stem vectors will require more storage, but the more compact topic cluster vectors are obtained at the cost of a great deal of extra computation. If we need to make documents available for analysis or retrieval as soon as possible, then index stem vectors would be our choice.

Another important difference is that index stem vectors will usually have zero values for most of its components, and its non-zero components will take discrete as opposed to continuous values. It is generally difficult to interpret such vectors in terms of a multi-dimensional metric space. Topic cluster vectors, on the other hand, will take on continuous component values for all intents and will have a smaller proportion of zero components. They also have an advantage in that their various components will be fairly independent of each other, unlike stems chosen to index documents, which may

be somewhat synonymous. There is a problem with the cluster representation, however, in that it is likely that a document will have no clusters pertaining to it whereas it is unlikely that the document will contain no index stem.

Topic cluster vectors have the appropriate properties for application of classical statistical pattern analysis techniques. It is easy to set the threshold on clustering to get a small enough number of clusters so that the dimensionality of vectors is small enough for analysis such as eigenvector derivation, and it is not essential that every vector be related to some cluster. Topic cluster vectors can be input to pattern analysis systems like OLPARS, which use interactive graphical displays to show the structure of data for purpose of classification.

7. DOCUMENT SIMILARITY MEASURE

Given a vector representation of documents, we have various ways of measuring their similarity. A common approach is again to take the normalized inner product of document vectors. This is easy to compute and works out fairly well in practice to gather documents of similar content, although the theoretical justification is sometimes shaky. For document retrieval applications, we have adopted the normalized inner product metric while rejecting the usual multi-dimensional vector space explanation for it.

In a common vector space approach, a metric is defined (in normalized form) as

$$M(u,v) = \frac{\sum_i u_i \cdot v_i}{[\sum_i u_i^2 \cdot \sum_i v_i^2]^{1/2}}$$

This is the "cosine" measure, which interprets the distance between documents as a kind of angle in N-dimensional vector space. For the measure to be valid, however, we must assume that the basis for the document vector coordinates is at least orthogonal and preferably orthonormal. Unfortunately, we know that index stems are generally not even orthogonal in any sense because we do not control synonymy of stems. The situation is slightly better for a cluster representation of documents, but there are still problems with scaling of the vector space defined by clusters and with orthogonality when clusters overlap.

Nevertheless, the cosine measure works out fairly well in practice. In the case of index stem vectors, we need only to make sure that the vector coordinate values for stems are weighted according to their discrimination value in order to get good results; that is, the vectors (u_i) and (v_i) above would have to be replaced with $(W_i u_i)$ and $(W_i v_i)$ respectively, where W_i is a

discrimination measure for stem i. The cosine measure may work out because in fact it approximates a correlation measure in the case of sparse vectors, which is typically so for index vectors. The Spearman product-moment correlation between N-vectors u and v is

$$M'(u,v) = \frac{\sum_i (u_i - \bar{u})(v_i - \bar{v})}{[(\sum_i u_i^2 - N\bar{u}^2)(\sum_i v_i^2 - N\bar{v}^2)]^{1/2}}$$

$$M'(u,v) = \frac{\sum_i u_i v_i - u_i\bar{v} - \bar{u}v_i + \bar{u}\bar{v})}{[(\sum_i u_i^2 - N\bar{u}^2)(\sum_i v_i^2 - N\bar{v}^2)]^{1/2}}$$

$$M'(u,v) = \frac{\sum_i u_i v_i - \bar{v}\sum_i u_i - \bar{u}\sum_i v_i + N\bar{u}\bar{v}}{[(\sum_i u_i^2 - N\bar{u}^2)(\sum_i v_i^2 - N\bar{v}^2)]^{1/2}}$$

$$M'(u,v) = \frac{\sum_i u_i v_i - N\bar{u}\bar{v}}{[(\sum_i u_i^2 - N\bar{u}^2)(\sum_i v_i^2 - N\bar{v}^2)]^{1/2}}$$

For sparse vectors with only a few non-zero entries, we have the mean value of entries $\bar{u}$ and $\bar{v}$ approximately equal to zero. If we drop the $N\bar{u}\bar{v}$, $N\bar{u}^2$, and $N\bar{v}^2$ terms, then we get:

$$M'(u,v) = \frac{\sum_i u_i v_i}{[(\sum_i u_i^2)(\sum_i v_i^2)]^{1/2}} = \text{Cosine Metric}$$

Other document similarity measures we have considered include the generalized Euclidean metric

$$M^E(u,v) = \sum_i | u_i - v_i |^c \quad ; c = 1,2,\ldots$$

and the Mahalounobis distance

$$M^M(u,v) = \sum_{ij} u_i v_j C_{ij}$$

where C_{ij} is the covariance between vector components i and j. These have their disadvantages: the Euclidean measure leads to problems of scaling, and the Mahalonoubis distance is computationally costly.

8. DOCUMENT PATTERN MATCHING

Pattern matching is commonly required in applications such as retrieving relevant documents from a data base or for filtering them out of an input stream. In a statistically organized system, this implies the computation of some measure of similarity between documents and a pattern. An obvious approach here is to have a pattern take the same form as a document and then to use document similarity measures as described in Section 7. Such a scheme turns out in fact to be workable, but in practice, complications arise.

(1) To begin with, patterns will usually contain less information than a document. When a pattern is not itself an actual document, it will tend to have many fewer words (be sparser) than a typical document. The similarity between a pattern and a document therefore will probably be low, making it hard to tell when a match is good.

(2) Document distance metrics as defined in Section 7 tend to be hard to scale. Although relative distances with respect to a single document may be significant to a certain precision, distances measured relative to different documents will be hard to compare. It will generally be impossible to set any distance threshold that means the same thing for all patterns and all documents, making it difficult to calibrate a pattern to minimize the probability of false matches.

(3) Most applications require fast pattern matching. In order to provide fairly immediate responses to search requests over large numbers of documents, we have to be able to limit the amount of computation involved in matching a document against a pattern.

These three considerations make pattern matching a nontrivial problem if we seek the best results. This section will describe a number of techniques that we have found effective in improving pattern matching at least from the

standpoint of document retrieval.

8.1 QUERY EXPANSION

In the METER system, a query for retrieval of documents is a vector of stems matched against corresponding vectors for documents being searched. The original motivation for query expansion in METER was the problem of including synonyms in queries. Inadvertent omission of a synonym by a user might result in an important document not being retrieved, and so, it seemed helpful to automatically complement a query with the first-order associations of its stems--those stems tending to occur with query stems in a given collection of documents. This seemed to improve retrieval performance in various experiments run with METER, but not in the way expected.

To begin with, we never found in experiments with various collections of documents any example where first-order associations resulted in retrieving a message that otherwise would have been overlooked. If associations do in fact improve the recall aspect of retrieval, then it must do so only in rare circumstances. Expansion with stem associations turned out instead to be helpful in improving the precision of retrievals. Although the total number of relevant documents seemed to be about the same with or without expansion, the addition of associations clearly made relevant documents move up in a retrieval listing ranked by estimated relevance. This effect was consistent enough that we felt justified in retaining query expansions in METER despite the cost in computing them and the fact that they seemed to be of little use in improving recall.

Improved precision may come about because an expanded query vector will have about the same number of non-zero coordinates as document vectors. As a rule of thumb for conserving file space, we generally set a lower threshold on retained association values to get an average of ten to twenty non-zero associations per stem; this also is about the number of times more stems a

typical document has versus a typical query. When query vectors and document vectors are about equally sparse, measuring a distance between them becomes more reasonable. We have found that, when a query is already comparable to documents in stem count, there seems to be no advantage in further expansion.

Query expansion in METER is currently defined as

$$q' = A \cdot W \cdot q$$

where A is a symmetric matrix of associations, W is a diagonal weighting matrix, and q is a query vector. The weighting is necessary to compensate for some words having many more associations above threshold than other words and thus getting too much emphasis in an expansion. We favor the so-called "inverse document frequency" measure for weighting, defined for a stem t as

$$\log_2 | N | - \log_2 | n_t | + 1$$

where N is the total number of documents in a collection and n_t is the number of documents containing stem t. This seems to work because the number of associations for a word tends to be positively correlated with the number of documents that word occurs in. The weighting also takes into account the discrimination value of each stem individually.

Other expansion schemes were evaluated early in the development of METER. Of note here perhaps is

$$q' = A^2 \cdot q$$

an expansion by second-order associations employed at the start of the METER

effort. The rationale for this came from noting that first-order associations indicate the semantic context of a word; second-order associations therefore relate words that occurred in similarly contexts, thus capturing synonymy between words at least in the classic structural linguistic sense. Unfortunately second-order associations never proved to be as good as first-order associations for query expansion in experiments and take much more time to compute. If our notion of comparable vectors here is valid, it may be that expansion with second-order associations yields too many non-zero coordinates in an expanded query vector.

Expansion with first-order associations runs into one major theoretical problem. This involves the value to be assigned to the main diagonal of matrix A, the association of something with itself. For stems, the value would be identically 1.0 along the diagonal if associations were actually computed on the basis of co-occurrence with normalized inner products. Such a value seems too high, however, if we want associations between different things to contribute significantly when multiplying by the matrix A. In psychological research on associations, for example, values like 0.5 are typically assigned to the diagonal of association matrices on theoretical grounds. In actual retrievals, though, a value of between 2 and 3 gives best results.

This result is nonsensical because associations are supposed to fall between 0.0 and 1.0. At first we tried to explain it away by attributing it to high statistical noise in associations that required compensation, but we then found that things remained the same even after considerable reduction of noise through improved stemming and retention of more precision in computations. We then tried to further reduce noise by limiting expansions to the top K associations for each stem; this had computational advantages in reducing the possibility of overflows when using sparse vector representations and in speeding up processing, but we were still unable to reduce the diagonal values giving the best results.

A more successful hypothesis was that the anomalous diagonal values reflected the occurrence of high association values that had to be downweighted for best results, a plausible idea when we note that adding a stem to a query cannot help if has the same distribution as a stem already in the query. To test this hypothesis out, we applied a non-linear weighting function w to non-diagonal entries of matrix A

$$w(x) = a \cdot (x - 1.0) \cdot (x - b)$$

where b is the lower threshold for associations and a is a normalization coefficient. This turned out to improve retrievals significantly, but a diagonal value between 2 and 3 still turns out to give best results. We currently have no good ideas for resolving the problem of the diagonal values. The impasse highlights the fact that we still lack a good theory of how statistical document retrieval works.

8.2 RELEVANCE MEASURE

Because of its simplicity and ease of computation, we currently employ the cosine measure for the relevance of a document to a query.

$$R = \frac{q \cdot d}{||q|| \; ||d||}$$

where q is a query vector and d is a document vector. We have experimented with other measures, but they do not seem to work out as well. A Mahalonobis measure

$$R = \frac{qCd}{[qCq \; . \; dCd]^{1/2}},$$

for example, is simply too slow for interactive retrievals with any collection of reasonable size; and it also fails to be an approvement over the cosine measure, although this may be because the covariance matrix C in our experiments actually measured only co-occurrence.

Another unsuccessful alternative to the cosine measure was a relevance mass measure. This assumes a certain increment of relevance is associated with each query word and total relevance is the sum of the weights of words occurring in a given document, after subtracting out possible overlap because of synonymy. The relevance computation is based on the inclusion-exclusion principle of sets. Cut off after corrections for overlap between pairs of stems in the manner of a Taylor series expansion, this becomes

$$R = \frac{(1\text{->}d) \cdot q}{\| q \|} - \frac{(1\text{->}q)\ A\ (1\text{->}q)^t\ q}{\| q \|}$$

where 1 is the vector of all 1's and u->v is a projection vector consisting of the non-zero coordinates of u corresponding to non-zero coordinates in v; the approximation here assumes that the overlap between any two stems is relatively small. This measure turned out to be significantly worse than the cosine measure for recall in experiments although its precision tended to be high. Given its complexity as well, we had to drop it from further consideration.

A more promising alternative was a simple projection measure, a variation on the cosine measure

$$R = \frac{q \cdot d}{\| q\text{->}d \| \cdot \| d \|}$$

In the cosine measure, documents with the same number of words as an expanded query are favored. Here the slight change in normalization results in looking

only at the degree of match between a query and some portion of a document. Experimental results showed, though, that the cosine measure was again slightly better; from examination of the document collections in such experiments, we have tentatively concluded that this means that it was better to favor documents of intermediate length rather than shorter or longer ones. Intermediate length documents appear to be most likely to be relevant to the general types of queries employed in our experiments.

The cosine measure in the end also turns out to be the fastest approach of those considered here, an important consideration for interactive retrievals. With fixed-point computations implemented in assembly language and supported by special I/O for overlaying large blocks of data, we can search a thousand document vectors with about seventy non-zero coordinates each in one second on a DEC PDP-11/70 processor with an RP06 disk for storage. This is about twice as fast as reported for some systems on larger processors augmented with special array processing hardware.

8.3 RELEVANCE FEEDBACK

In statistical document retrieval, we are by definition using a shotgun approach to identify relevant documents, and the process of query generation attempts to optimize the chances of getting a hit. There are various possible approaches to accomplishing this, but unless we have some independent information on what documents to look for, the best bet is to start with as broad an aim as possible on the first try. If anything relevant is found, then we can readily home in on it, using a concept known as "relevance feedback."

In relevance feedback a user specifies certain retrieved documents as relevant and others as irrelevant. The retrieval system then revises the original query according to this information and retries the retrieval. In the statistical version of relevance feedback with the notion of a distance measure, this usually involves adding the mean vector for relevant documents

to a query and subtracting the mean vector for irrelevant documents. The idea here is quite plausible, but in practice, we found it of limited value.

Although this scheme appears to work for one or two iterations, it soon converges to where further relevance information from the user makes no difference at all in documents retrieved. Such rapid convergence would be attractive if retrievals actually improved significantly, but typically the final result in our experience was only a slight increase in precision with no better recall. In fact, retrievals may even deteriorate markedly after several iterations.

The problem appears to be the recurrent one that a vector space distance measure for documents has to be interpreted with care. The alternate idea of a similarity measure based on sparse vectors with roughly the same number of non-zero coordinates suggests a different kind of relevance feedback. It should be noted that summing many document vectors to compute a mean vector because this tends to yield a vector that is non-sparse (dense). Once this happens, the cosine computation no longer approximates the correlation computation, which is our theoretical justification for the cosine measure in a vector space with an obvious nonorthogonal basis.

A better implementation to iterative refinement of query vectors is limit adjustments to non-zero coordinates of the original vector and to make adjustments in proportion to the original weightings of stems in the query.

$$q' = q + c \cdot \frac{\sum_{\text{relevant}} d_r^t q - \sum_{\text{irrelevant}} d_r^t q}{2}$$

where c is a coefficient determining the magnitude of adjustment in an iteration. With this facility, it has been possible in METER to iterate as many as 6 to 8 times on a query without convergence in improvement on retrievals. Moreover, the facility appears to work well even when a user

specifies only irrelevant documents. Combined with a fast retrieval search, relevance feedback becomes a kind of browsing capability for users. Although there are problems in that a browsing user can get eventually stuck in a cluster of documents with highly similar content, the facility is still extremely helpful in making a retrieval system more friendly to users.

8.4 KEYWORD EQUIVALENTS

A common concern voiced about the selection of a small set of words to index documents for retrieval is that a user may sometimes want to retrieve documents according to words not selected for indexing. In METER, a partial resolution of this problem was the incorporation of a special keyword retrieval module not restricted to index stems; this is similar to conventional Boolean document retrieval systems now commercially available and also has the capability of its output being used as a source file for standard METER retrievals or being simply clustered to group documents by content for examination.

Introducing a separate keyword retrieval facility, however, breaks up the flow of user interaction with a statistical system. For example, if an important query word corresponds to no index stem, a user must invoke a special command to collect documents containing the word. If there is only a few of such documents, then a user might choose to look at all of them, but if there are many, they have to be fed back into the interrupted query sequence. This becomes awkward when there are two or three non-index keywords each occurring in different documents because it is unclear whether to take the union or intersection of the documents for each keyword.

An alternate approach is to reduce the need for a user ever to resort to the keyword retrieval facility. There will be times when that facility may be unavoidable, but usually we can compensate for certain words not selected for indexing by a statistical technique that simulates having those as index-

words. The idea is a simple application of the structural linguistic notion that a word is defined in meaning by its context. Here a context could be those index words that a non-index word occurs with in a given sample of documents. So instead of putting the non-index word into a query, we add in an index word contextual equivalent vector to the query vector.

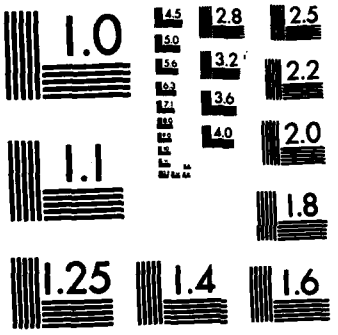
This technique is complicated by the need to avoid a query vector becoming too non-sparse in the process. We can select the most salient components of a contextual equivalent vector x_s for a stem s by squaring the value for each coordinate and normalizing to drop out lower weighted words.

$$x_s = w_s \sum_{\text{non-index}} \frac{d_n \cdot d_n^t}{\| d_n \cdot d_n^t \|}$$

where w_n is a weight associated with a nonindex stem in a query and the d_n are selected documents containing a stem. In the case of several stems, we would simply add their equivalent vectors together. Sparseness can also be controlled by computing equivalents only for words occurring in no more than a certain number of documents, making it more likely that such documents will have something in common. With extremely frequent words, this technique becomes less effective, but then a keyword retrieval facility can be used effectively.

AD-A125 166 OPERATIONAL TEST AND EVALUATION OF THE METER ENGINEERING DEVELOPMENT MODEL(U) PAR TECHNOLOGY CORP NEW HARTFORD NY R J D'AMORE ET AL. NOV 82 4/4

UNCLASSIFIED RADC-TR-82-304 F30602-80-C-0232 F/G 14/2 NL



9. MODELS OF USERS

Although our orientation here has been to employ statistics primarily to describe the general content of a document collection, we could equally well apply our various techniques to describe users. Augmenting a statistical content analysis system with this kind of data can make it more responsive to user requirements in the selection of index stems, in the derivation of stem associations, and in the determination of the significance of documents. A statistical approach inevitably must filter out information; knowledge of users allows this to be done more judiciously.

Data for a user model would basically be either about the word stems important to a user or about the kinds of documents of interest to the user. We can obtain this data by monitoring the interaction of the user with a system such as in the query generation and retrieval process and noting the kinds of documents a user wants to see versus those in a given collection. The idea here would be to model where user interests diverge significantly from the emphases implicit in the overall statistics for a document collection.

9.1 STEM DATA FOR MODELS

Most information retrieval systems contain online instrumentation to aid a manager in evaluating their operation and in configuring a system for best performance under changing conditions. Such instrumentation data can readily be incorporated into a statistically based system as part of a model of users. In a system like METER with a user interface for retrieval of documents, three main kinds of data are available:

- Non-index stems chosen by a user for inclusion in a query through their index stem equivalents; such stems would ostensibly be important to the user, but were not statistically important enough in a collection to be

selected for indexing documents.

- Specific stems entered to restrict retrievals to documents containing them. These stems may or may not have been chosen for indexing; they reflect important logical divisions in a user's view of the organization of a collection of documents.

- Index stems downweighted or upweighted as the result of relevance feedback. If the stem was from the original user query, then this reflects an adjustment of system stem weighting. If the stem was not, then this reflects an adjustment of stem associations.

If accumulated usage data for a user show consistent patterns, it becomes possible for a system to adjust document collection statistics when processing for that particular user. If such patterns hold for a number of different users, then it may be advantageous to make global changes in collection statistics; for example, we might want to select a stem for indexing and give it a high weight regardless of its actual distribution within a collection.

9.2 DOCUMENT DATA FOR MODELS

In a text-based system, the basic information provided to a user is in the form of documents. When certain documents are known either implicitly or explicitly to be of interest to a given user, it becomes possible to hypothesize what other documents might be also of interest and how to characterize these so that a system might recognize them. All we need is to be able to obtain large enough samples of the various documents of interest from which to derive statistics.

Document subsets of interest may be derived in many ways. For example, there are the retrieved documents specifically printed out by a user for reference, those selected by a user for generating subsequent queries, and those collected by a user for further analysis. Or, a user may simply identify a subset of documents specifically to support the building of a user model. Once a subset is defined, it becomes possible to apply the the various techniques of statistical content analysis to get a description of it.

A subset need not actually be drawn from the documents of a given subset. In systems where users can create their own work files of text, we can process these as if they were documents, and when work files happen to be saved under major headings, we would have a good basis for characterizing specific areas of interest for a user. Another possibility is to predefine an area of interest according to gathered background material; for example, extracting text from atlases and other reference sources to characterize a particular geographic interest.

In addition to the discriminant analysis for document subsets described in Section 9, we can compute statistics for a subset as a subcollection and to compare these with those of a larger document collection. This is the classical type of content analysis for documents and provides some general information about a subset to complement information about how it differs from something else. In the case of a subset external to a collection as in the geographic example above, it may be advantageous to incorporate the statistics of the subset into those for the collection. In this way, a model of users would become in effect part of the general world knowledge built into a system.

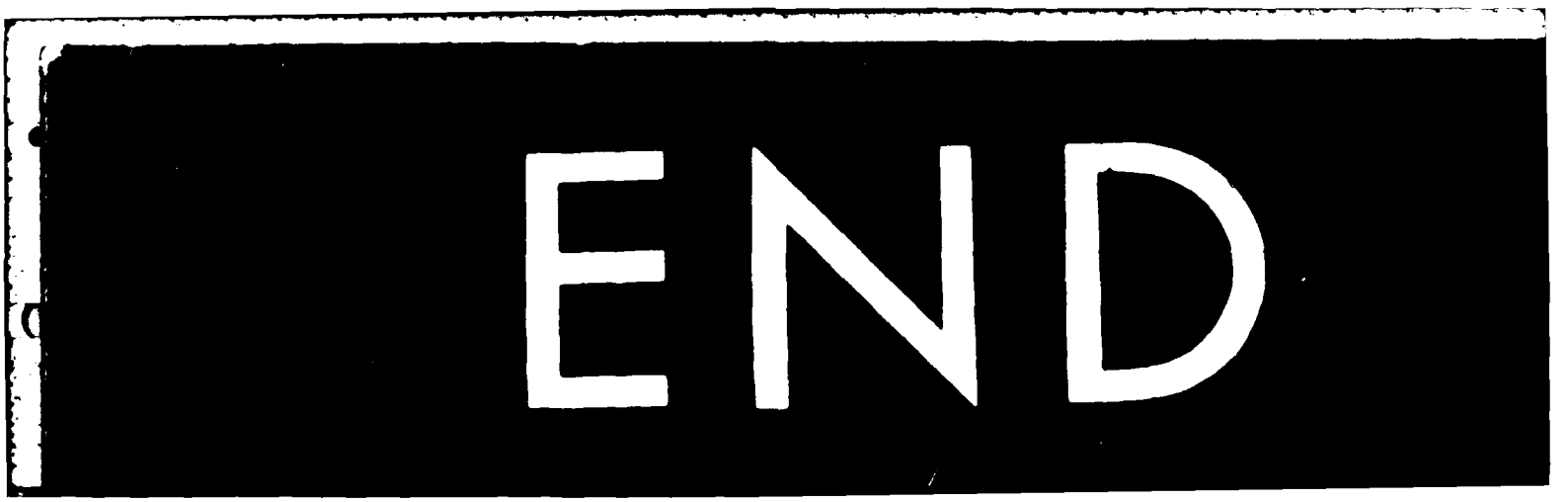
Document subsets of interest may be derived in many ways. For example, there are the retrieved documents specifically printed out by a user for reference, those selected by a user for generating subsequent queries, and those collected by a user for further analysis. Or, a user may simply identify a subset of documents specifically to support the building of a user model. Once a subset is defined, it becomes possible to apply the the various techniques of statistical content analysis to get a description of it.

A subset need not actually be drawn from the documents of a given subset. In systems where users can create their own work files of text, we can process these as if they were documents, and when work files happen to be saved under major headings, we would have a good basis for characterizing specific areas of interest for a user. Another possibility is to predefine an area of interest according to gathered background material; for example, extracting text from atlases and other reference sources to characterize a particular geographic interest.

In addition to the discriminant analysis for document subsets described in Section 9, we can compute statistics for a subset as a subcollection and to compare these with those of a larger document collection. This is the classical type of content analysis for documents and provides some general information about a subset to complement information about how it differs from something else. In the case of a subset external to a collection as in the geographic example above, it may be advantageous to incorporate the statistics of the subset into those for the collection. In this way, a model of users would become in effect part of the general world knowledge built into a system.

MISSION
of
Rome Air Development Center

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


END