

BBN REPORT NO. 3129

REVISED edition

DIALOG SPECIFICATION PROCEDURES

Richard W. Pew  
Ann M. Rollins

30 September 1975

Prepared for

Applications Development Branch  
Data Systems Division  
Agricultural Stabilization and Conservation Service  
U. S. Department of Agriculture  
Room 4126 Auditors Building  
Washington, D. C. 20250

Under Contract No. 123351034

## TABLE OF CONTENTS

	<u>Page</u>
SECTION 1 - Introduction	
1.1 Background - - - - -	1
1.2 Purpose and Scope- - - - -	1
1.3 Applicability- - - - -	2
1.4 Terminology- - - - -	3
SECTION 2 - General Principles of Effective Dialog Design	
2.1 Know the User Population - - - - -	4
2.2 Respond Consistently and Clearly - - - - -	5
2.3 Carry Forward a Representation of the User's Knowledge Base - - - - -	6
2.4 Adapt Wordiness to User Needs- - - - -	6
2.5 Provide the Users with Every Opportunity to- Correct Their Own Errors- - - - -	7
2.6 Promote the Personal Worth of the Individual User - - - - -	7
SECTION 3 - Interactive Terminal System Characteristics	
3.1 System Response Time - - - - -	8
3.2 Terminal System Memory and Processing- - - - -	8
3.3 Terminal Capabilities- - - - -	8
SECTION 4 - Recommended Practices for the Design of Dialog	
4.1 Menu Selection - - - - -	12
4.2 Input Frames - - - - -	12
4.3 Output Frames- - - - -	12
4.4 Permissible Frame Combinations - - - - -	13
4.5 General Recommendations for Frame Specification- - - - -	13
4.6 Recommendations for Display of Menu Frames - - - - -	14
4.7 Menu Frame Control - - - - -	16
4.8 Recommendations for Display of Input Frames- - - - -	17

## TABLE OF CONTENTS (cont)

	<u>Page</u>
4.9	Input Frame Control - - - - - 21
4.10	Recommendations for Display of Output Frames- - 21
4.11	Output Frame Control- - - - - 22
4.12	Recommendations for Error Messages- - - - - 23
 SECTION 5 - Steps in Systematic Dialog Development	
5.1	Task Analysis and Flow Chart of Transaction from User's Perspective - - - - - 27
5.2	Flow Chart of Required Dialog Frame by Frame- - 28
5.3	Specification and Layout of Dialog Frames - - - 28
5.4	Specification of User Supplied Variable Fields- 28
5.5	Specification of Possible Error Messages Associated with Each Field - - - - - 29
5.6	Documentation - - - - - 29
 SECTION 6 - Dialog Documentation Aids and Instructions for Their Use	
6.1	Dialog Frame Specification Form - - - - - 30
6.2	Variable Definition Form - - - - - 32
6.3	Error Message Specification Form- - - - - 35
 SECTION 7 - Sample Dialog for Commodity Loan Repayment Subsystem	
7.1	Introduction - - - - - 37
7.2	Transaction Task Flow Chart - - - - - 38
7.3	Dialog Flow Chart - - - - - 39
7.4	Dialog Frame and Variable Documentation - - - - 41-82
	 APPENDIX A - Terms and Abbreviations - - - - - 83

## SECTION 1

Introduction1.1 Background

In February 1975 personnel of the Applications Development Branch of the Data Systems Division prepared and distributed Preliminary Dialog Specification Procedures (Document .11.03.00) as a set of interim standards and procedures to be applied to the ongoing dialog development by subsystem designers. In April of 1975 Bolt Beranek and Newman Inc. was placed under contract to revise and expand this set of specifications into a final set of recommended practices for dialog design to be used in all subsequent dialog development. This document represents the results of this effort.

1.2 Purpose and Scope

The Integrated Systems Project has as its goal the creation of a large-scale interactive data management system that will eventually replace all other ASCS computer operations. The concept of the project is to place computer terminals in various state, county and district service centers, as well as in headquarters. These terminals will be connected to the central computer facility and data base in Kansas City through a telecommunications network. Aside from the data processing operations in Kansas City, the largest volume of activity will be at the terminals distributed throughout the system. Field office personnel will be entering data, calling on the computer for loan, conservation, reconstitution and other computations and making inquiries of the data base concerning the status of particular transactions as well as statistical inquiries summarizing various state and county programs.

Field office personnel will make contact with the computer almost entirely through their local terminal. Thus the image of the system to its primary user population will result largely from the interchange between the terminal and users themselves. This interchange between the computer terminal and its users has come to be called a dialog because it implies initiative on the part of both parties and because, at least in a limited sense, it has the properties of a conversation. If the users like the dialog in which they take part they are likely also to accept the system. If, on the other hand, the dialog is awkward, overly verbose, or projects an image of the users as subservient to the computer, then they are likely to take a hostile view of the system as a whole.

The purpose of this set of specifications is to promote the development of easily interpreted, friendly dialogs by (1) encouraging uniformity of dialogs from one applications package to another, (2) exploiting the full capabilities of the programmable terminals to be installed in the field for creating effective, genuinely interactive dialog, and (3) providing designer aids for dialog creation and documentation. These aids are designed to make the best use of the designer's time and talents. At the same time they meet the requirements for communicating the dialogs that are created to the programmers who must implement them and the managers who must review them.

These specifications contain design recommendations that have implications for the selection of terminal hardware and for the design of terminal software. Together with the dialog documentation that results from their application they will, in addition, be useful to those specialists who will be responsible for preparation and revision of user-training materials and field-office job descriptions.

The development of dialog specification procedures should progress in interaction with the development of detailed specification of the system hardware and software features. At the time this project was undertaken detailed system specification was just beginning and it was necessary to make tentative assumptions, in collaboration with the project monitors, concerning the characteristics of the system that would be built.

Preparation of dialog specification procedures as a planning step prior to the actual implementation of dialogs is without precedent. It has involved many judgements on the part of the contractor that should be evaluated in practice in the specific context of an ASCS operating system before these recommendations are implemented on a more general basis.

For these reasons this should be considered to be a working document. As more detailed system specification is completed and as experience with the use of these procedures accumulates the recommendations contained herein should be periodically reviewed, updated or revised.

### 1.3 Applicability

The specifications herein apply primarily to the interactive dialog that will be implemented at the programmable terminals that include a CRT display surface, local computer memory, a key-

board and a device for producing hard copy of the image on the CRT. They have not been developed to apply to any of the batch processing operations envisioned in the system.

#### 1.4 Terminology

Given in Appendix A in alphabetical order are the set of terms and definitions used in the specifications developed in subsequent sections. The reader unfamiliar with interactive systems should read through them before proceeding. They may also be used as a reference list.

## SECTION 2

General Principles of Effective Dialog Design

Dialogs must be developed with two perspectives in mind. On the one hand they must be compatible with the objectives of the software and hardware constraints of the system and must use computer resources in a cost-effective way. On the other hand they must be compatible with the needs and desires of the terminal users who are the consumers of these computer resources. These two sets of objectives are sometimes in conflict and trade-offs must be evaluated and resolved. In other words many of the goals of good dialog are incompatible with minimizing storage requirements, creating efficient operating programs and minimizing programming effort. While such criteria of efficient systems have been considered repeatedly in reaching the recommendations herein, the larger systems view recognizes that the operator is a fundamental limitation in overall system effectiveness and no matter how automated the system becomes, personnel costs will still be a significant proportion of overall operating expense. Maximizing operator efficiency and minimizing operator error are, in the long run, as important, if not more important, than the usual criteria of good software applied by applications programmers in judging their colleagues' work. It can be expected that a significant proportion of system programming effort will be devoted to providing for an effective "front end" or user interface.

In the paragraphs that follow a set of general principles for the production of good dialogue from the users perspective are presented. Many of the specific recommendations given in later sections are derived from these principles. It is never possible to formulate recommendations to address every possible situation that may arise. Therefore these principles may also be useful when the designer reaches a point where the recommendations fail to provide a solution to a specific development problem.

### 2.1 Know the User Population

The development of user-oriented systems implies that the designer knows who the users are -- their background and knowledge, their level of training, the turnover in their jobs; and level of computer expertise. One of the significant challenges to ASCS system designers is to create dialogs for users who have

little or no understanding of computers but who are highly knowledgeable with respect to programs of ASCS. The county field offices vary from two-person operations to large installations. Work assignments imply that in some offices the county executive director will be interacting with this system himself while in others he will only supervise operating clerks. Thus the user population is diverse in background but specialized in job knowledge and the system needs to be responsive to this diversity.

## 2.2 Respond Consistently and Clearly

Even though the dialog is driven by the computer, it is desirable to make the user feel that he is in control of the system by always providing a sensible next step at every point. Dialog frames should never dead-end with no further action available to the user. If he has provided an entry that is inappropriate to a particular subsystem he should be so informed via an error message, perhaps with a paragraph reference to a manual to clarify the error, and he should be given explicit options concerning what to do next. The designer should anticipate the possible states in which the user can find himself and provide options at each point.

Create error messages and other instructions to the user that are simple, direct and unambiguous. Avoid reference to the system as a person. Avoid entertaining references. They may be friendly and humorous the first few times but after many repetitions they become tedious and aversive. Acceptable: "Loan search produced no output, check loan #." Poor: "Try again, I cannot find that loan in my files."

When reporting errors or misinformation use wording that does not imply fault on the part of the user. Offer information which may be useful for correcting the error. Acceptable: "Loan number should have four characters." Poor: "You entered the wrong number of characters."

Error messages may be prepared at several different levels of specificity, for example:

"Search for this entry produced no output."  
"Search for this loan produced no output."  
"Search for Loan #3723 produced no output."

Greater specificity implies more programming effort and preparation of a greater variety of messages. However this kind of specificity contributes to the friendliness of the system and should be employed wherever practical.

A critical ingredient of friendliness is consistency. Once the user has learned a procedure or set of simple rules, he or she has the right to expect that they will always work in every context that is perceived to be similar to the context in which it was first encountered. One of the primary goals of the standard procedures described here is to promote consistency.

### 2.3 Carry Forward a Representation of the User's Knowledge Base

As a user undertakes a transaction, he accumulates and enters information relevant to that transaction. Whenever possible and useful the system should remember the choices already made and data already entered so that at a later point in the transaction he is not required to make that choice or enter those data again.

In some cases user choices or data entry have logical implications beyond that made explicit by the choice itself. These implications should be recognized by the application designer and used to reduce the demands placed on the user.

Example: Suppose a user wishes to complete a repayment for selected bins from a particular loan that is outstanding. Specification of the loan number should retrieve information from the data base about the specific seal numbers or warehouse receipts included in that loan. This should be displayed so that the user does not need to enter this information from the keyboard.

### 2.4 Adapt Wordiness to User Needs

Since the user population is assumed to be familiar with ASCS procedures and the same dialogs will be used repeatedly by the same individuals, relatively brief messages and the use of standard ASCS terminology and abbreviations is recommended. Brevity should not compromise clarity however. If a few extra words will prevent ambiguity they should be used.

## 2.5 Provide the Users with Every Opportunity to Correct Their Own Errors

One of the most significant advantages of interactive systems over batch processing systems is the ability to provide immediate feedback to the user about the data that has been entered and the integrity and consistency of new data compared with information already in the data base. This advantage should be exploited in every way possible to minimize errors. It is rewarding to the user and, at the same time, contributes directly to system effectiveness.

## 2.6 Promote the Personal Worth of the Individual User

The dialog should respect the intelligence and capabilities of the users and not reduce them to the level of data coders. A relevant concept for this purpose is that of "open-ended design." The system may provide a minimum task requirement that all users can accomplish, but leaves open the opportunity for the more knowledgeable user to go beyond this minimum. It provides the capabilities to permit the user to take initiative in solving problem cases, seeking further source data, resolving conflicts between data base information and new source data, utilizing knowledge of code structures and categories to initiate inquiries beyond the capabilities expected during routine activity.

Example: The naive user may examine each menu in turn to thread his way through a branching structure to reach a desired transaction. The experienced user may use the menu bypass feature (see paragraph 4.7.3) to move directly to the desired transaction frame, saving time and effort.

Example: The inexperienced user may need to refer back to a producer or county office file to resolve an error in producer ID. On the other hand, a knowledgeable user may place the current transaction in suspense, and enter a different subsystem by farm number to seek a list of names and ID's for producers associated with the farm in question. In this way, the proper ID may be identified.

## SECTION 3

Interactive Terminal System Characteristics

As stated in paragraph 1.2, the dialog forms and characteristics that are appropriate to the ASCS system depend ultimately on the assumptions one can make about the characteristics of the hardware and software system on which the dialog will be implemented. Given below are a set of assumptions about these characteristics that form the basis for subsequent dialog specifications.

This set was developed through the joint efforts of the project monitors and contractor personnel who considered the environment in which the ASCS system must operate, realistic limits on obtainable hardware, software and telecommunications capability and human factors criteria for satisfactory interactive dialog.

3.1 System Response Time

Response time from key press to character display when utilizing software within the terminal itself will be virtually instantaneous (less than 0.1 sec).

Response time when access to the communications concentrator is required will be less than 5 sec. at least 80% of the time.

Response time when access to the central computer data base is required will be less than 15 sec.

3.2 Terminal System Memory and Processing

There will be sufficient rapid-access memory in the terminal to maintain approximately two CRT pages plus associated terminal software. Terminals will have a programmable general-purpose microprocessor.

3.3 Terminal Capabilities

3.3.1 1920 display positions: 80 characters per line by 24 lines.

3.3.2 Split screen capability: Programmable brightness and partial screen transmit functions which will permit a distinction between background fixed-fields and foreground variable-fields. Background fields may be protected and made transparent to keyboard control.

3.3.3 Addressable cursor control: Cursor may be positioned at a location determined by an application program or by the terminal software.

3.3.4 Video inversion capability: White characters on black background may be changed, for any field, under program control, to black characters on white background.

3.3.5 Keyboard cursor control: Cursor may be moved backwards-forwards one character; up-down one line from keyboard. When protected fields are used, cursor will jump to next or last previous unprotected character position.

3.3.6 Character replace directly from keyboard: If the cursor is positioned under a given unprotected character and a new character is typed, the new character replaces the old one directly. Typing a space under these conditions simply advances the cursor one character space.

3.3.7 Character delete: When the cursor has been positioned under a particular character and the delete key is pressed, one character is deleted and the cursor is advanced one character space.

3.3.8 Clear screen and foreground: Screen and foreground may be cleared under program control.

3.3.9 Enter key: A key that takes variable field data at terminal and transmits them to the applications program. It is normally pressed to complete data entry for one input frame.

3.3.10 Next page key: Advances dialog control program to next display page or frame and processes [or stores] response from previous page.

3.3.11 Last page key: Instructs the dialog control program to return to the last previous page. As long as there is local storage available the foreground information contained in the current page is retained. As soon as any action is taken with respect to the newly displayed page (i.e., the previous page) that requires using residual terminal storage capacity, then the current frame foreground data are deleted. The cursor is reset to the beginning of the first variable field in the previous frame.

3.3.12 Print key: Produces hard copy of any fixed and variable field data currently being displayed.

3.3.13 Tab key: In the event that there is more than one variable field in a single frame, this command advances cursor to the first position of the next variable field. If no variable fields are left beyond current cursor position then it performs next page function. At the time of Tab, the variable field just entered is subjected to local field editing constraints. If an error exists, the error message is printed at the bottom of the display, the audible signal is sounded and the cursor is reset to the beginning of that field instead of advancing to the next field. Use of Tab when no data have been entered will be interpreted by the terminal software as a null entry.

3.3.14 Default key: Where possible and appropriate the designer will specify the single most likely entry in each variable field. Use of the default key will enter and display that default value for the given variable field and set the cursor to the next available character position. If no default value has been specified, use of the default key will sound the audible signal and retain the cursor at the first character of that variable field.

The default key will be most effective if the default values may be stored in the terminal software with the current frame and appear when activated with no delay. If delays longer than 2-3 seconds are encountered on the average, as is likely if default values are stored at the concentrator, the usefulness of this key is reduced.

The default key may also be used to branch to a frequently selected alternative frame. In this case, the function of pressing the default key must be explicitly described on the display screen.

3.3.15 Help key: At the discretion of the designer help messages may be provided for any variable field entry that might be either confusing or obscure, or where, as a result of system use, it is found that many erroneous entries have occurred. The help key prints an explanatory message in the message field but leaves the cursor in its current position. Help messages may describe alternative acceptable entries for the variable field in question, or may provide a paragraph reference to a user manual that explains acceptable entries for that field. If no help message has been provided, this command simply sounds the audible signal and leaves the program in its current state. Five-second delays to obtain help messages are acceptable.

3.3.16 Escape key: This key transfers control back to the beginning of the currently active subsystem and displays the first frame of that subsystem. If no subsystem is active, or if the escape key is pressed a second time, it transfers control back to a point immediately after log-in and displays the starting menu. Any uncompleted transactions in process at the time are lost.

3.3.17 Suspense files: It has been found useful to introduce the concept of suspense files in subsystem design. A suspense file is a temporary storage location in the data base in which input data that are, for one reason or another, incomplete may be held "in suspense" pending completion.

Creation of suspense files adds to the memory demands on the central data base and requires introduction of software and user activity to periodically review and eliminate outdated files. However, the concept can make an important contribution to user effectiveness by reducing significantly the need to reenter valid data. It is recommended that the ASCS system provide for suspense files and their use is illustrated in the sample dialogs in Section 7.

## SECTION 4

Recommended Practices for the Design of Dialog

A man-computer dialog is a structured conversation which is formally controlled by the computer or computer terminal software but which gives the user the perspective that he is in control. The user must be able to (1) select the task he wants to do; (2) enter directives to the computer; (3) enter information to create or change data-base files; (4) view information provided by the computer; (5) obtain executable instructions in the event the operator requires help or makes errors. Item (1) employs menu selection; (2) and (3) use input frames and (4) and (5) are displayed on an output frame. However, single frames having a mixture of these frame types are permitted.

4.1 Menu Selection

When a sequence of choices are used to identify applications programs or subsystems to be activated, it is often called menu selection by analogy with the use of a restaurant menu. With this procedure a user is presented with two or more choices and asked to select the one from among the set that reflects what he wants to do. Usually one selection causes the dialog to branch to a further set of alternatives. Similar choices are made until the last choice has narrowed the overall options available down to one. The dialog then continues by presenting the first active frame of that program or subsystem.

4.2 Input Frames

The purpose of an input frame is to provide a structured context for the entry of data. In some cases the format will be modelled after a paper form with labelled columns and rows. In other cases, as in a simple inquiry, it may be a simple sequence of requests for different inputs that provide the parameter values for a data-base search. A part of the input frame is presented as underlined spaces with blanks indicating that characters are to be filled in by keyboard entry.

4.3 Output Frames

An output frame in pure form contains no provision for any keyboard entry except the means of advancing to the next frame. Its purpose is (1) to present to the user the information

he or she requested for inspection, (2) for production of a hard copy for transfer onto legal document form in the county office, or (3) simply to acknowledge receipt of an input frame.

#### 4.4 Permissible Frame Combinations

It is permissible to incorporate a menu as a part of an output frame in order to indicate the next options available to the user. This application will be common in the case where the output frame consists of an acknowledgement and the menu indicates a choice of what to do next. Any frame that contains both output and menu items will be defined as a menu frame.

It is also permissible to combine parts of an output frame with an input frame. As an example consider the illustration in Section 7 (Frame CAAL-12). In the case of a loan repayment transaction the agent would initiate an inquiry by loan number. The system would provide data concerning the seal numbers or warehouse receipts associated with that loan as an output frame. The agent would indicate in space provided on that frame which receipts were to be repaid as a new input and the system would then calculate the repayment amount. Thus both retrieved information (the output frame) and new information (input frame) would be combined in a single frame. It is not sensible to segment these two parts because the natural and efficient way to proceed is to operate on the retrieved information directly. Any frame that contains both input and output items will be defined as an input frame.

#### 4.5 General Recommendations for Frame Specification

4.5.1 Left justification: All frame lines, except where indicated below, should be left justified.

4.5.2 Frame title: Every frame must contain a title or header. The header is a brief label that describes the purpose of the frame. The header should be separated by one blank line from the remainder of the frame text.

4.5.3 Message window: The last four lines of every page are reserved for presentation of messages. Message types may include indications of errors, communication links or system status messages.

4.5.4 Frame references: Every non-menu frame should contain an abbreviated reference to an operating manual and paragraph number indicating the source of an explanation of the procedures involved in that frame. This reference should be on the same line as the header but separated from it by 5 blank spaces. These references will be subject to change control mechanisms just as for other aspects of the system documentation.

4.5.5 Frame page numbering: When an output frame contains more than one page, the notation, "page \_\_\_ of \_\_\_" should appear right-justified on line 20. When, on an input frame, table entries require more than one page to be used, the notation should simply be "page \_\_\_."

4.5.6 Frame wording: The importance of using carefully thought out wording for all frame requests, column and row headers and field labels should be emphasized. Do not create new jargon. Only abbreviations contained in ASD approved list of abbreviations and codes or terms specifically approved by the ISP Project Manager or his designee may be used. Pretest all wording on a sample of potentially qualified users.

4.5.7 Frame layout: If the full 20 lines are not required then use care in layout to produce a balanced, uncluttered screen.

#### 4.6 Recommendations for Display of Menu Frames

4.6.1 Menu definition: Each menu frame presents a set of numbered items for selection and a space for entering the item number to be selected.

4.6.2 Menu numbering: Menu items should be numbered in consecutive order beginning with 1.

4.6.3 Menu choice text: Use the phrase "Enter choice \_\_\_\_\_" separated from the menu items by at least one blank line. A sample menu frame is shown in Fig. 4.1.

4.6.4 Menu layout: The item numbers are left-justified and presented on single lines. If the menu items are brief and if, in the judgment of the designer, it is logically appropriate, menu items may be arranged in two separate columns. In this case the second column should be justified at a character location that provides a balanced screen.

4.6.5 Menu length: Each menu frame presents an opportunity for only one selection. If the menu list exceeds 10-15 items, then the designer should consider reorganizing the list into two separate menu frames, maintaining the logical organization within the hierarchy. Multi-page menus are prohibited.

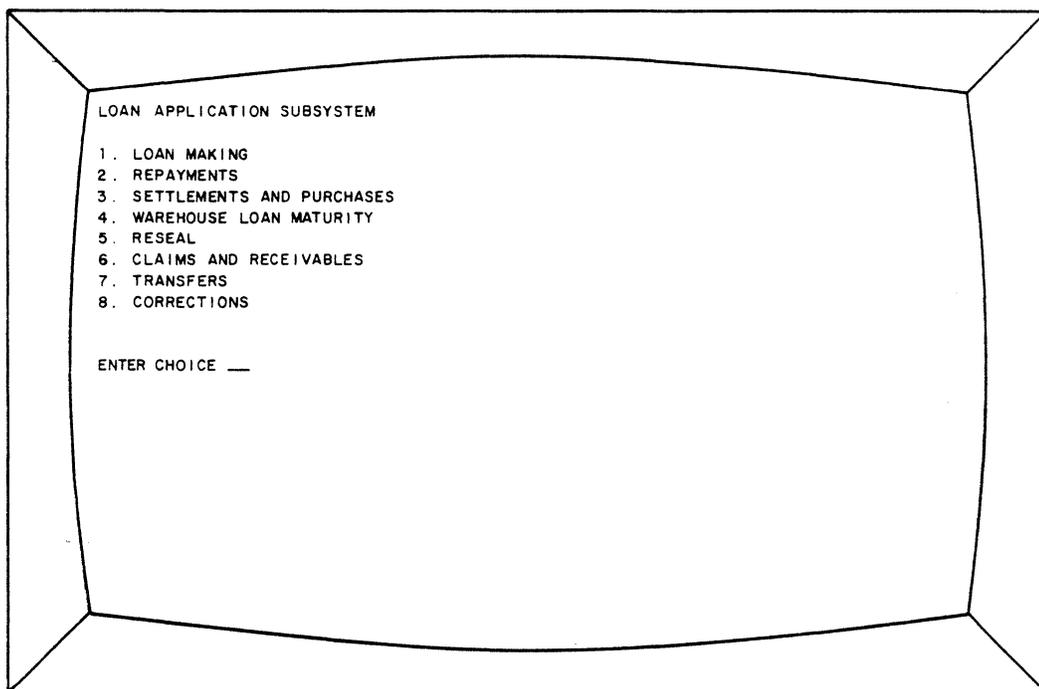


FIG 4-1 Menu Selection Frame

4.6.6 Menu structure: Menu items should be ordered in the list on the basis of the logical structure of the list. If the list has no logical structure then items should be ordered according to a ranking of their expected frequency of use, with the most frequent placed first. If the list contains logical subunits, these subunits should then be ranked in expected frequency of use and ordered accordingly.

4.6.7 Menu sequence: Menu frames should be sequenced in an order dictated by the logical flow of the user's analysis of the transaction. In some cases this will mean holding choices in store within a transaction until the choice is relevant to later menu branching or to selection of an input or output frame.

4.6.8 Menu options: Menu items should not be worded to imply a yes or no answer. Such items should be phrased so that either alternative requires a positive choice and item selection is always by number.

#### 4.7 Menu Frame Control

4.7.1 Cursor positioning: Upon presentation of each menu frame the cursor is positioned at the first character of the choice entry line (first unprotected field).

4.7.2 Sequence control: The number entered at the end of each menu frame designates the next frame to be selected. The user types in the number and presses the "next" or "tab" key.

4.7.3 Menu bypass: At any point in a sequence of menus, it is permissible to type the selection, followed by a comma followed by another number and so forth up to the limit of the number of remaining menu frames in a sequence. The last number is followed by pressing the "next" key or "tab." This set of actions signifies that the user is selecting menu choices beyond the current frame.

This procedure serves to short circuit a tedious and highly overlearned menu-frame sequence. With time, the user learns, for example, that a reconstitution may be initiated by the sequence 3, 1, 2, 4. If he has not yet learned this complete sequence he may type 3, 1; view the next menu, and continue from there. If too many numbers are typed, the system carries the sequence forward until an input or output frame is reached and then presents the remaining unused numbers in the message field with the comment, "end of menu sequence. x, y ... ignored."

#### 4.8 Recommendations for Display of Input Frames

4.8.1 Input frame definition: An input frame consists of fixed background fields and variable foreground fields. The fixed fields are protected when they appear. They cannot be changed or edited. They always contain the same information and the cursor skips over them when space or tab is executed. The variable fields appear as underlined blanks to be filled in by the user. As the user types, the spaces are filled.

4.8.2 Input frame structure: Input frames may be structured in several ways. If the sequence of information is non-repetitive then provision should be made to enter a single data field on each line. The description is placed to the left of the field and a set of blank spaces with underlining are provided to indicate the maximum length of the field to be keyed. A sample of this kind of input frame is shown in Fig. 4-2.

4.8.3 Input frame format: If the information to be entered is non-repetitive but is to be transferred directly from a hard copy form then it is desirable to reproduce the sequence and arrangement of variable fields to correspond to the sequence and arrangement of fields on the original form. However, when adapting the input frame, the designer should examine each hard copy form to ensure that it is designed in a format that is compatible with the needs of the user organization.

4.8.4 Data entry tables: If provision must be made to enter a table of data, that is, a sequence of different items to be entered row by row, then the input frame may be set up as a table with column and row headings. When it first appears there is only one line provided for data entry with input field underlining indicated. At the end of the first row the user presses either "tab" or "next". If "next" is typed the table is accepted and the cursor moves to the next field or page. If tab is typed, the first row remains, the cursor jumps to the first character of the first entry in the next row, the input field underlining appears for this row, and a second row may be entered. This process may be repeated any number of times until the last entered row corresponds to row 20. Pressing the tab key at this point places this page in temporary storage and produces a new page having the same table headings and variable field definitions. This page will be labelled page 2. Two samples of such an input-frame in different stages of data entry are shown in Fig. 4-3.

FARM STORED GRAIN LOANS CCC-666, P2

PRODUCER ID \_\_\_\_\_

NAME AND ADDRESS \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

STATE AND CO. CODES \_\_\_\_\_

LOAN NO. \_\_\_\_\_

FARM NO. \_\_\_\_\_

COMMODITY \_\_\_\_\_

CROP YEAR \_\_\_\_\_

PRESS "ENTER" TO RETRIEVE LOAN RECORD

FIG 4-2 Input Frame Ready for Data Input

FARM STORED GRAIN LOANS      CCC-666

LOT/SEAL NO.	QUANTITY IN BIN	QUANTITY FOR LOAN
C 73 106 2	655	589

PRESS "ENTER" TO START LOAN CALCULATION

FARM STORED GRAIN LOANS      CCC-666

LOT/SEAL NO.	QUANTITY IN BIN	QUANTITY FOR LOAN
C 73 106 2	655	589
C 73 109 1	3278	2950
C 73 118 2	2132	1921
C 73 120 2	1654	1488

PRESS "ENTER" TO START LOAN CALCULATION

FIG 4-3      Input Frame Illustrating Tabular  
Data Input

4.8.5 Input field sequence: The order in which variable fields are presented may be determined by the order in the source document from which they will be obtained, by the logical sequence in which the user can expect to think of them, or by placing all required fields prior to all optional fields.

4.8.6 Optional fields: Fields that are not required or expected to be relevant for every transaction should be distinguished by placing parenthesis around the protected field descriptor that designates that field. On optional fields no checking will be done to ensure their presence.

4.8.7 Use of \*: Failure to provide an entry in a required field will produce an error message. Therefore, if it is desired to omit temporarily a required field that is to be typed in later, a \* should be typed in the first variable field character position. The \* will be accepted as a valid entry in some programs in which suspense files are used. At the time when the data are to be entered into the data base the \* will be flagged with an error message.

4.8.8 Use of field delimiters: Field separators such as / or -, when used universally, should be included as protected background fields in the frame and only underlining for actual variables to be entered need to be shown. Such delimiters do not force the definition of separate fields between each pair of delimiters, however.

4.8.9 Date format: Dates should be entered in six digit code with field separators built into the protected background fields: 04/27/75.

4.8.10 Monetary field format: Dollar and cents amount fields will be entered left-justified. However when amount fields are returned to the display by an application program they will be justified with respect to a fixed decimal point. Tables presenting columns containing whole dollar amounts exclusively need not include zeros in the cents columns.

4.8.11 Unit of measure: Dollar signs and other amount unit labels, such as bushels or pounds when used universally will be provided in the protected fields and will not need to be entered by the user. However, if more than one unit of measure is acceptable for a given field entry then space should be provided for the user to enter the unit descriptor and that entry should be suitably interpreted by the applications program.

#### 4.9 Input Frame Control

4.9.1 Cursor positioning: Upon initial appearance of each input frame the cursor will be positioned to the first character position of the first input field. When an input field entry is complete and a tab is pressed, the cursor jumps to the first position in the next input field, searching sequentially from left to right within a line and down one line at a time in order to find it. When the last input field of a given frame is completed, pressing tab, next, or enter will cause the screen to be erased. When the application program has completed its operations on those data, then the next appropriate frame will appear. It may be a new menu, a new input frame or an output frame. The appearance of a new frame constitutes an acknowledgement that the desired action has actually been taken by the program. If for any reason the transaction or subaction cannot be taken, the input frame returns together with one or more error messages.

4.9.2 Action prompting: Whenever an input frame is completed and a substantive action is to be performed by the central computer as a result, the user should be prompted as to the specific function that will be performed when he presses the "enter" key. For example, "Press enter to start loan calculation." or "Press enter to list outstanding loans."

#### 4.10 Recommendations for Display of Output Frames

4.10.1 Terminal output frame definition: Computer output will be produced either by a line-printer or other batch output device or through the terminal. Terminal output frames will be used to present computer output when the output is (1) of momentary interest only as in the case of acknowledgements; (2) represents the results of an intermediate calculation that is to be used to produce later results or decisions; or (3) is of longer term interest but can be presented easily within a few pages with the option to produce a local hard copy if necessary.

4.10.2 Hard copy needs: It should be emphasized that most data base information will be available on an inquiry basis at the terminal. The production of hard copy in the field offices should be limited to that which is required for legal documentation or for complex output that does not lend itself to retrieval on an inquiry basis. As the system becomes more sophisticated and as users accumulate confidence in the ability to obtain data when needed by inquiry, the demand for hard copy should decline.

4.10.3 Field description: Output frames will contain fixed and variable fields in which the fixed fields represent standard forms and the variable fields provide for data entry from the data base. The user cannot interact with an output frame.

4.10.4 Output frame page numbering: If an output frame is longer than one page then the notation page \_\_\_ of \_\_\_ should appear on each such page right-justified on line 20.

4.10.5 Output frame forms: Output frames may present a sequence of information, such as the result of an inquiry. Each line should represent a single item of information and each item should be presented left-justified except where it makes logical sense to arrange in columns some numeric data. Such numeric data should be right-justified. When all the information items are short it may be permissible to use a second column. In this case care should be used to balance the columns in the frame.

4.10.6 Output frame tables: Occasionally it will be suitable to present tables of data. In this case the number of tabular lines should be variable depending on the information in the data base. Provision for the maximum numbers of lines within a page, or across pages if necessary, should be made. Numeric data or dollar amounts should be right-justified in each column.

4.10.7 Output frame layout: Sometimes output frames will be used as the source document to type up a legal document form to be signed and retained in the field office. In this case the output frame layout should correspond closely to the sequencing of items in the typed form. However redesign of the form itself should be considered in creating an optimum format for both the output frame and the paper form. It seems likely that at some future time it will be possible to prepare the hard copy paper forms directly by computer printout.

#### 4.11 Output Frame Control

Unless an output frame also contains a menu, the last line of each page of the frame will provide the instruction, "press 'next' to continue." This line normally will be separated from the body of the frame by two blank lines. If the body of the output frame extends to line 18 then a single blank line is permissible with the continue instruction placed on line 20. The cursor will appear in the lower right hand corner of the display, but will have no function and will not be able to be moved by the user.

#### 4.12 Recommendations for Error Messages

4.12.1 Properties of error messages: All error messages should communicate three elements of the error: (1) where the error occurred; (2) what error occurred; and (3) one or more ways to recover from the error or where to find out how to recover from it. The location of the error is reported by placing the cursor at the first character position in the field at which an error was detected and causing a video inversion of the field, that is, the white characters and black background will become black characters on a white background. The nature of the error and the recovery procedure are reported in an error message using lines 21-24 on the display screen.

4.12.2 Error message content: Error messages are regarded as a form of output frame. They should be clear, concise, and specific. Where possible, they should also be standardized. The message itself may not provide a menu or form-filling entry.

If the user is unable to recover from any error he may replace the first character with an asterisk and delete the remaining characters in the field using the character delete key. When the frame is entered, this action indicates to the program that a correct entry will be supplied later and will be accepted for entry into a suspense file. Alternatively the transaction may be aborted by pressing the escape key and returning to the beginning.

4.12.3 Error message layout: More than one error message may be supplied for a single page or frame. No error message may be longer than two 80-character lines. If the four lines reserved for messages are insufficient, then the terminal software will supply the words "x errors detected" right-justified on line 24. The symbol x refers to the total number of errors on the currently displayed page. In addition, all fields found to be in error will be video inverted until they are modified by the user. If there are unseen messages because of lack of space, they move up into visible position as the earlier ones are dealt with. If new terminal-detected errors are generated in the attempt to correct old ones, the most recent error message moves to the top of the list and the others move down. The cursor position is always related to the first offending character of the field referred to by the top remaining error message. Partial messages that would fit on the bottom of the screen are not displayed until there is space for the full message. This procedure for handling multiple error messages is designed primarily to address extreme cases. In general, error messages will not exceed one 80-character line and only in rare instances would more than four messages be detected by the application program for a single page.

4.12.4 Terminal software error checks: The variable definition forms provide a place to define whether a variable is all alphabetic, alphanumeric (including a mixture of alphabetic characters and digits), or it is a number. For these simple errors the recovery procedure is to recheck the entry. This information is implied and it is not necessary to restate it in each message.

4.12.4.1 Check for alphabetic characters: If the entry must be alphabetic it can be checked for stray digits or non-alphabetic codes. The appropriate error message is:

"Alphabetic characters required."

4.12.4.2 Check for number and number range: If it is a variable length number for which an acceptable range can be defined then it can be checked for stray alphabetic or non-numeric characters. The appropriate error message is:

"Number required."

If a number falls outside the defined range then the appropriate error message is:

"Number should be between x and y."

4.12.4.3 Check for number of characters: In a fixed length field if an incorrect number of characters is entered the error message is:

"X characters required."

4.12.4.4 Check for required fields: The terminal will also check for the presence of variables required to complete the transaction. If entry is required and no entry is given, then the appropriate error message is:

"This field required."

4.12.4.5 Invalid keys: At any point in a transaction certain function key responses may be expected and planned for while others are invalid. Pressing a key that is invalid at a particular point causes no action and calls forth an error message indicating

which keys were expected at that point. For example, at the end of a frame there is a statement reporting expected responses, such as "Press "Enter" to compute loan. Press "Default" to create suspense file." If the user presses any other key at this point in the transaction, it will produce the error message:

Press "Enter" or "Default".

4.12.4.6 Timing of terminal error checks: For terminal-detected errors, each error is reported when tab or next is pushed to move to the next field. Thus only errors referring to a single field may appear at one time. When the field has been corrected, whatever action was appropriate to terminate that field originally is still appropriate.

4.12.5 Applications program error checks and entry validation: There will be many cases in which error correction may be based on knowledge contained in the application program: Producer ID may be compared with producer name; certain farm programs may no longer exist; file searches may be incorrectly specified, and so forth. These error checks occur at the end of an input frame, when an attempt is made to "enter" the data. No specific wording for such errors can be specified for all cases in advance, but given below and in Section 2.2 are some principles to be followed. The capability to detect and correct errors of this type is one of the most important and useful features of an interactive system and should be exploited to reduce the incidence of incorrect data reaching the data base.

4.12.5.1 Error message specificity: Make the error reference as specific as possible, for example, "Search for loan #6342 produced no output" is preferred to "Search for inquiry produced no output."

4.12.5.2 Error message content: In specifying what remedial action to take, it is permissible to refer to a specific paragraph of a reference manual; to provide a direct instruction such as, "use warehouse stored loan procedure" or a more general instruction such as, "recheck loan number."

4.12.6 Error message acknowledgement: All error messages require an acknowledgement. When the user introduces any change in the input field in question and/or presses the tab key the field will change from black on white to white on black. The further response of the system depends on several conditions. If the error was detected by the terminal software, pressing tab

will either advance the cursor to the next input field or, in the case the error has not been suitably corrected, it will return the cursor to the beginning of the field in question.

If the error was detected by the application program, then correcting the field and pressing tab will advance the cursor to the next input field. The top error message is tied to the position of the cursor. If the cursor is positioned at a field in error, the highest error message will apply to that field. As the cursor is moved to other fields, that error message will remain until the cursor points to another field in error. At that time the first error message will disappear and the one appropriate to the new field will move up, followed by subsequent error messages. Thus, the user is able to examine all the error messages in order to diagnose the problem before he takes any action. He also is able to change only one or two fields if he feels that they will correct other errors.

A positive action is always required by the user to reenter a frame on which errors have been corrected. In the case of errors detected by the applications program, further validation of corrected errors is not accomplished until the frame is reentered into the system. Thus, if an error correction is found also to be in error, the error reporting procedure will be repeated just as it was in originally identifying the error. The only exception to this protocol occurs if an attempt to correct an error detected by the applications program produces a further error that fails to meet the requirements of the terminal error checking routines. In this case the errors will be reported field by field as described above and in paragraph 4.12.3.

## SECTION 5

Steps in Systematic Dialog Development5.1 Functional Task Analysis and Flow Chart of Transaction from User's Perspective

Prior to the development of dialog, relatively detailed system requirements are prepared. System requirements are a necessary input, but they are not sufficient to establish dialog requirements. What is needed is an analysis of the task requirements associated with each transaction from the user's perspective. The task analysis breaks each transaction into a logical set of component tasks and specifies the sequence in which those tasks must be completed.

In formulating the task analysis the designers should be sensitive to (1) the functional tasks to be performed; (2) the information needs and the source of that information for each task and (3) the output information requirements and output destination of each item.

If the transaction is to conduct an inquiry of the data base, then the tasks include (1) selection of the frame for specifying the parameters of the inquiry; (2) entering the parameters from the keyboard, making note of where the user is to acquire the information needed to select the proper parameters; and (3) examining the output report of the results and making use of it for the intended purpose of the inquiry. As illustrated in Figure 7.1 in Section 7 a task flow chart is similar to the flow chart of a computer program, but the unit of analysis is different. Some aspects of a task flow chart will have no corresponding aspect in the dialog control program that will be written to implement it while some parts of the dialog control program will not be represented in the task analysis. A logical unit to a user is not necessarily the same as the logical unit of the program used to create that piece of transaction.

In many cases the starting point for the task analysis of a computerized transaction may be the transaction as it is currently performed manually. It must be emphasized that this is only the starting point. The designer should ask himself, whether, given the computer capability, it would not be better to reorganize the transaction to minimize the data input requirements, maximize the responsiveness of the system or improve the logical flow. Logically, the user-requirements task analysis should precede the development of any application program structure, but in practice the two are likely to progress in parallel.

## 5.2 Flow Chart of Required Dialog Frame by Frame

The next step after creating a flow chart of the logical steps in a transaction is to expand this flow chart into specification of all dialog frames that will be required to implement this flow chart. It is helpful to begin this process by creating an index card for each proposed frame that identifies the information to be included in each frame so that these cards may be arranged and rearranged as progress is made toward the final frame requirements. Menu selection must be broken down into logical units. Input frames must be identified, keeping in mind the source of the data to be entered. Output frames must be identified keeping in mind the use to which this output information will be applied.

It may be necessary to review the task flow chart together with the frame flow chart and to iterate these plans to ensure compatibility and effectiveness of the conduct of the overall transaction. It is in the creation of the overall structure and implementation of the frame by frame flow chart that the greatest challenges to the talents of the dialog designer are called on. Small improvements in structure at this point can produce order of magnitude changes in the success of the final design from the users perspective.

## 5.3 Specification and Layout of Dialog Frames

Once the frame structure has been defined, effort should be focussed on implementing each frame so that each item within the frame communicates its purpose and the procedures used to complete it as clearly and directly as possible. Careful attention must be paid to the choice of specific wording of each frame. Frame layout must be specified that is consistent with the standards outlined in this document.

## 5.4 Specification of User Supplied Variable Fields

Concurrent with the specification of frames it is necessary to describe the characteristics of each variable field to be entered by the user. The focus of this description is on those aspects of each field that may be subject to validation, error checking and a variety of user aids to filling in each field. See Section 6.2 for a full description.

### 5.5 Specification of Possible Error Messages Associated with Each Field

Local terminal software checks are prescribed by the conditions checked for each variable, but those validation checks to be performed by the applications program must be described and appropriate error messages defined for each case that can be anticipated. Follow the instructions for error message creation given in Sec. 4.12 and for the chart completion given in Section 6.3.

### 5.6 Documentation

The following items of documentation represent the minimum required set for each subsystem and are subject to Configuration Management Control.

1. Transaction Task Flow Chart
2. Dialog Frame Flow Chart
3. Complete set of Frame Specifications
4. Complete set of Variable Definitions
5. Complete set of Error Message Specifications

## SECTION 6

Dialog Documentation Aids and Instructions for Their Use6.1 Dialog Frame Specification Form - Fig. 6-1

This form provides a complete specification of the content and layout of each page of each frame. One should be prepared for each page of each frame that is to be implemented in each subsystem.

1. Enter the frame number. This is a unique number assigned to each frame by the DSFO, DBMS.
2. Enter the page number of the frame. Multi-page frames will be numbered sequentially beginning with "1".
3. Enter the subsystem name.
4. Enter the date that the frame specification is prepared (MM/DD/YY).
5. Check whether the frame specified is an input, output, or menu frame. Any frame which is a combination of input and output should be considered an input frame. Any frame containing a menu and an output frame should be considered a menu.
6. Enter the text of the frame as it will appear to the user. Each character must be specified by its line number and character position on that line. For input frames, represent the variable data that has to be keyed in by the operator by an underline in that character position. For output frames, indicate characters that will be supplied by the computer by placing one of three characters in the corresponding character positions. Use an "A" if the variable supplied will be all alphabetic, a "9" if the variable will be entirely numeric, and an "X" if it will be alphanumeric.
7. Enter and explain anything ambiguous, unclear or unusual about the frame, how it works or the variables it contains; for tables indicate the maximum number of lines that might be needed when the frame is used. Enter comments about it in relation to the entire sequence of frames. Give cross references to other documentation relating to that frame. Ideally every frame should have at least one comment.



6.2 Variable Definition Form - Fig. 6-2

The purpose of this form is to provide the designer and programmer with a complete description of all back-up supporting the definition of each variable. Variables associated with each new page of dialog should be described on a separate variable specification form. Since output frame variables are not subject to checking, or modification, this form is only required for input frame variables to be entered at the keyboard.

1. Enter the frame number of the variables to be defined.
2. Enter the page number and the total number of pages in the frame.
3. Enter the name of the subsystem.
4. Enter the date the variable definition form is prepared (MM/DD/YY).
5. Enter a sequential number for each variable being defined. Begin numbering the variables as they are encountered from left to right starting with line one; for example, line one variable one is numbered "1", line one variable 2 is numbered "2", etc. If a series of entries of the same item is specified (e.g., a series of loan numbers), don't assign each one a different variable number. If the only difference between them is position on the screen, then assign one variable number and give a range of line numbers and character positions. If there are other differences between them (such as the first one being mandatory and the others not), assign different variable numbers.
6. Enter the line number or range of line numbers on which the variable is located.
7. Enter the character position in which the variable starts.
8. Enter the maximum number of characters that the variable contains. In counting characters do not include field delimiters that are part of the protected background. For example, the date 06/12/75 would be 6 characters.

9. Enter "X" under "ALL ALPHA" if the variable is to be checked to insure that only alphabetic characters are present. This also includes special characters such as "\*", or "-".  
Enter "X" under "ALL NUMERIC" if the variable is to be checked to insure that only numbers are present.  
Enter "X" under "MIXED" if both alphabetic and numeric characters are allowable.
10. Enter the range of allowable values for the variable. The following operators may be used: less than (<), greater than (>), and equal to (=). The boolean operators "AND" and "OR" may also be used.
11. If the variable is required to contain data enter "X" in the "Y" column. If it may be omitted enter "X" in the "N" column.
12. Enter the value which the variable should assume if the terminal operator presses "default." This should usually be the first item in a menu or the most common keyed-in values. If there is no logical value leave this column blank.
13. Write the message or a reference to the message which would appear if the terminal operator pressed the help key instead of entering the variable.
14. Enter a title for the variable or an explanation of it. Also include any other relevant information about the variable and its interactions with other variables or other frames.
15. Enter a title or identifying label for the frame. Also describe any interactions between variables within that frame or with variables in other frames. Explain any ambiguities or unusual characteristics of the variables.

VARIABLE DEFINITION

FRAME NO.: ① PAGE: ② OF ② DATE: ④

SUBSYSTEM NAME: ③

VAR. NO.	LINE NO.	STARTING CHAR. POSITION	VARIABLE EDIT CRITERIA						DEFAULT VALUE	HELP MESSAGE	COMMENTS			
			NO. OF CHAR.	TYPE			NUMERIC RANGE OR PERMITTED CHARACTER	MANDATORY						
				ALL ALPHA	ALL NUMERIC	MIXED		Y				N		
⑤	⑥	⑦	⑧		⑨			⑩	⑪		⑫		⑬	⑭
COMMENTS: ⑮													DESIGNED BY:	

6.3 Error Message Specification Form - Fig. 6-3

This form provides for documentation of error messages to be supplied in conjunction with terminal generated errors that are detected at the level of the applications program. Terminal error checking is routine and need not be documented beyond the level provided in the Variable Specification Form.

1. Enter the number of the variable where the error was detected.
2. Enter the name of or a description of the validation check that was occurring when the error was caught.
3. Enter the error message as it will be displayed at the terminal.

ERROR MESSAGE SPECIFICATION	FRAME NO.:	PAGE:	DATE:
	SUBSYSTEM NAME:		
VARIABLE IN ERROR ①	VALIDATION THAT PRODUCED ERROR ②		
ERROR MESSAGE ③			
_____ _____ _____ _____			

VARIABLE IN ERROR	VALIDATION THAT PRODUCED ERROR
ERROR MESSAGE	
_____ _____ _____ _____	

VARIABLE IN ERROR	VALIDATION THAT PRODUCED ERROR
ERROR MESSAGE	
_____ _____ _____ _____	
COMMENTS:	DESIGNED BY:

## SECTION 7

Sample Dialog for Commodity Loan Repayment Subsystem7.1 Introduction

The material that follows presents a worked example in dialog development that applies the principles and meets the standards described in Section 2-6 of this document. The example is complete with the exception that it provides only some samples of applications program level error checks and messages.

The dialog begins by selecting from menus the commodity repayment subsystem, and sets up a branch depending on whether repayment is to be made directly or through sale of the commodity to a third party. The choice implied in CAAL03 is held in store until after the loan to be repaid has been identified. Next the user is asked whether he already knows the loan number, has entered data relevant to this repayment previously and wishes to work from a suspense file or whether he wishes to obtain the loan number from a search of possible loans associated with a particular producer. Once the loan number is known the program then branches, either to obtain data for preparing a marketing authorization or to provide data for creating a repayment record.

It is expected that there may be a substantial time lapse between the time a Marketing Authorization is completed and when repayment is actually made. Similarly, in the case of direct repayment, a producer may request information for planning repayment that will be implemented at a later date. In both these cases suspense files have been provided to hold the transaction for later implementation.

Completion of a Marketing Authorization provides most of the data needed for the final repayment so when a suspense file has been created, provision is made to bypass the Repayment Record forms and go directly to the last step, completing the accounting associated with repayment. Similarly if a repayment suspense file exists it is not necessary to step through the input and output frames associated with the repayment record itself.

The documentation provided should permit the reader to reconstruct in full detail both the user's view of the dialog and the information needed by the programmer to implement it.

7.2

Transaction Task Flow Chart Commodity Loan Repayment  
Subsystem Codes CAAL

