SANDIA REPORT

SAND91—1575 • UC—705
Unlimited Release
Printed January 1992

Technical Specification for the Sandia Management Restructure Study Team (MRST) Prototype Information System

Timothy R. Wyatt, Ronald C. Hall, Lyle T. Davis,
E. Janet Klamerus, Irene Thurston

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A06
Microfiche copy: A01

SAND91-1575
UC-705
Printed January 1992

TECHNICAL SPECIFICATION FOR THE SANDIA MANAGEMENT RESTRUCTURE STUDY TEAM (MRST) PROTOTYPE INFORMATION SYSTEM

Timothy R. Wyatt
Ronald C. Hall
Lyle T. Davis
E. Janet Klamerus
Irene Thurston
Information Systems Design
Sandia National Laboratories
Albuquerque, New Mexico

ABSTRACT

This document contains implementation details for the Sandia Management Restructure Study Team (MRST) Prototype Information System, which resides on a Sun SPARC II workstation employing the INGRES RDBMS. The INGRES/Windows 4GL application editor was used to define the components of the two user applications which comprise the system. These specifications together with the MRST information model and corresponding database definition constitute the MRST Prototype Information System technical specification and implementation description presented herein. The MRST Prototype Information System represents a completed software product which has been presented to the Management Restructure Study Team to support the management restructuring processes at Sandia National Laboratories.

TECHNICAL SPECIFICATION FOR THE SANDIA
MANAGEMENT RESTRUCTURE STUDY TEAM (MRST)
PROTOTYPE INFORMATION SYSTEM

Table of Contents

TECHNICAL SPECIFICATION FOR THE SANDIA MANAGEMENT RESTRUCTURE STUDY TEAM (MRST) PROTOTYPE INFORMATION SYSTEM

1. Introduction

The purpose of this report is to document the development and implementation of the Sandia Management Restructure Study Team (MRST) Prototype Information System. The system was developed using the relational database management system INGRES/Windows 4GL [1] on a Sun SPARC II workstation running the Version 4.1.1 Sun OS operating system [2]. The implementation is based on Standard Query Language (SQL) queries from a fifth-normal-form data base definition derived through Nijssen Information Analysis Methodology (NIAM) information modeling tools [3]. Data loading was performed using Version 6 INGRES Applications-By-Forms (ABF) [4] on a VAX computer running the Version 5 VMS operating system [5].

The MRST Prototype Information System was designed to support the activities of the Sandia Management Restructure Study Team as they applied to the management restructuring efforts at Sandia National Laboratories. By memorandum dated May 8, 1991, T. C. Cannon, 2850, to John Sharp, 2818, Subject: Data Comparisons Desired for Restructuring Study, the need was identified for an information system to support the analysis and comparison of the structural data of surveyed external companies, the purpose being to determine as much as possible about the benefits and drawbacks of particular structures. Also, for those companies that had restructured recently, information was to be gathered on the previous management structure, the motives for restructuring, and the effects of the restructuring process on several efficiency-related items. The design of the system needed to be conducive to these examination, analysis, and comparison activities.

The system was modeled based on the survey questionnaire used to obtain data from the external companies. This survey requested that the companies provide numerical data, graphical data in the form of organization charts, as well as textual responses. Also, the need was identified for the system to have the capability to index certain text by keywords for easy reference.

Based on these needs, the user system was to have two applications. The primary application would be used for the analysis, comparison, and examination of data items, and the secondary application would allow the user to index text by keywords. Both applications were to reside on a Sun SPARC II workstation. The loading of numerical and textual data was performed using an application developed in Version 6 INGRES Applications-By-Forms on a VAX computer running the Version 5 VMS operating system. Graphical data was read into files using a 300 dots-per-inch (dpi) scanner, and an application was developed using INGRES/Windows 4GL on the Sun SPARC II workstation to insert the filenames into the database. In order for the complete user system to be delivered in a timely manner, a schedule was determined for completion of the various phases of development, and was well kept by all involved.

The purpose of the report is not to be a user's guide or a design specification, but rather to document the technical details pertinent to the implementation of the MRST Prototype Information System. Standard commercial software and development tools were used to carry out the implementation. These tools are listed in the References section of this report and are well-documented by the vendors. In order to best describe the system implementation, the remainder of this report will consist of the information model and database definitions, screen definitions, and specifications for frames and associated procedures. As many of the external companies do not wish for their responses to be released, the actual data is not included in this report.

2. Information Model

NIAM, a formal information modeling method, was used to obtain the optimal database model. The model is based on the survey questionnaire used to obtain data from external companies. This section identifies the various labels (lexical object types, i.e., the data occurrences), and the relationships among them, which altogether comprise the information model. A number of relationships are presented between the entities and other entities, and between the entities and labels. Some objects included in the model are not fully implemented in the MRST Prototype Information System. These are retained for the purpose of extensibility required for follow-up implementations, but are not reflected in database schema derivations for this system.

Following is the table of contents and diagrams for the MRST Prototype Information System information model. The table of contents precedes the model diagram. Each label and entity is included in the table. Relationships are classified as either a bridge type or an idea type. A bridge is a relationship between an entity and a label. It bridges real objects with their representations. An idea expresses a relationship between two entities. The entities are shown as circular shapes made with solid lines, and the labels are shown as circles of dotted lines. The tool used to document the model, PC-IAST [6], refers to entities as "no lots" (nonlexical object types) and labels as "lots" (lexical object types). A "natural language" expression of the information model is also included following the diagram.

"

"

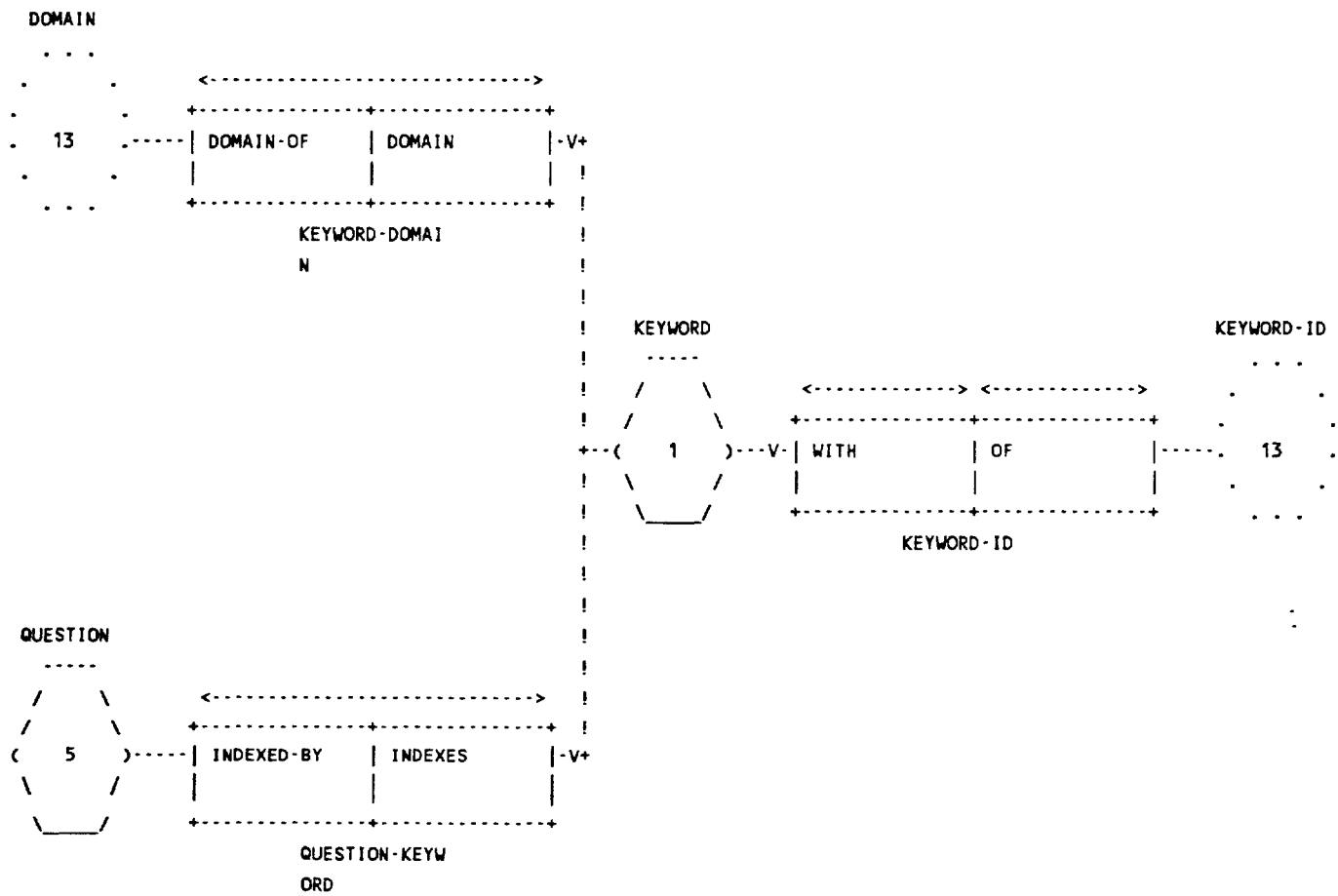
"10H"

""

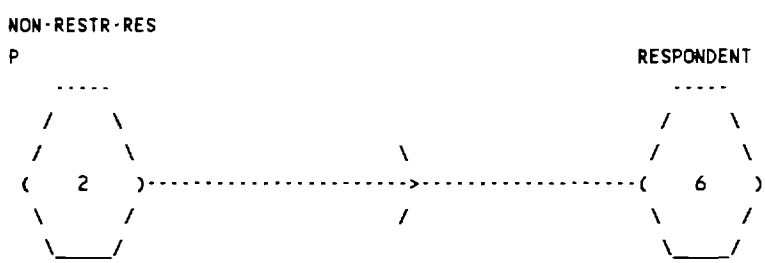
PRECISE* PC-IAST 2.40a TABLE OF CONTENTS DATE : 1/ 7/91 TIME : 22.10.15
MODEL NAME: MRST

NOLOT	ON PAGE	NUMBER OF BRIDGE TYPES	NUMBER OF IDEA TYPES	NUMBER OF SUB-TYPES	NUMBER OF SUPER-TYPES
KEYWORD	1	2	1	0	0
NON-RESTR-RESP	2	0	0	0	1
ORG-CHART	3	2	1	0	0
PAGE	4	1	1	0	0
QUESTION	5	2	2	0	0
RESPONDENT	6	7	3	2	0
RESTR-RESP	8	10	0	0	1
SURVEY-IMAGE	10	1	2	0	0

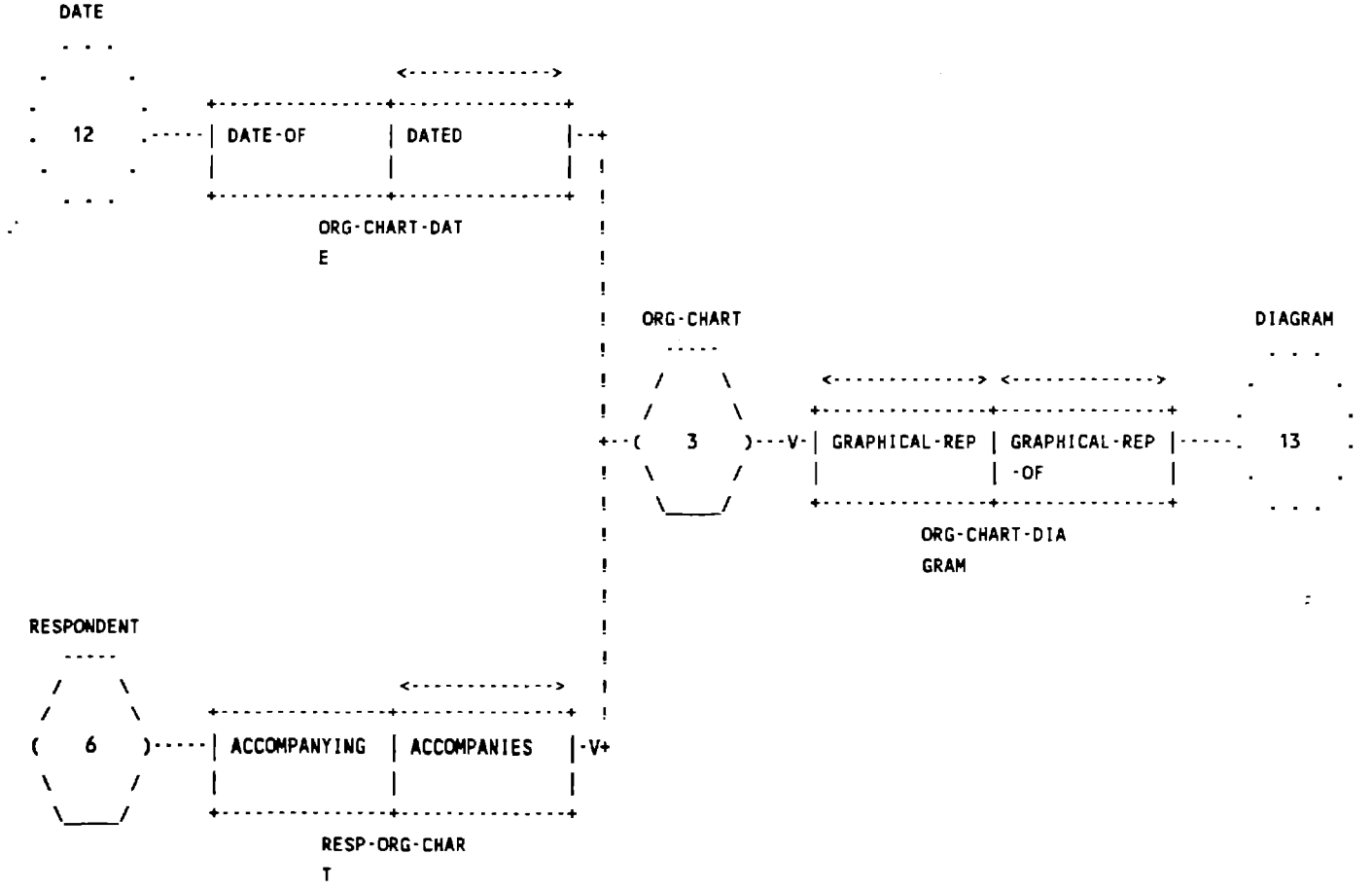
NOLOT : KEYWORD



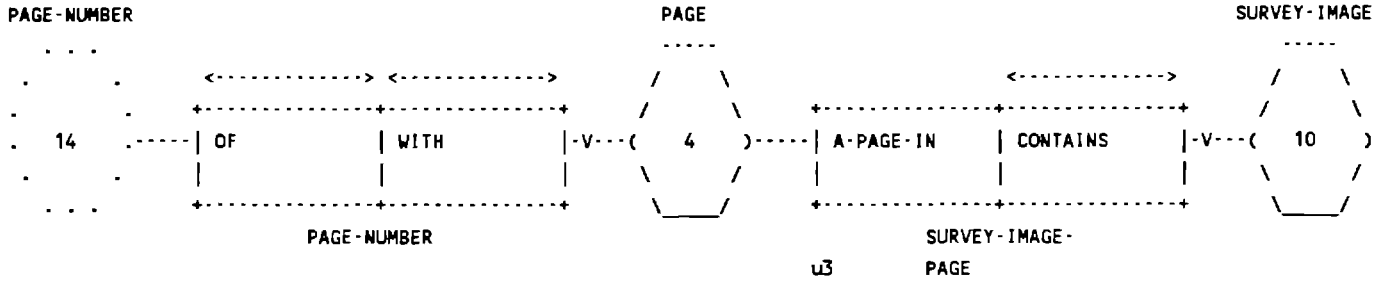
MOLOT : NON-RESTR-RESP



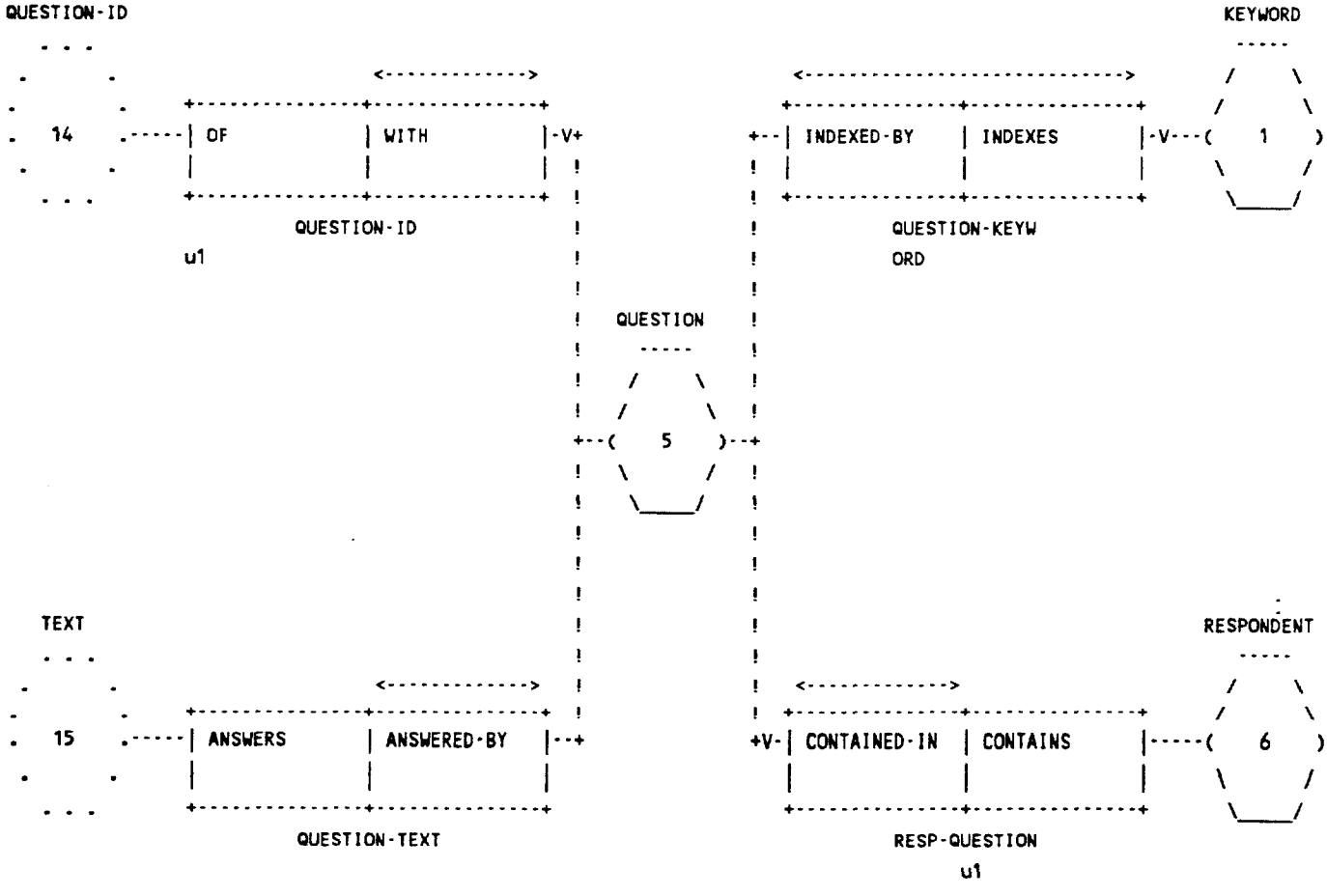
NOLOT : ORG-CHART



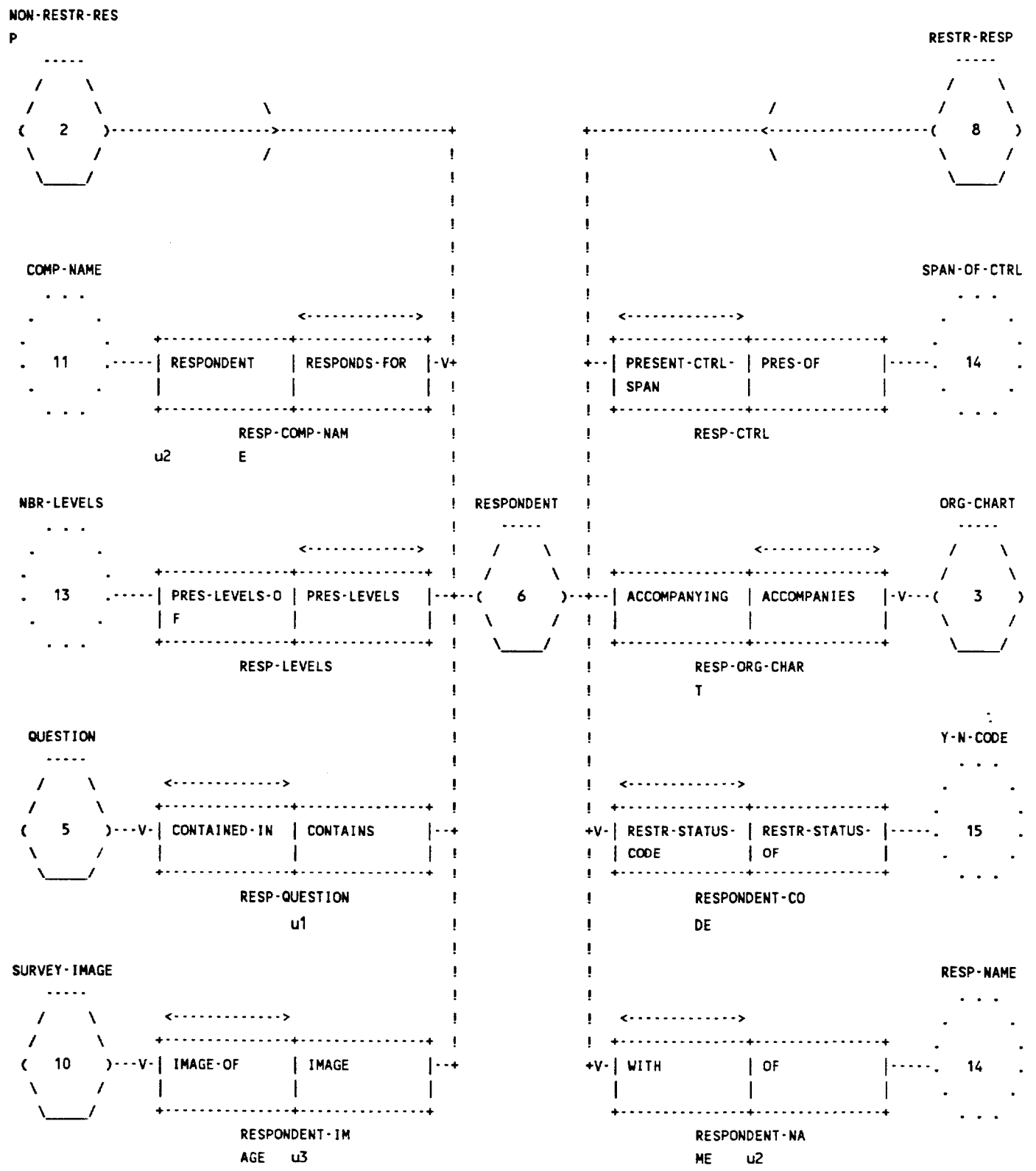
NOLOT : PAGE


PAGE-NUMBER
PAGE
SURVEY-IMAGE
14
OF
WITH
4
A-PAGE-IN
CONTAINS
10
PAGE-NUMBER
SURVEY-IMAGE-
PAGE
u3

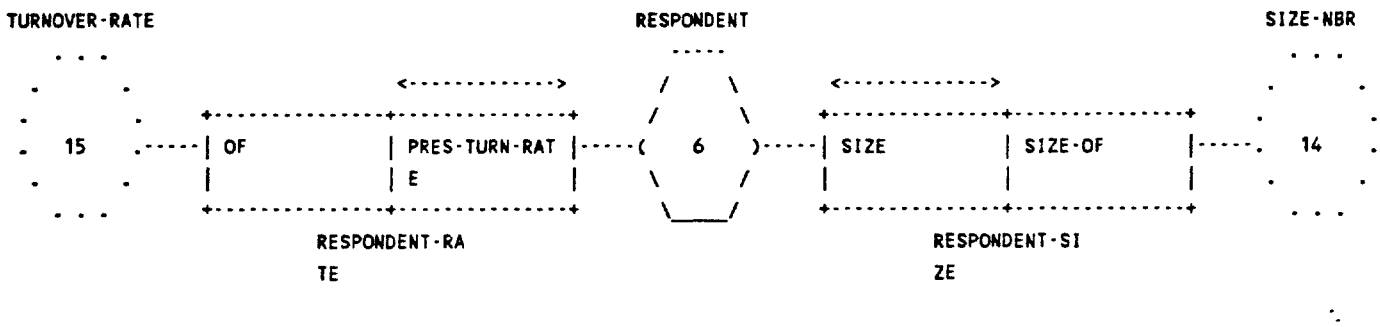
NOLOT : QUESTION



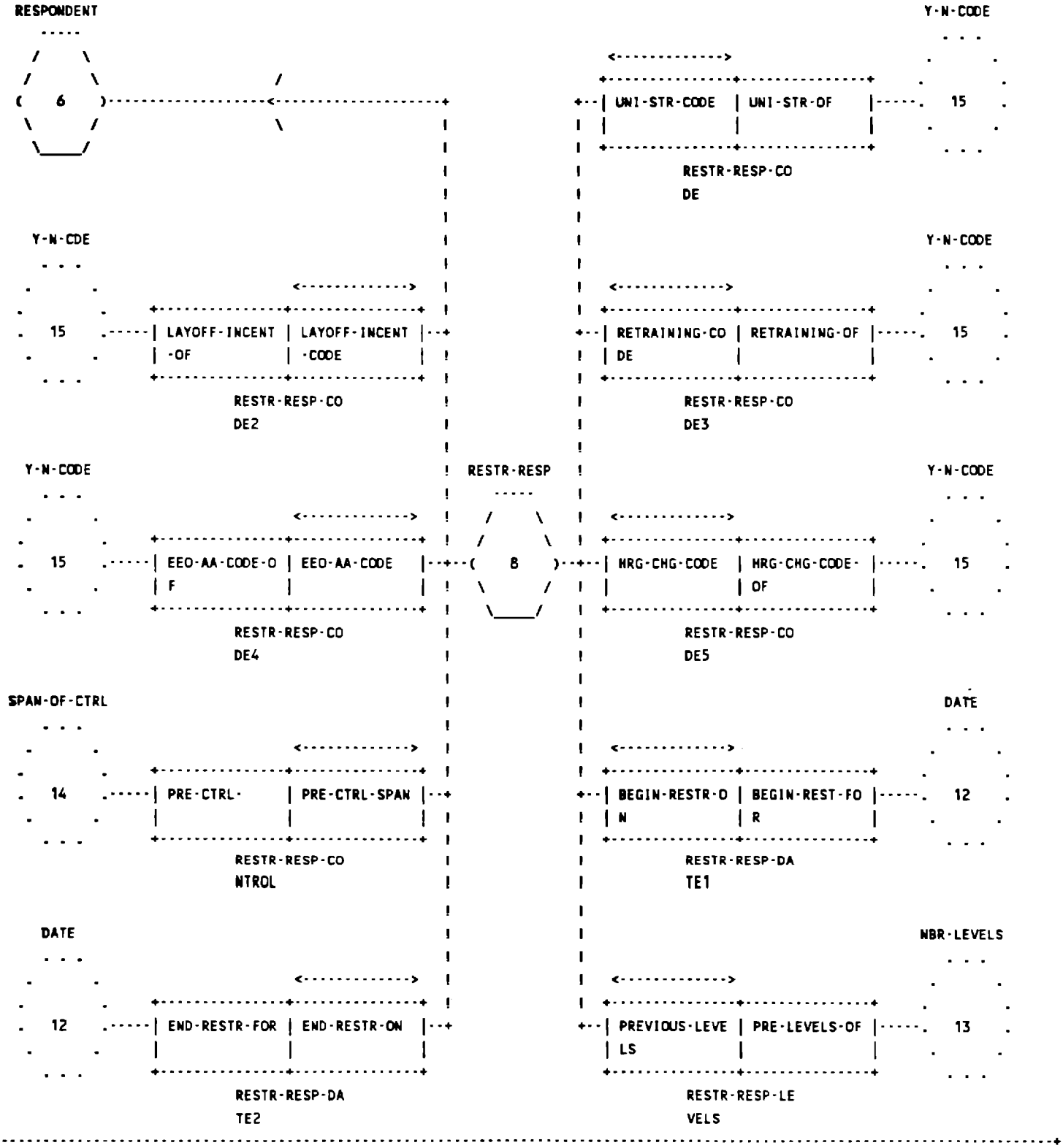
NOLOT : RESPONDENT



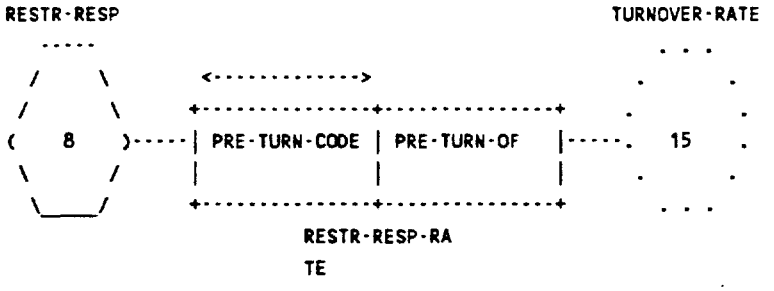
MOLOT : RESPONDENT (continued)



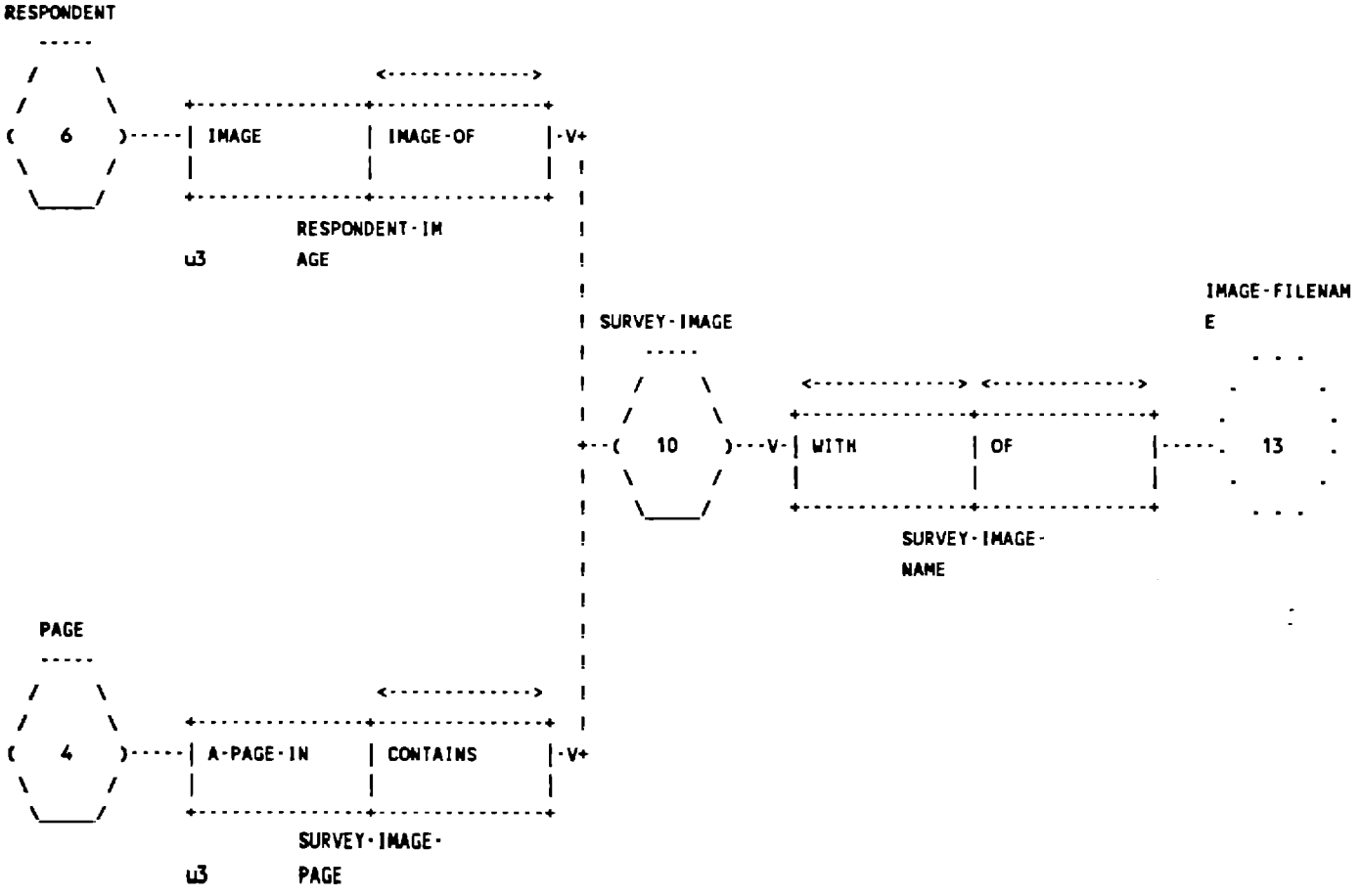
NOLOT : RESTR-RESP



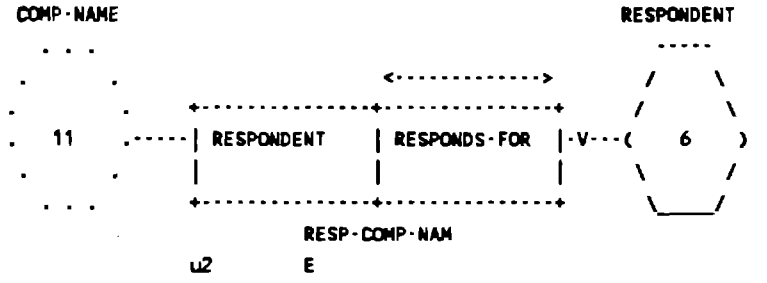
MOLOT : RESTR-RESP (continued)



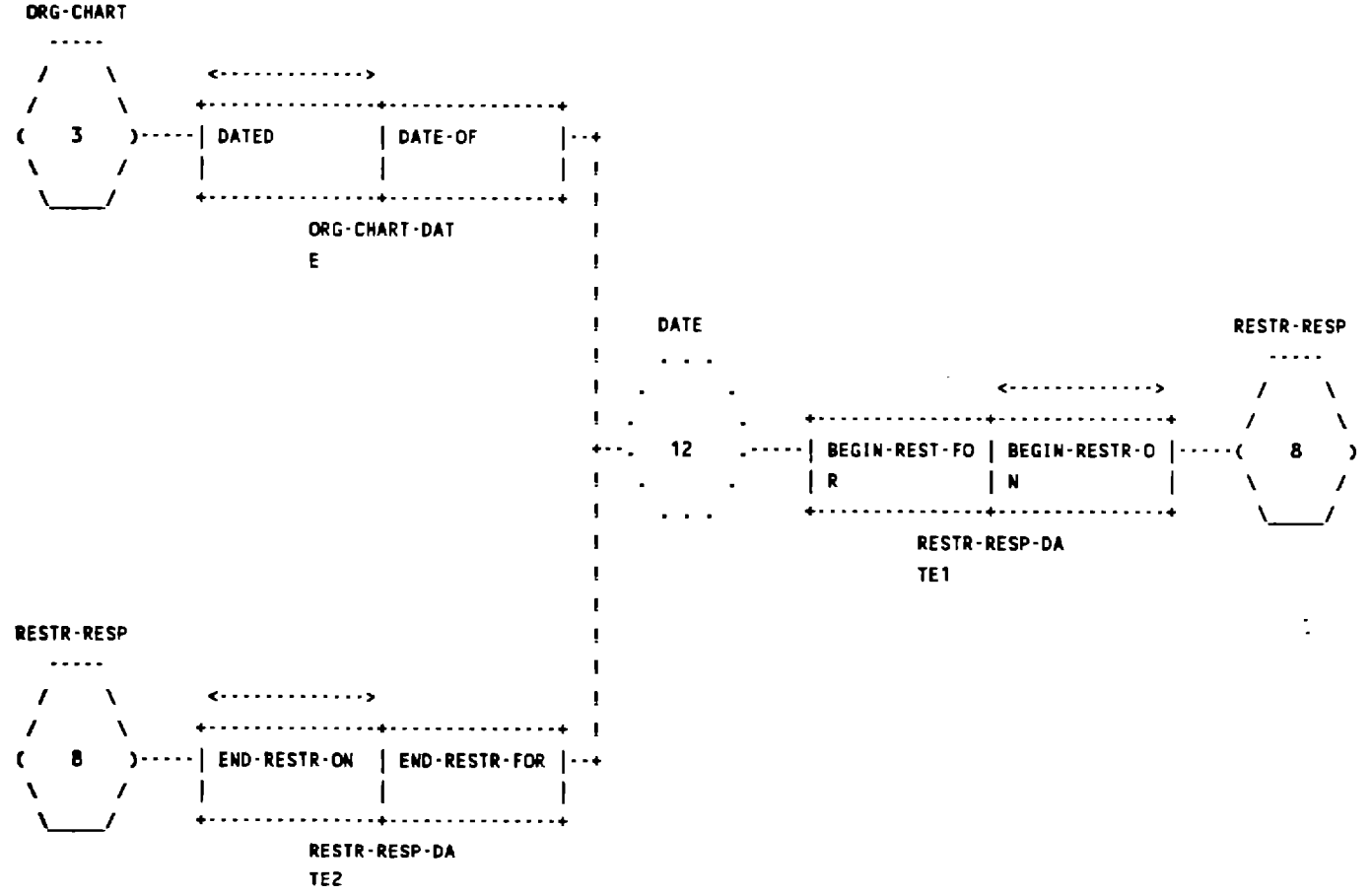
MOLOT : SURVEY-IMAGE



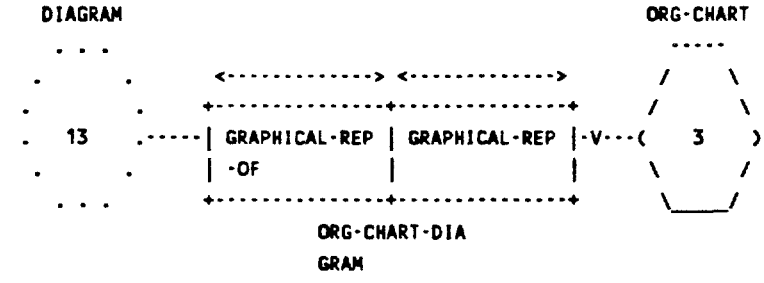
LOT : COMP-NAME

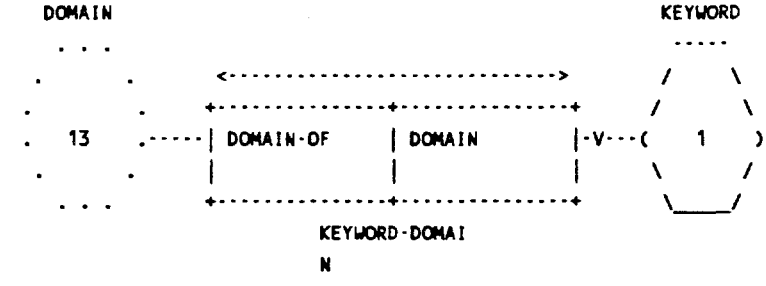


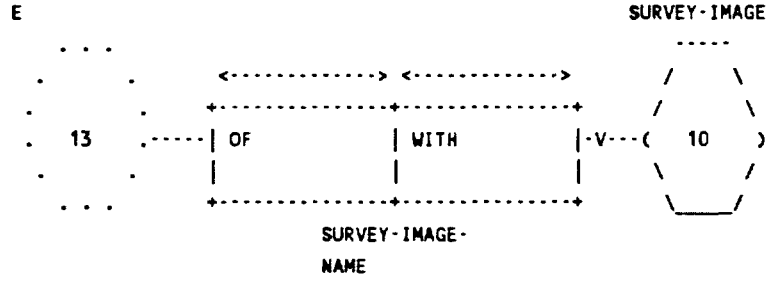
LOT : DATE

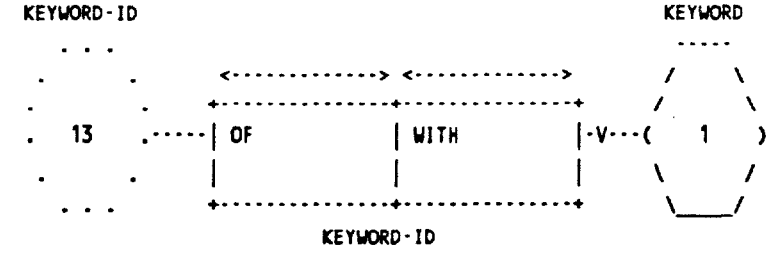

ORG-CHART
3
DATED
DATE-OF
ORG-CHART-DAT
E
DATE
12
BEGIN-REST-FO
R
BEGIN-RESTR-O
N
RESTR-RESP-DA
TE1
RESTR-RESP
8
RESTR-RESP
8
END-RESTR-ON
END-RESTR-FOR
RESTR-RESP-DA
TE2

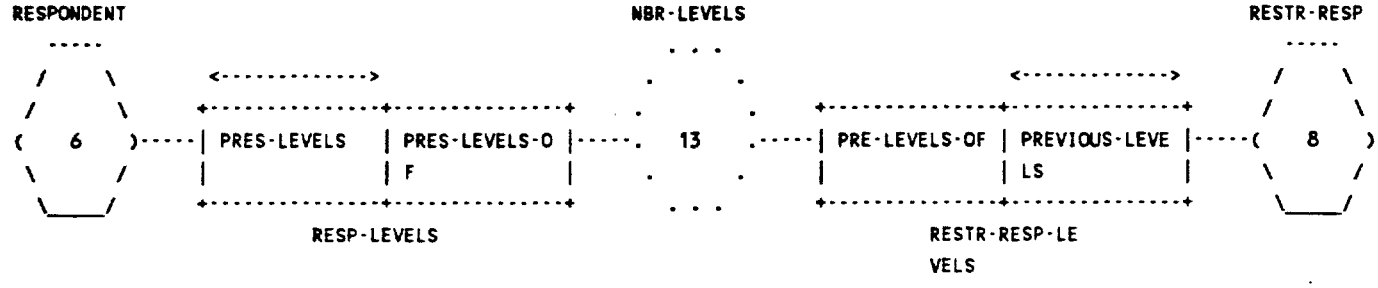
LOT : DIAGRAM DOMAIN IMAGE-FILENAME KEYWORD-ID NBR-LEVELS



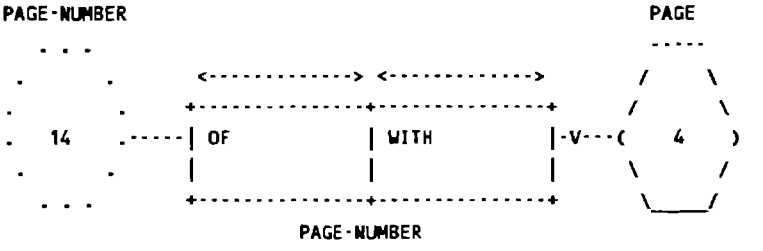


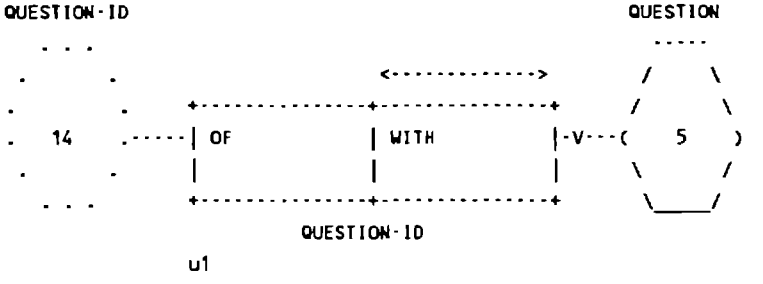


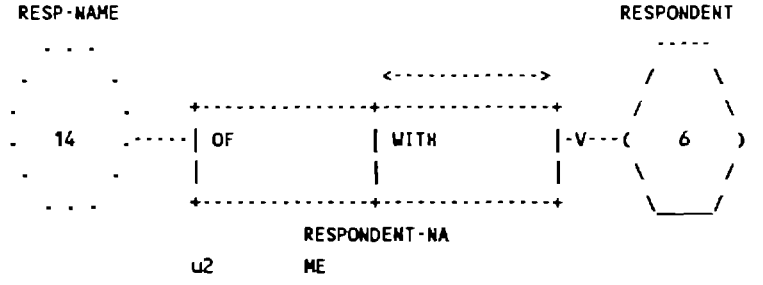


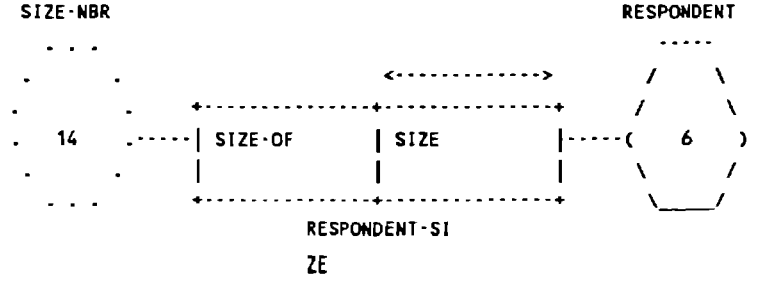


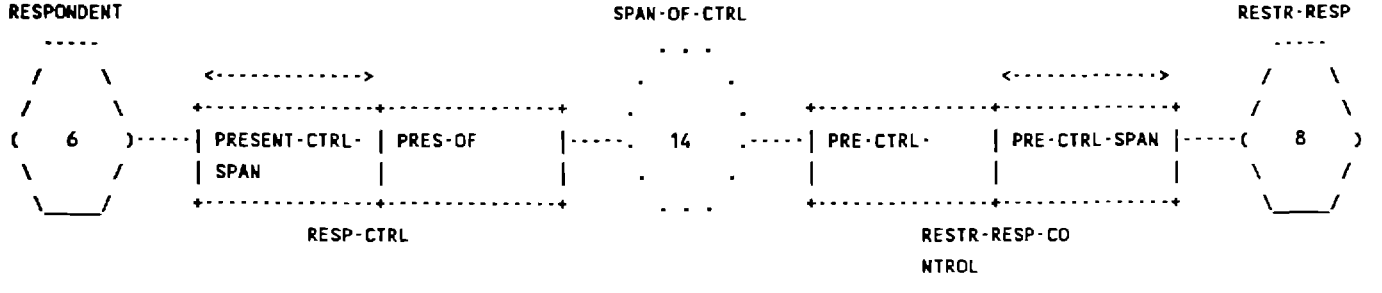
LOT : PAGE-NUMBER QUESTION-ID RESP-NAME SIZE-NBR SPAN-OF-CTRL



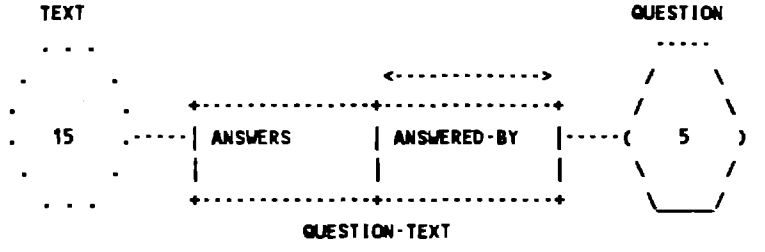


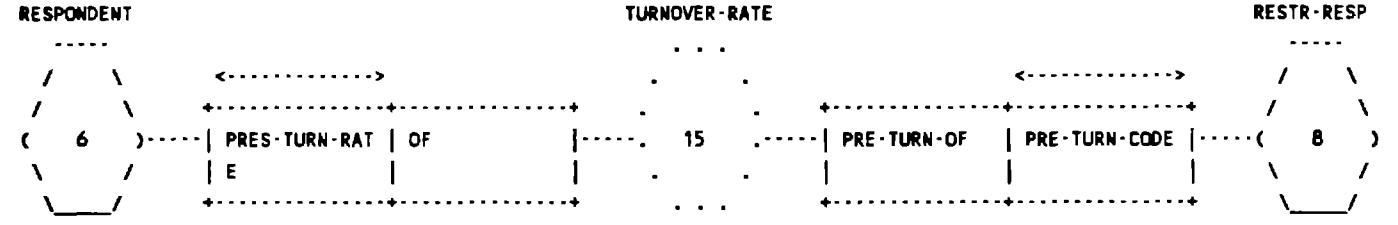


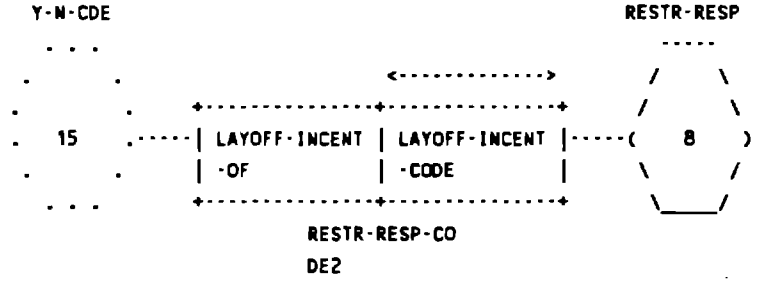




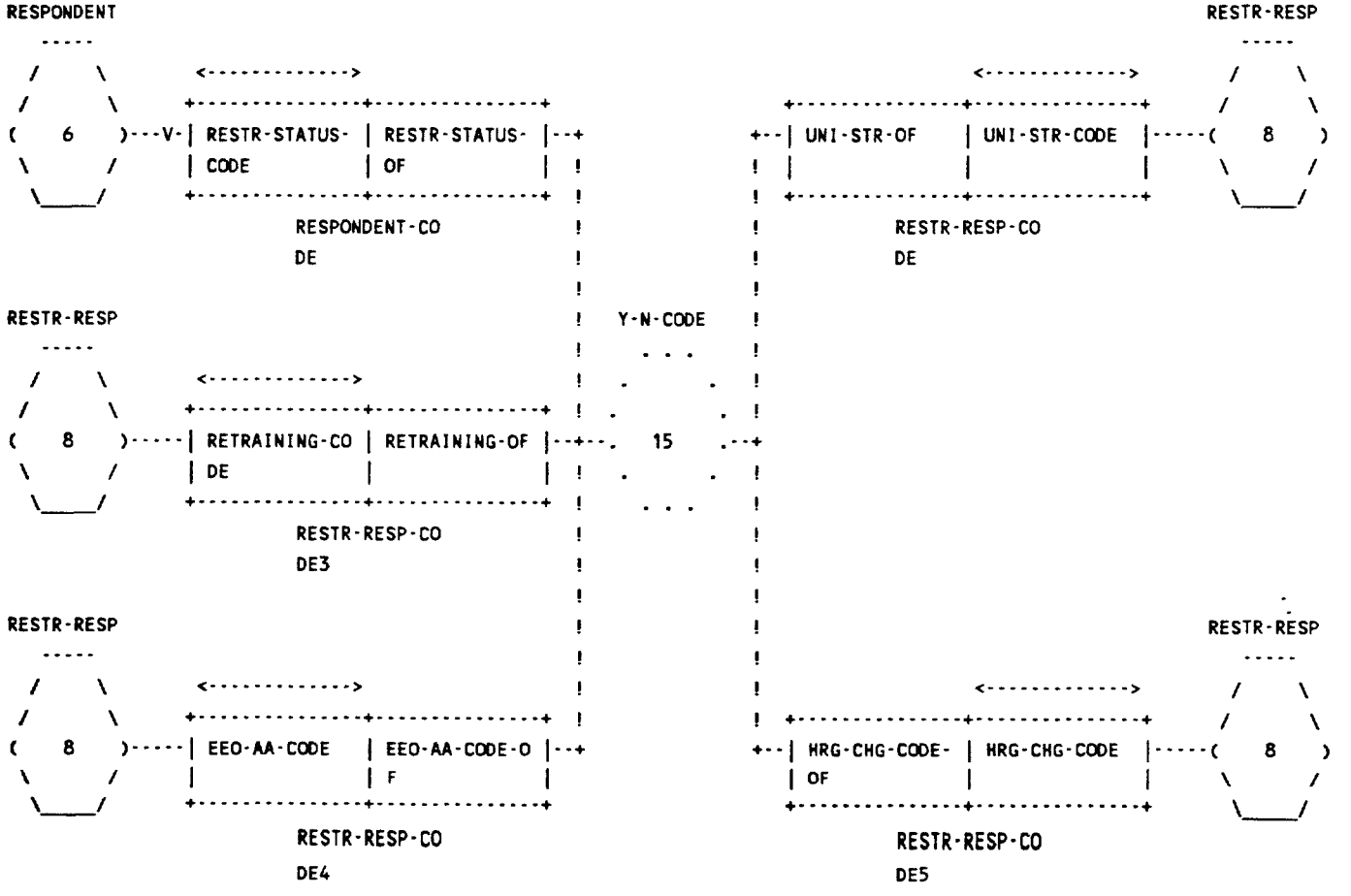
LOT : TEXT TURNOVER-RATE Y-N-CDE Y-N-CODE







LOT :


RESPONDENT
6
RESTR-STATUS-CODE
RESTR-STATUS-OF
RESPONDENT-CODE DE
RESTR-RESP
8
UNI-STR-OF
UNI-STR-CODE
RESTR-RESP-CO DE
RESTR-RESP
8
RETRAINING-CO DE
RETRAINING-OF
RESTR-RESP-CO DE3
Y-N-CODE
15
RESTR-RESP
8
EEO-AA-CODE
EEO-AA-CODE-OF
RESTR-RESP-CO DE4
RESTR-RESP
8
HRG-CHG-CODE-OF
HRG-CHG-CODE
RESTR-RESP-CO DE5

LIST OF CONSTRAINTS

Constraints on roles

---

u1 CUR QUEST-RESP-ID
u2 CUR RESP-COMPNAME
u3 CUR SURVEY-PAGE-RESP

Elementary Sentence.

A COMP-NAME is a LABEL TYPE with representation Character 30.
A COMP-NAME may be RESPONDENT any number of RESPONDENT's.

A DATE is a LABEL TYPE with representation Character 30.
A DATE may be BEGIN-REST-FOR any number of RESTR-RESP's.
A DATE may be DATE-OF any number of ORG-CHART's.
A DATE may be END-RESTR-FOR any number of RESTR-RESP's.

A DIAGRAM is a LABEL TYPE with representation Character 30.
A DIAGRAM may be GRAPHICAL-REP-OF at most one ORG-CHART.

A DOMAIN is a LABEL TYPE with representation Character 30.
A DOMAIN may be DOMAIN-OF any number of KEYWORD's.

An IMAGE-FILENAME is a LABEL TYPE with representation Character 30.
An IMAGE-FILENAME may be OF at most one SURVEY-IMAGE.

A KEYWORD is a OBJECT.
A KEYWORD must be DOMAIN one or more DOMAIN's.
A KEYWORD must be INDEXES one or more QUESTION's.
A KEYWORD must be WITH exactly one KEYWORD-ID.

A KEYWORD-ID is a LABEL TYPE with representation Character 30.
A KEYWORD-ID may be OF at most one KEYWORD.

A NBR-LEVELS is a LABEL TYPE with representation Character 30.
A NBR-LEVELS may be PRE-LEVELS-OF any number of RESTR-RESP's.
A NBR-LEVELS may be PRES-LEVELS-OF any number of RESPONDENT's.

A NON-RESTR-RESP is a OBJECT.
A NON-RESTR-RESP is always a kind of RESPONDENT.

Elementary Sentence.

An ORG-CHART is a OBJECT.
An ORG-CHART must be ACCOMPANIES exactly one RESPONDENT.
An ORG-CHART may be DATED at most one DATE.
An ORG-CHART must be GRAPHICAL-REP exactly one DIAGRAM.

A PAGE is a OBJECT.
A PAGE may be A-PAGE-IN any number of SURVEY-IMAGE's.
A PAGE must be WITH exactly one PAGE-NUMBER.

A PAGE-NUMBER is a LABEL TYPE with representation Binary integer 6.
A PAGE-NUMBER may be OF at most one PAGE.

A QUESTION is a OBJECT.
A QUESTION may be ANSWERED-BY at most one TEXT.
A QUESTION must be CONTAINED-IN exactly one RESPONDENT.
A QUESTION may be INDEXED-BY any number of KEYWORD's.
A QUESTION must be WITH exactly one QUESTION-ID.
Every QUESTION
 is associated uniquely with one combination of
 a QUESTION ID OF the QUESTION
and a RESPONDENT CONTAINS the QUESTION.

A QUESTION-ID is a LABEL TYPE with representation Character 30.
A QUESTION-ID may be OF any number of QUESTION's.

A RESP-NAME is a LABEL TYPE with representation Character 30.
A RESP-NAME may be OF any number of RESPONDENT's.

A RESPONDENT is a OBJECT.
A RESPONDENT may be ACCOMPANYING any number of ORG-CHART's.
A RESPONDENT may be CONTAINS any number of QUESTION's.
A RESPONDENT may be IMAGE any number of SURVEY-IMAGE's.
A RESPONDENT may be PRES-LEVELS at most one NBR-LEVELS.
A RESPONDENT may be PRES-TURN-RATE at most one TURNOVER-RATE.
A RESPONDENT may be PRESENT-CTRL-SPAN at most one SPAN-OF-CTRL.
A RESPONDENT must be RESPONDS-FOR exactly one COMP-NAME.

Elementary Sentence.

A RESPONDENT must be RESTR-STATUS-CODE exactly one Y-N-CODE.
A RESPONDENT may be SIZE at most one SIZE-NBR.
A RESPONDENT must be WITH exactly one RESP-NAME.
Every RESPONDENT
 is associated uniquely with one combination of
    a COMP NAME RESPONDENT the RESPONDENT
and a RESP NAME OF the RESPONDENT.

A RESTR-RESP is a OBJECT.
A RESTR-RESP is always a kind of RESPONDENT.
A RESTR-RESP may be BEGIN-RESTR-ON at most one DATE.
A RESTR-RESP may be EEO-AA-CODE at most one Y-N-CODE.
A RESTR-RESP may be END-RESTR-ON at most one DATE.
A RESTR-RESP may be HRG-CHG-CODE at most one Y-N-CODE.
A RESTR-RESP may be LAYOFF-INCENT-CODE at most one Y-N-CDE.
A RESTR-RESP may be PRE-CTRL-SPAN at most one SPAN-OF-CTRL.
A RESTR-RESP may be PRE-TURN-CODE at most one TURNOVER-RATE.
A RESTR-RESP may be PREVIOUS-LEVELS at most one NBR-LEVELS.
A RESTR-RESP may be RETRAINING-CODE at most one Y-N-CODE.
A RESTR-RESP may be UNI-STR-CODE at most one Y-N-CODE.

A SIZE-NBR is a LABEL TYPE with representation Character 30.
A SIZE-NBR may be SIZE-OF any number of RESPONDENT's.

A SPAN-OF-CTRL is a LABEL TYPE with representation Character 30.
A SPAN-OF-CTRL may be PRE-CTRL- any number of RESTR-RESP's.
A SPAN-OF-CTRL may be PRES-OF any number of RESPONDENT's.

A SURVEY-IMAGE is a OBJECT.
A SURVEY-IMAGE must be CONTAINS exactly one PAGE.
A SURVEY-IMAGE must be IMAGE-OF exactly one RESPONDENT.
A SURVEY-IMAGE must be WITH exactly one IMAGE-FILENAME.
Every SURVEY IMAGE
 is associated uniquely with one combination of
    a PAGE A PAGE IN the SURVEY IMAGE
and a RESPONDENT IMAGE the SURVEY IMAGE.

A TEXT is a LABEL TYPE with representation Character 30.
A TEXT may be ANSWERS any number of QUESTION's.

Elementary Sentence.

A TURNOVER-RATE is a LABEL TYPE with representation Character 30.
A TURNOVER-RATE may be OF any number of RESPONDENT's.
A TURNOVER-RATE may be PRE-TURN-OF any number of RESTR-RESP's.

A Y-N-CDE is a LABEL TYPE with representation Character 30.
A Y-N-CDE may be LAYOFF-INCENT-OF any number of RESTR-RESP's.

A Y-N-CODE is a LABEL TYPE with representation Character 30.
A Y-N-CODE may be EEO-AA-CODE-OF any number of RESTR-RESP's.
A Y-N-CODE may be HRG-CHG-CODE-OF any number of RESTR-RESP's.
A Y-N-CODE may be RESTR-STATUS-OF any number of RESPONDENT's.
A Y-N-CODE may be RETRAINING-OF any number of RESTR-RESP's.
A Y-N-CODE may be UNI-STR-OF any number of RESTR-RESP's.

3. Database Definition

The formal data model derived from the information model in Section 2 was translated into the database definition contained in this section. This relational definition, produced directly from the data model, is in fifth-normal-form. Fifth-normal-form implies that the structure is elementary because the same information content (data and relationships) cannot be reconstructed from smaller record types derived from the original (fifth-normal-form) representation. Such an elementary representation is particularly productive because of the flexibility it offers when making enhancements to accommodate additional requirements. Extensions can be made without rendering prior functionality inoperable. As this is a prototype system, extensions may be desired in the future.

The database definition has been used to create an INGRES. relational database. The SQL definitions produce the tables which comprise the database. Following are the SQL statements which define and create tables in the database implemented for the MRST Prototype Information System.

```
CREATE TABLE KEYWORD_DOMAIN     (DOMAIN_DOMAIN_OF                varchar(30) not null,
                                 KEYWORD_ID_DOMAIN               varchar(30) not null);
CREATE UNIQUE INDEX IDX1 ON KEYWORD_DOMAIN     (DOMAIN_DOMAIN_OF,
                                  KEYWORD_ID_DOMAIN);
CREATE TABLE NON_RESTR_RESP     (RESP_NAME_OF                    varchar(30) not null,
                                 COMP_NAME_RESPONDENT            varchar(30) not null);
CREATE UNIQUE INDEX IDX2 ON NON_RESTR_RESP     (RESP_NAME_OF,
                                  COMP_NAME_RESPONDENT);
CREATE TABLE ORG_CHART          (GRAPHICAL_REP_OF                varchar(90) not null,
                                 RESP_NAME_ACCOMPANYING          varchar(30) not null,
                                 COMP_NAME_ACCOMPANYING          varchar(30) not null,
                                 DATE_DATE_OF                    varchar(4));
CREATE UNIQUE INDEX IDX3 ON ORG_CHART          (RESP_NAME_ACCOMPANYING,
                                  COMP_NAME_ACCOMPANYING,
                                  DATE_DATE_OF);
CREATE UNIQUE INDEX IDX4 ON ORG_CHART          (GRAPHICAL_REP_OF);
CREATE TABLE QUESTION           (QUESTION_ID_OF                  varchar(30) not null,
                                 COMP_NAME_CONTAINS              varchar(30) not null,
                                 RESP_NAME_CONTAINS              varchar(30) not null,
                                 TEXT_ANSWERS                    varchar(90));
CREATE UNIQUE INDEX IDX5 ON QUESTION           (QUESTION_ID_OF,
                                  COMP_NAME_CONTAINS,
                                  RESP_NAME_CONTAINS);
CREATE TABLE QUESTION_KEYWORD   (KEYWORD_ID_INDEXES              varchar(30) not null,
                                 QUESTION_ID_INDX_BY             varchar(30) not null,
                                 COMP_NAME_INDEXED_BY            varchar(30) not null,
                                 RESP_NAME_INDEXED_BY            varchar(30) not null);
CREATE UNIQUE INDEX IDX6 ON QUESTION_KEYWORD   (KEYWORD_ID_INDEXES,
                                  QUESTION_ID_INDX_BY,
                                  COMP_NAME_INDEXED_BY,
                                  RESP_NAME_INDEXED_BY);
CREATE TABLE RESPONDENT         (Y_N_CODE_RESTR_STATUS_OF        varchar(1) not null,
                                 RESP_NAME_OF                    varchar(30) not null,
                                 COMP_NAME_RESPONDENT            varchar(30) not null,
                                 NO_LEVELS_PRES_LEVELS_OF        varchar(8),
                                 SIZE_NBR_SIZE_OF                varchar(8),
                                 SPAN_OF_CTRL_PRES_OF            varchar(8),
                                 TURNOVER_RATE_OF                varchar(8));
CREATE UNIQUE INDEX IDX7 ON RESPONDENT         (RESP_NAME_OF,
                                  COMP_NAME_RESPONDENT);
CREATE TABLE RESPONDENT_IMAGE   (RESP_NAME_IMAGE                 varchar(30) not null,
                                 COMP_NAME_IMAGE                 varchar(30) not null,
                                 PAGE_NUMBER_IMAGE_OF            integer not null,
                                 IMAGE_FILENAME                  varchar(90) not null);
CREATE UNIQUE INDEX IDX8 ON RESPONDENT_IMAGE   (RESP_NAME_IMAGE,
                                  COMP_NAME_IMAGE,
                                  PAGE_NUMBER_IMAGE_OF);
CREATE TABLE RESTR_RESP         (RESP_NAME_OF                    varchar(30) not null,
                                 COMP_NAME_RESPONDENT            varchar(30) not null,
                                 DATE_BEGIN_REST_FOR             varchar(4),
                                 DATE_END_RESTR_FOR              varchar(4),
                                 NBR_LEVELS_PRE_LEVELS_OF        varchar(8),
                                 SPAN_OF_CTRL_PRE_CTRL           varchar(8),
                                 TURNOVER_RATE_PRE_T_R_OF        varchar(8),
                                 Y_N_CODE_LAYOFF_INCEN_OF        varchar(1),
                                 Y_N_CODE_EEO_AA_CODE_OF         varchar(1),
                                 Y_N_CODE_HRG_CHG_CODE_OF        varchar(1),
                                 Y_N_CODE_RETRAINING_OF          varchar(1),
                                 Y_N_CODE_UNI_STR_OF             varchar(1));
CREATE UNIQUE INDEX IDX9 ON RESTR_RESP         (RESP_NAME_OF,
```

COMP_NAME_RESPONDENT);

4. Screen Definitions

The MRST applications are comprised of a collection of frames, each consisting of a screen and a number of operations. A screen is required for each frame. Windows 4GL refers to a screen definition as a component (of an application). Components were generated using 4GL application editor [7]. The first form to be displayed by the primary (restructure analysis) application is the Main Menu screen, as the application has multiple functional areas. The first component to be displayed by the secondary (keyword indexing) application is the Keyword Index screen, which is the main screen for this application's one functional area. Other components are displayed as necessary to carry out the operations specified by the user. These components are documented in this section, arranged in alphabetical order by name.

Components provide the input and output facilities for the application. Buttons and menus of functions control the application flow. Following are the technical specifications for the components generated for the MRST Prototype Information System. The corresponding component scripts carry the same names as the forms and are documented in a later section of this report.

Basic Restructuring Information

Respondent Name | <company> | <respondent>

Begin Date <begin-date> <end-date>

Size <size>

Is structure uniform over all lines of business? (1)

Were layoff incentives offered? (2)

Were employees retrained? (3)

Did restructuring impact: EEO/AA profile? (4)

hiring procedures? (5)

SELECT COMPARE CANCEL

BASIC INFO Screen

(1) <uni-structure-code>
(2) <layoff-incentive-code>
(3) <retraining-code>
(4) <eeo-aa-code>
(5) <hiring-code>

CHART EXAMINATION

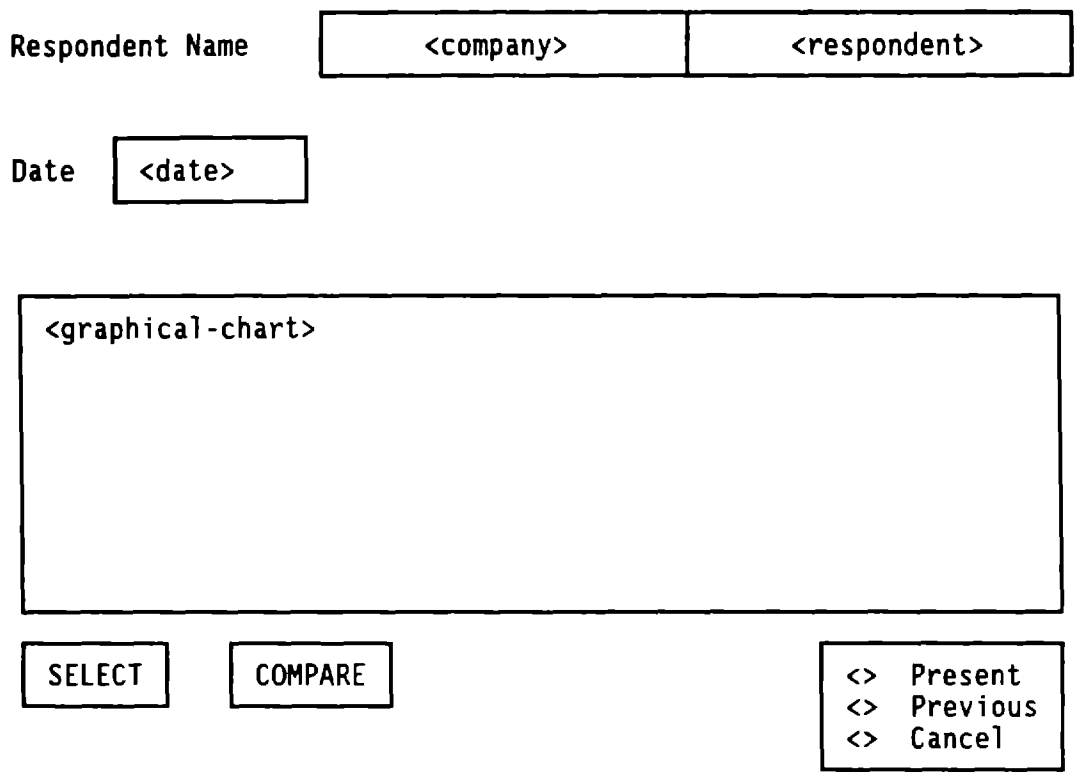
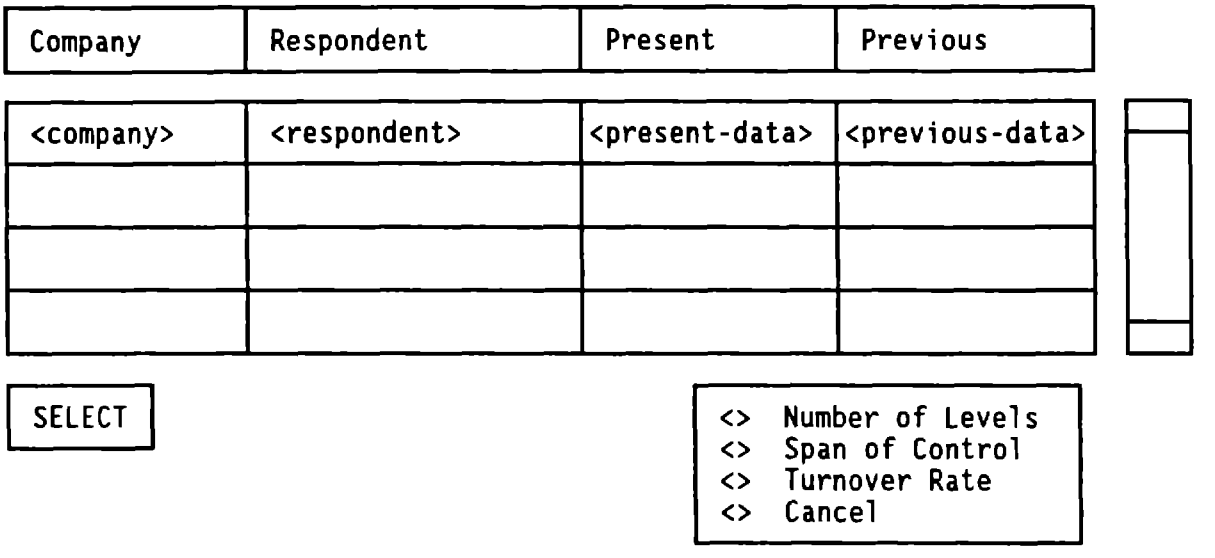


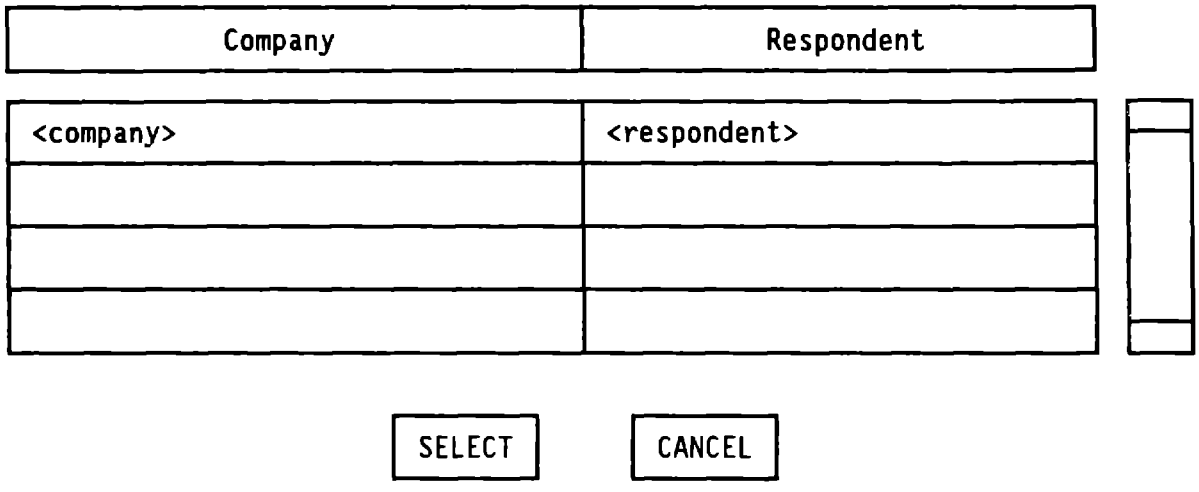
CHART EXAMINATION Screen

COMPARISON OF RESPONDENTS



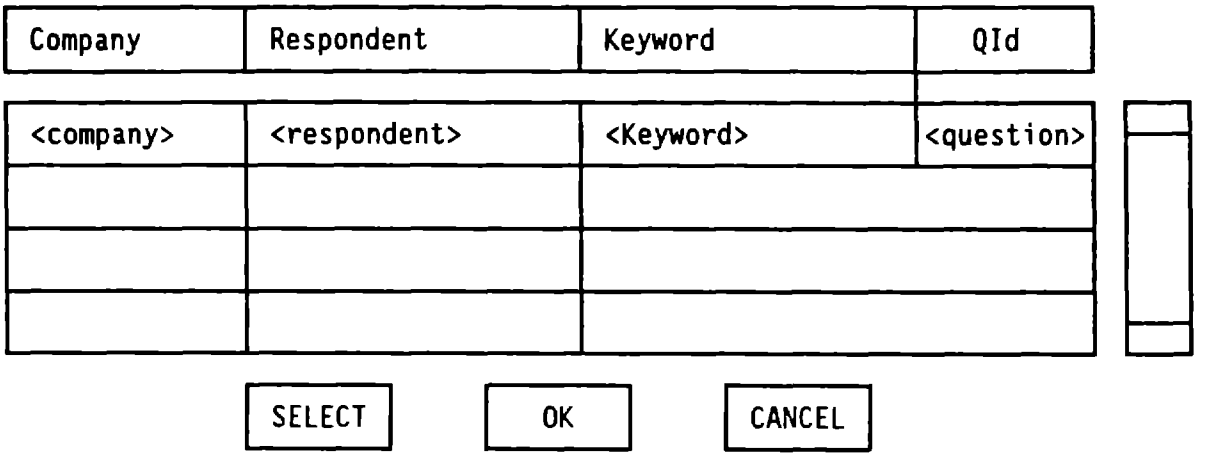
COMPARE INFO Screen

COMPANY / RESPONDENT



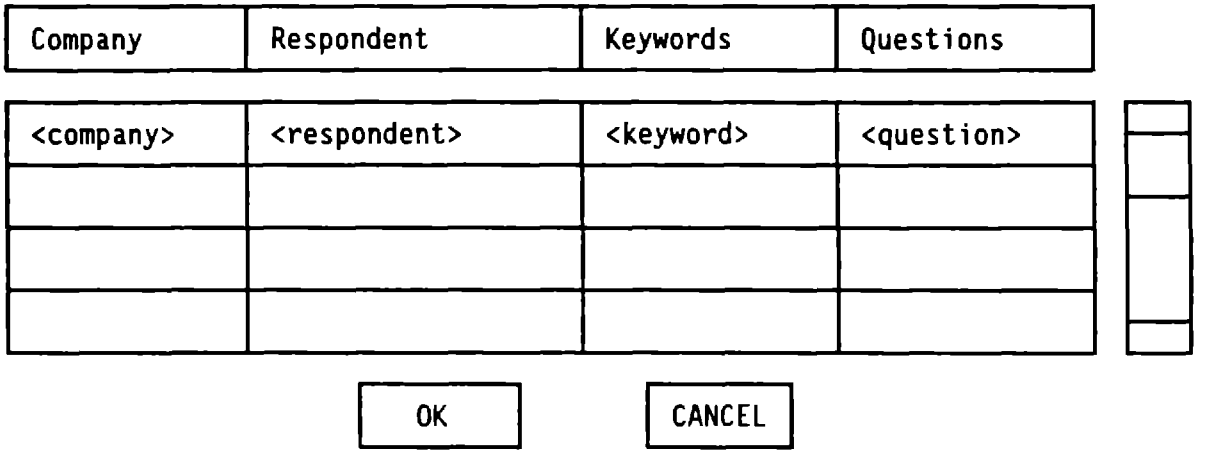
GETNAME Screen

KEYWORD INDEX



KEYWORD INDEX Screen

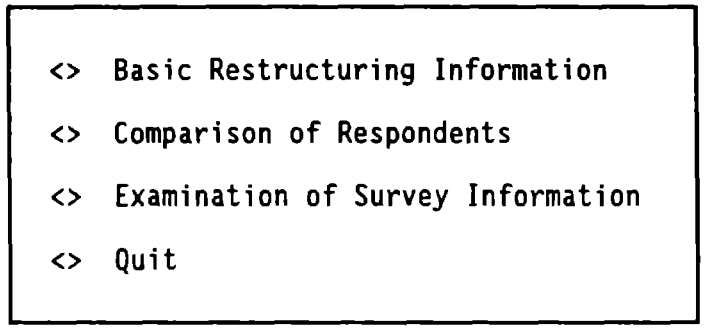
KEYWORD MENU



KEYWORD MENU Screen

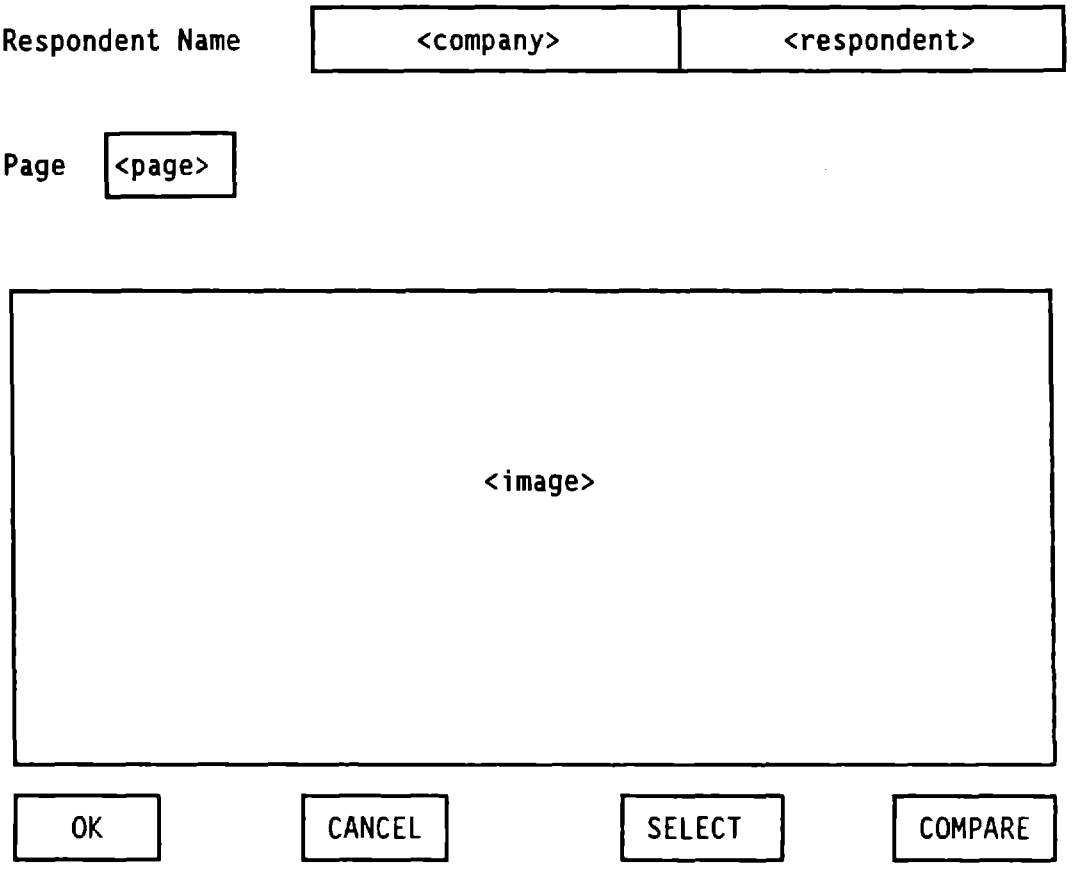
MRST PROTOTYPE INFORMATION SYSTEM

MAIN MENU



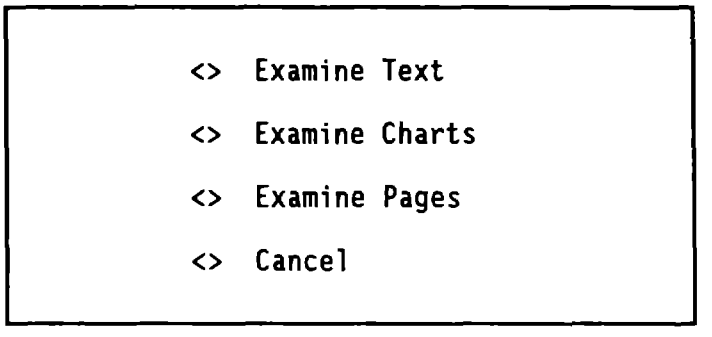
MAIN MENU Screen

PAGE EXAMINATION

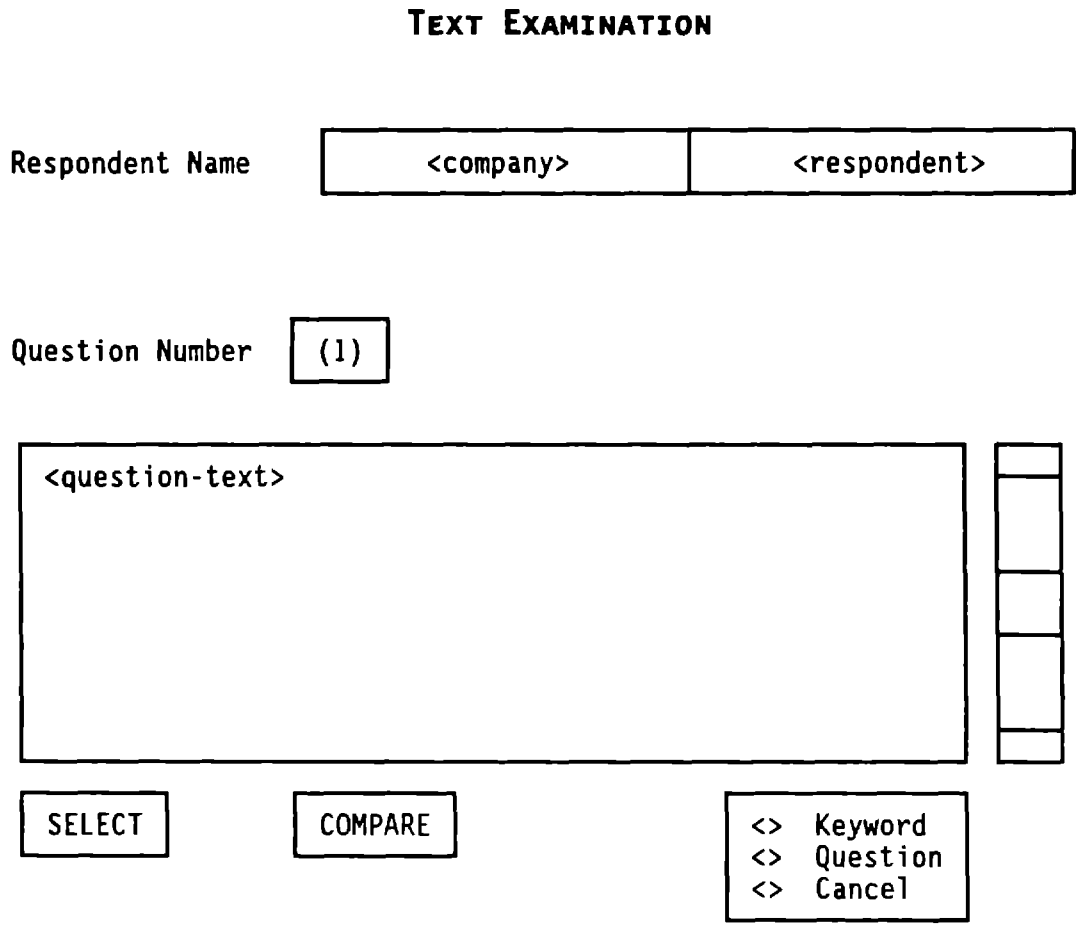


PAGE EXAMINATION Screen

Survey Information Menu



SURVEY INO MENU Screen



TEXT EXAMINATION Screen

(1) <question-number>

5. Frames and Procedures

INGRES/Windows 4GL frames are used to add operation selections to components, and to invoke procedures which control execution of the applications. Functions implemented by the MRST Prototype Information System correspond to the functional requirements identified in the Sandia Management Restructure Study Team (MRST) Prototype Information System Pilot System Requirements Document [8]. The application accomplishes the functions by making calls on procedures written in the 4GL programming language.

Following are the procedures written in 4GL to implement the MRST Prototype Information System. SQL statements in the source define precisely the operations which are being performed on the database. The procedures are arranged in alphabetical order. In the primary (restructure analysis) application, the first component called is the Main Menu screen. In the secondary (keyword indexing) application, the first component to be called is the Keyword Index screen. Control passes from operation selections made by the user, according to the program structure for the MRST system. A procedure script is provided for each component.

BASIC INFORMATION

```
 /*
**  Application:    MRSTDB
**  Frame:          BASICINFO
**      Call up data according to Respondent Name and Company selected
*/
INITIALIZE
(
rname        = varchar(30) not null,
cname        = varchar(30) not null,
bdate        = varchar(4),
edate        = varchar(4),
ynrs         = varchar(1),
uni          = varchar(1),
lay          = varchar(1),
ret          = varchar(1),
eeo          = varchar(1),
hrg          = varchar(1),
sizen        = varchar(10),
comp         = varchar(30) not null,
resp         = varchar(30) not null
) =
{
}

on click select_button =
{       callframe getname (comp := byref(comp),
                           resp := byref(resp))
        with WindowPlacement = WP_PARENTRELATIVE,
             WindowXleft = -70,
             WindowYtop = 3500;
        company = comp;
        respondent = resp;
/*      CurFrame.Flush(); */


     /* Clear out blocks  */
     begin_date = ' ';
     end_date = ' ';
     uni_structure_code = ' ';
     layoff_incentive_code = ' ';
     retraining_code = ' ';
     eeo_aa_code = ' ';
     hiring_code = ' ';


     /* Select respondent name and company  */
     SELECT
            :company                = respondent.comp_name_respondent,
            :respondent             = respondent.resp_name_of,
            :size                   = respondent.size_nbr_size_of,
            :ynrs                   = respondent.y_n_code_restr_status_of
     FROM   respondent
     WHERE  respondent.comp_name_respondent = :company
        and respondent.resp_name_of         = :respondent;
       IF ynrs = 'Y' OR ynrs = 'y' then
         SELECT
            :begin_date             = restr_resp.date_begin_rest_for,
            :end_date               = restr_resp.date_end_restr_for,
            :uni_structure_code     = restr_resp.y_n_code_uni_str_of,
            :layoff_incentive_code = restr_resp.y_n_code_layoff_incen_of,
```

```
            :retraining_code        = restr_resp.y_n_code_retraining_of,
            :eeo_aa_code            = restr_resp.y_n_code_eeo_aa_code_of,
            :hiring_code            = restr_resp.y_n_code_hrg_chg_code_of
          FROM restr_resp
          WHERE restr_resp.comp_name_respondent = :company
               and restr_resp.resp_name_of = :respondent;
        ELSE
          CurFrame.InfoPopup(messagetext = 'No restructure data');
        ENDIF;

}


on click compare_button =
     'begin
          callframe Basicinfo
          with WindowPlacement = WP_PARENTRELATIVE,
               WindowXleft = -70,
               WindowYtop = 3500;
     end;
```

CHART EXAMINATION

```
initialize (comp  = varchar(30) not null,
            resp  = varchar(30) not null,
            restr = varchar(1) not null,
            tfname = varchar(30) not null,
            edate = varchar(4)) =

{
     images.chart_image.FileHandle = ' ';
}

on click select_btn =
{
    callframe getname (comp := byref(comp),
                       resp := byref(resp))
        with WindowPlacement = WP_PARENTRELATIVE,
             WindowXleft = 5000,
             WindowYtop = -1500;
    company = comp;
    respondent  = resp;
    chart_date = ' ';
    CurFrame.Flush();
}


on CLICK operation =
{

     images.chart_image.FileHandle = ' ';
     if operation = 'Cancel' then
         return;
     endif;

     chart_date  = ' ';
     edate       = ' ';
     restr       = ' ';

 /* check to see if company has restructured                    */

     select  :restr = respondent.Y_N_code_restr_status_of
         from respondent
        where comp_name_respondent = :company
          and resp_name_of         = :respondent;

     if :restr = 'Y' then
         select :edate = date_end_restr_for
             from restr_resp
            where comp_name_respondent = :company
              and resp_name_of         = :respondent;
     endif;

     if operation = 'Present' then

 /* if end_date found, look for chart with date >= edate        */
          if :edate  != ' ' then
              select :chart_date  = date_date_of,
                     :images.chart_image.FileHandle = graphical_rep_of
                  from org_chart
                 where comp_name_accompanying = :company
                   and resp_name_accompanying = :respondent
                   and date_date_of >= :edate;
```

```
        else

/*  look for chart with any date                                    */
            select  :chart_date  = date_date_of,
                    :images.chart_image.FileHandle = graphical_rep_of
                from org_chart
               where comp_name_accompanying = :company
                 and resp_name_accompanying = :respondent;
        endif;

        if chart_date = ' ' then
            message 'No organization chart for this company';
        else
            CurFrame.Flush();
        endif;
    elseif operation = 'Previous' then

/* if end_date found, look for chart with date < edate      */

        if :edate  != ' ' then
            select  :chart_date  = date_date_of,
                    :images.chart_image.FileHandle = graphical_rep_of
                from org_chart
               where comp_name_accompanying = :company
                 and resp_name_accompanying = :respondent
                 and date_date_of < :edate;
        endif;

        if chart_date = ' ' then
            message 'No previous organization chart for this company';
        endif;


    endif;

}


on click compare_button =
    begin
        callframe surveyinfomenu
        with WindowPlacement = WP_PARENTRELATIVE,
             WindowXleft = 9000,
             WindowYtop = 2500;
    end;
```

COMPARE INFORMATION

```
initialize
(
    name_num = integer,
    row_num = integer,
    comp = varchar(30) not null,
    resp = varchar(30) not null,
) =
  begin
  end;

on click comp_resp =
  .  begin
          name_num = name_num + 1;
          callframe getname (comp := byref(comp),
                             resp := byref(resp))
          with WindowPlacement = WP_PARENTRELATIVE,
               WindowXleft = -70,
               WindowYtop = 2000;
          comptbl[name_num].company = comp;
          comptbl[name_num].respondent = resp;
     end;



on click operation =
     begin
          commit work;

          row_num = 1;
          while row_num <= comptbl.LastRow() do
               comptbl[row_num].present_data = ' ';
               comptbl[row_num].previous_data = ' ';
               row_num = row_num + 1;
          endwhile;



          if operation = 'Number of Levels' then
               row_num = 1;
               while row_num <= comptbl.LastRow() do
                    select :comptbl[row_num].present_data =  NO_LEVELS_PRES_LEVELS_OF
                    from RESPONDENT
                    where RESP_NAME_OF = :comptbl[row_num].respondent
                    and COMP_NAME_RESPONDENT = :comptbl[row_num].company;
                    select :comptbl[row_num].previous_data = NBR_LEVELS_PRE_LEVELS_OF
                    from RESTR_RESP
                    where RESP_NAME_OF = :comptbl[row_num].respondent
                    and COMP_NAME_RESPONDENT = :comptbl[row_num].company;
                    row_num = row_num + 1;
               endwhile;



          elseif operation = 'Span of Control' then
               row_num = 1;
               while row_num <= comptbl.LastRow() do
```

```
            select :comptbl[row_num].present_data  = SPAN_OF_CTRL_PRES_OF
            from RESPONDENT
            where RESP_NAME_OF = :comptbl[row_num].respondent
            and COMP_NAME_RESPONDENT = :comptbl[row_num].company;
            select :comptbl[row_num].previous_data = SPAN_OF_CTRL_PRE_CTRL_
            from RESTR_RESP
            where RESP_NAME_OF = :comptbl[row_num].respondent
            and COMP_NAME_RESPONDENT = :comptbl[row_num].company;
            row_num = row_num + 1;
        endwhile;


    elseif operation = 'Turnover Rate' then
        row_num = 1;
        while row_num <= comptbl.LastRow() do
            select :comptbl[row_num].present_data = TURNOVER_RATE_OF
            from RESPONDENT
            where RESP_NAME_OF = :comptbl[row_num].respondent
            and COMP_NAME_RESPONDENT = :comptbl[row_num].company;
            select :comptbl[row_num].previous_data = TURNOVER_RATE_PRE_T_R_OF
            from RESTR_RESP
            where RESP_NAME_OF = :comptbl[row_num].respondent
            and COMP_NAME_RESPONDENT = :comptbl[row_num].company;
            row_num = row_num + 1;
        endwhile;


    elseif operation = 'Cancel' then
        return;
    endif;
end;
```

GET NAME

```
initialize (
           i = integer,
        comp = varchar(30) not null,
        resp = varchar(30) not null) =
{
        i = 1;
        repeated select
                :respondent_name[i].company = comp_name_respondent,
                :respondent_name[i].respondent = resp_name_of
        from respondent
        order by company,respondent
        begin
                i = i + 1
        end;

        commit;
}


on click select_button =
{
        comp = respondent_name[].company;
        resp = respondent_name[].respondent;

        return;
};

on click cancel_button =
{
        return;
}
```

KEYWORD INDEX

```
initialize
(
    name_num = integer,
    row_num = integer,
    comp = varchar(30) not null,
    resp = varchar(30) not null,
) =
  begin
  end;

on click select_btn =
     begin
          name_num = name_num + 1;
          callframe getname (comp := byref(comp),
                             resp := byref(resp))
          with WindowPlacement = WP_PARENTRELATIVE,
               WindowXleft = -70,
               WindowYtop = 2000;
          kword_index[name_num].company = comp;
          kword_index[name_num].respondent = resp;
     end;

on click ok_btn =
     begin
          row_num = 1;
          while row_num <= kword_index.LastRow() do
               repeated insert into question_keyword (comp_name_indexed_by,
                                                      resp_name_indexed_by,
                                                      keyword_id_indexes,
                                                      question_id_indx_by)
               values (:kword_index[row_num].company,
                       :kword_index[row_num].respondent,
                       :kword_index[row_num].keyword,
                       :kword_index[row_num].q_id);
               row_num = row_num + 1;
           endwhile;
      end;

on click cancel_btn =
     begin
          return;
     end;
```

KEYWORD MENU

```
initialize ( i = integer,
             comp = varchar(30) not null,
             resp = varchar(30) not null,
             question = varchar(30) not null )=
     begin
          i = 1;
          repeated select :kword_index[i].keywords = KEYWORD_ID_INDEXES,
                          :kword_index[i].questions = QUESTION_ID_INDX_BY,
                          :kword_index[i].company = COMP_NAME_INDEXED_BY,
                          :kword_index[i].respondent = RESP_NAME_INDEXED_BY
          from QUESTION_KEYWORD
          order by keywords
          begin
               i = i + 1;
          end;
     end;


on click ok_btn =
     begin
          comp = kword_index[].company;
          resp = kword_index[].respondent;
          question = kword_index[].questions;
          return;
     end;
```

MAIN MENU

```
on click operation =
     begin
          if operation = 'Basic Restructuring Information' then
               callframe Basicinfo;
          elseif operation = 'Comparison of Respondents' then
               callframe Compareinfo;
          elseif operation = 'Examination of Survey Information' then
               callframe surveyinfomenu;
          elseif operation = 'Quit' then
               exit;
          endif;
      end;
```

PAGE EXAMINATION

```
initialize (comp = varchar(30) not null,
            resp = varchar(30) not null,
            rest_code = varchar(1) ) =
{
     page_image.FileHandle = ' ';
}




on click compare_button =
     begin
          callframe surveyinfomenu
          with WindowPlacement = WP_PARENTRELATIVE,
               WindowXleft = 6700,
               WindowYtop = -550;
     end;


on click select_button =
     begin
          page_image.FileHandle = ' ';
          callframe getname (comp :=byref(comp),
                             resp :=byref(resp))
          with WindowPlacement = WP_PARENTRELATIVE,
               WindowXleft = 5000,
               WindowYtop = -1000;
          company = comp;
          respondent = resp;
          CurFrame.Flush();
     end;


on click ok_btn =
     begin
          page_image.FileHandle = ' ';
          select :rest_code = Y_N_CODE_RESTR_STATUS_OF
          from RESPONDENT
          where RESP_NAME_OF = :respondent
          and COMP_NAME_RESPONDENT = :company;
          if page_number < 1 or page_number > 10 then
               message 'Invalid page number!';
          elseif page_number > 7 then
               if rest_code = 'y' or rest_code = 'Y' then
                    message 'Page does not apply to this respondent!';
               endif;
          elseif page_number < 8 and page_number > 1 then
               if rest_code = 'n' or rest_code = 'N' then
                    message 'Page does not apply to this respondent!';
               endif;
          endif;

          select :page_image.FileHandle = IMAGE_FILENAME
          from RESPONDENT_IMAGE
          where RESP_NAME_IMAGE = :resp
          and COMP_NAME_IMAGE = :comp
          and PAGE_NUMBER_IMAGE_OF = :page_number;
      end;
```

SURVEY INFORMATION

```
on click operation =
     begin
          if operation = 'Examine Text' then
               callframe Textexamination;
          elseif operation = 'Examine Charts' then
               callframe Chartexamination;
          elseif operation = 'Examine Pages' then
               callframe Pageexamination;
          elseif operation = 'Cancel' then
               return;
          endif;
     end;
```

TEXT EXAMINATION

```
initialize
(
     comp = varchar(30) not null,
     resp = varchar(30) not null,
     filename = varchar(90) ,
     rest_code = varchar(1) not null,
     i = integer,
     cmd = varchar(70)
) =
{}

on click select_button =
     begin
          callframe getname (comp := byref(comp),
                             resp := byref(resp))
          with WindowPlacement = WP_PARENTRELATIVE,
               WindowXleft = -70,
               WindowYtop = 4300;
          company = comp;
          respondent = resp;
          CurFrame.Flush();
     end;



on click compare_button =
     begin
          callframe surveyinfomenu
          with WindowPlacement = WP_PARENTRELATIVE,
               WindowXleft = -150,
               WindowYtop = 4300;
     end;



on click operation =
     begin
          i = 1;
          while i <= text_tbl.LastRow() do
               text_tbl[i].question_text = ' ';
               i = i + 1;
          endwhile;
          if operation = 'Phrase' then




          elseif operation = 'Keyword' then
               callframe keywordmenu (question := byref(question) ,
                                      comp := byref(comp) ,
                                      resp := byref(resp) )
               with WindowPlacement = WP_PARENTRELATIVE,
                    WindowXleft = 6000,
                    WindowYtop = 4500 ;
               company = comp;
               respondent = resp;
```

```
    elseif operation = 'Question' then
        filename = ' ';
        create table q_text (q_row  varchar(70) );
        if question = ' ' then
            message 'No question is specified!';
        else
            select :filename = TEXT_ANSWERS
            from QUESTION
            where COMP_NAME_CONTAINS = :company
            and RESP_NAME_CONTAINS = :respondent
            and QUESTION_ID_OF = :question;
            if filename = ' ' then
                message 'The respondent did not answer this question';
            else
                cmd := 'mv ' + lowercase(filename) + ' q_text.txt';
                call system cmd;
                commit;
                call system 'sql -s <q_text_withdraw mrstdb';
                i = 1;
                repeated select :text_tbl[i].question_text = q_row
                from q_text
                begin
                    i = i + 1;
                end;
                cmd := 'mv q_text.txt ' + lowercase(filename);
                call system cmd;
                commit;
            endif;









        endif;
        drop table q_text;
        commit;
    elseif operation = 'Cancel' then
        return;
    endif;

end;
```

References

1. Ingres Corporation, "Language Reference Manual for INGRES/Windows 4GL for the UNIX and VMS Operating Systems, "Ingres Corporation, Alameda, CA, INGRES Release 6, August 1990.

2. Sun Microsystems, Inc., "SunOS 4.1.1 Release Manual," Sun Microsystems, Inc., Mountain View, CA, SunOS Part Number 800-5480-10, October 1990.

3. Hafsted, K., and Skagestein, G., "Nijssen Information Analysis Method as a Basis for Construction of Information Systems," NORD-DATA 81 Conference, Copenhagen, 1981, published in DATA, 718, 1981.

4. Relational Technology, Inc., "INGRES ABF/4GL Reference Manual for the UNIX and VMS Operating Systems," Relational Technology, Inc., Alameda, CA, Release 6.3, November 1989.

5. Digital Equipment Corporation, "VMS Version 5.0 Release Notes," Digital Equipment Corporation, Maynard, MA, April 1988.

6. Control Data Corporation, "PRECISE Information Engineering, PC-IAST Users Guide," Control Data Corporation, Minneapolis, MN, 1990.

7. Ingres Corporation, "Application Editor User's Guide for INGRES/Windows 4GL for the UNIX and VMS Operating Systems," Ingres Corporation, INGRES Release 6, August 1990.

8. Wyatt, T. R., "Sandia Management Restructure Study Team (MRST) Prototype Information System Pilot System Requirements Document," Sandia National Laboratories, Division 2818, Albuquerque, NM, May 31, 1991.

TECHNICAL SPECIFICATION FOR THE SANDIA
MANAGEMENT RESTRUCTURE STUDY TEAM (MRST)
PROTOTYPE INFORMATION SYSTEM

APPENDIX A

MRSTDB DATA ENTRY

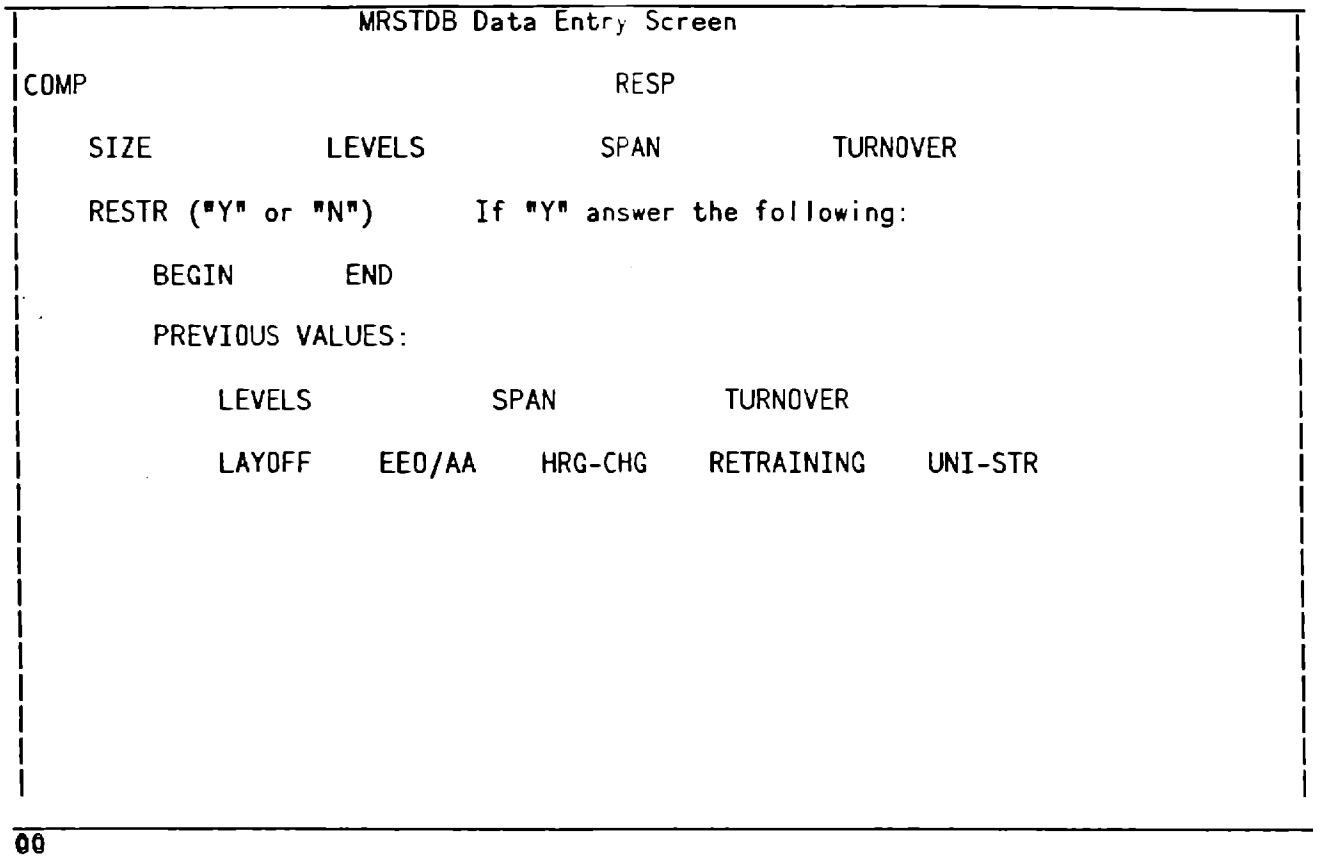
TECHNICAL SPECIFICATION FOR THE SANDIA MANAGEMENT RESTRUCTURE STUDY TEAM (MRST) PROTOTYPE INFORMATION SYSTEM

This appendix contains the technical specification for the data entry application used for the MRST database (MRSTDB). The Version 6.3 INGRES Applications-By-Forms (ABF)[1] was used for the implementation on a VAX Computer running the version 5 VMS[2] operating system. Full-function TCP/IP network connectivity is maintained between the VAX and the Sun workstation on which the MRSTDB database and Windows 4GL retrieval applications reside. This connectivity is provided by WIN/TCP for VMS[3], a member of the WIN/TCP family of products from the Wollongong Group, Inc.

Implementation details for the MRSTDB data entry application include the ABF frames and procedure definitions comprising the remainder of this appendix. The top-level frame is MRSTDB-MAIN, from which MRSTDB_DATA_ENTRY or MRSTDB_TEXT_ENTRY are invoked to input data and text respectively. Two files identified as Q_TEXT_INPUT and Q_TEXT_W NDRAW are also included which contain Standard Query Language (SQL) source required by the application to access textual data stored with the database.

1. ABF is a product and trademark of Pelatroial Technology, Inc., Alameda, California.
2. VMS is a product and trademark of Digital Equipment Corporation, Maynard, Massachusetts.
3. WIN/TCP is a product and trademark of the Wollongong Group, Inc., Paloatts, California.

##



00

Form name: mrstdb_data_entry
Form owner: rchall
Form Display Style is: FullScreen.
Number of columns on screen: 80
Number of lines on screen: 25
Number of fields: 17
Number of trim strings: 3
Date first created: 1991_06_10 18:32:44 GMT
Date last modified: 1991_06_12 16:23:21 GMT

FIELD DESCRIPTIONS

Field name: comp_name
Field title: COMP
Data type: Non-nullable varchar(30)
Display format: c30
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: resp_name
Field title: RESP
Data type: Non-nullable varchar(30)
Display format: c30
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: size
Field title: SIZE
Data type: Nullable varchar(8)
Display format: c8
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: levels
Field title: LEVELS
Data type: Nullable varchar(8)
Display format: c8
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: span
Field title: SPAN
Data type: Nullable varchar(8)
Display format: c8
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: turnover
Field title: TURNOVER
Data type: Nullable varchar(8)
Display format: c8
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: restr
Field title: RESTR ("Y" or "N")
Data type: Non-nullable varchar(1)
Display format: c1
Foreground display color number is: '0'
Special display attributes: Reverse video, Force upper case
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: begin_d
Field title: BEGIN
Data type: Nullable varchar(4)
Display format: c4
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: end_d
Field title: END
Data type: Nullable varchar(4)
Display format: c4
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: plevels
Field title: LEVELS
Data type: Nullable varchar(8)
Display format: c8
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: pspan
Field title: SPAN
Data type: Nullable varchar(8)
Display format: c8
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: pturnover
Field title: TURNOVER
Data type: Nullable varchar(8)
Display format: c8
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

.

Field name: layoff
Field title: LAYOFF
Data type: Nullable varchar(1)
Display format: c1
Foreground display color number is: '0'
Special display attributes: Reverse video, Force upper case
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: eeo_aa
Field title: EEO/AA
Data type: Nullable varchar(1)
Display format: c1
Foreground display color number is: '0'
Special display attributes: Reverse video, Force upper case
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: hrg_chg
Field title: HRG-CHG
Data type: Nullable varchar(1)
Display format: c1
Foreground display color number is: '0'
Special display attributes: Reverse video, Force upper case
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: retraining
Field title: RETRAINING
Data type: Nullable varchar(1)
Display format: c1
Foreground display color number is: '0'
Special display attributes: Reverse video, Force upper case
Type of field: regular
Default value:
Validation check:
Validation error message:

```
Field name: uni_str
Field title: UNI-STR
Data type: Nullable  varchar(1)
Display format: c1
Foreground display color number is: '0'
Special display attributes: Reverse video, Force upper case
Type of field: regular
Default value:
Validation check:
Validation error message:
```

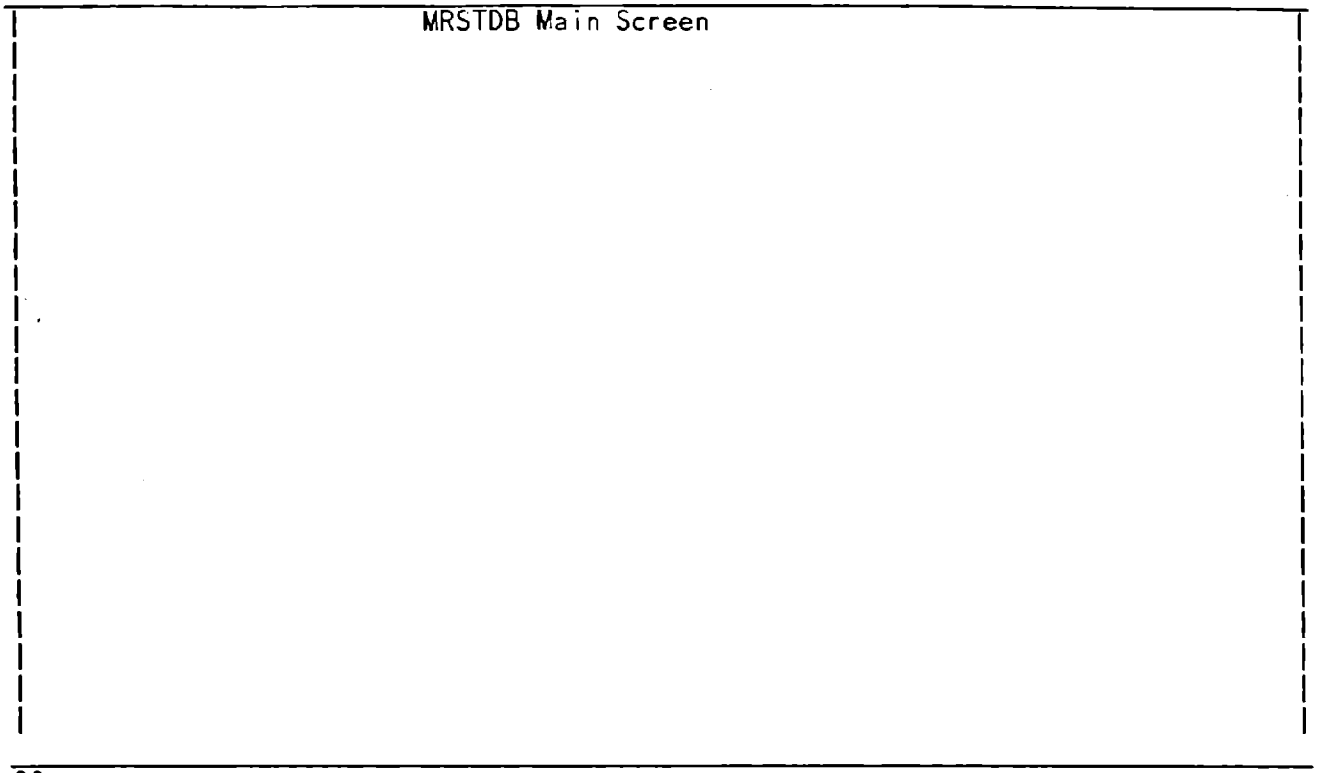
TRIM DESCRIPTIONS

```
Text trim at row '10' and column '8' is:
        'PREVIOUS VALUES:'.
Foreground display color number is: '0'
Special display attributes: None


Text trim at row '0' and column '21' is:
        'MRSTDB Data Entry Screen'.
Foreground display color number is: '0'
Special display attributes: None


Text trim at row '6' and column '28' is:
        'If "Y" answer the following:'.
Foreground display color number is: '0'
Special display attributes: None
```

```
##
```




```
00
```

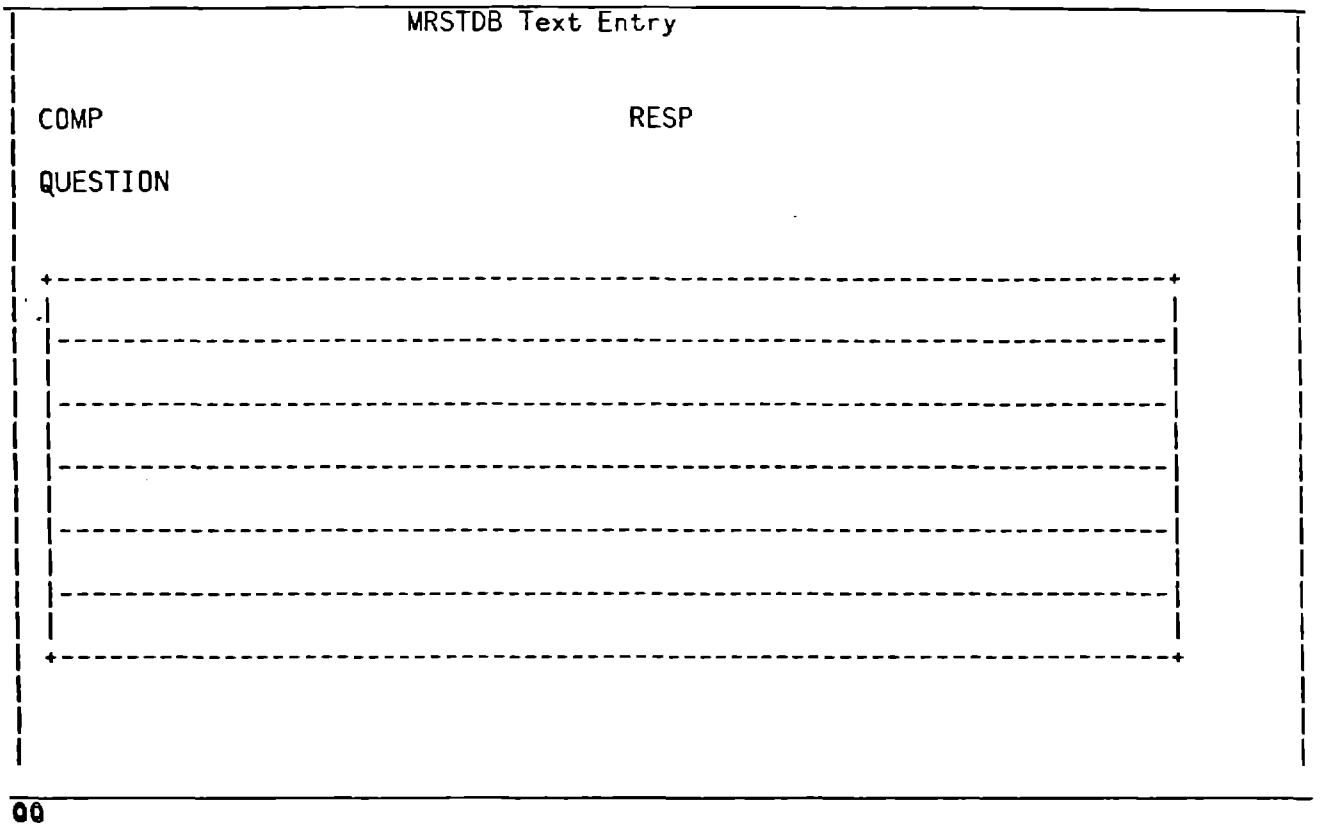
```
Form name: mrstdb_main
Form owner: rchall
Form Display Style is: FullScreen.
Number of columns on screen: 80
Number of lines on screen: 23
Number of fields: 0
Number of trim strings: 1
Date first created: 1991_06_10 18:03:46 GMT
Date last modified: 1991_06_10 18:03:46 GMT


FIELD DESCRIPTIONS




TRIM DESCRIPTIONS


Text trim at row '0' and column '25' is:
        'MRSTDB Main Screen'.
Foreground display color number is: '0'
Special display attributes: None
```



Form name: mrstdb_text_entry
Form owner: rchall
Form Display Style is: FullScreen.
Number of columns on screen: 80
Number of lines on screen: 24
Number of fields: 4
Number of trim strings: 1
Date first created: 1991_06_10 21:42:38 GMT
Date last modified: 1991_06_10 21:42:38 GMT

FIELD DESCRIPTIONS

Field name: comp_name
Field title: COMP
Data type: Non-nullable varchar(30)
Display format: c30
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:

Field name: resp_name

```
Field title: RESP
Data type: Non-nullable  varchar(30)
Display format: c30
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:


Field name: question_id
Field title: QUESTION
Data type: Non-nullable  varchar(30)
Display format: c30
Foreground display color number is: '0'
Special display attributes: Reverse video
Type of field: regular
Default value:
Validation check:
Validation error message:


Table field name: q_lines
Number of rows in table field: 6
Special table field attributes: Highlighting current row enabled, don't display column titl


Column name: q_line
Column title: LINE
Data type: Nullable  varchar(70)
Display format: c70
Foreground display color number is: '0'
Special display attributes: None
Type of field: column in table field
Default value:
Validation check:
Validation error message:



TRIM DESCRIPTIONS


Text trim at row '0' and column '24' is:
        'MRSTDB Text Entry'.
Foreground display color number is: '0'
Special display attributes: None
```

MRSTDB_DATA_ENTRY.OSQ

```
initialize (
  r_count = integer) = {
  r_count := 0;
}
'INPUT' = {
  if restr != 'Y' and restr != 'N' then
    message 'RESTR code must be Y or N.  Transaction rejected.'
      with style=popup;
    resume field restr;
  endif;
  insert into respondent (
    y_n_code_restr_status_of,
    resp_name_of,
    comp_name_respondent,
    no_levels_pres_levels_of,
    size_nbr_size_of,
    span_of_ctrl_pres_of,
    turnover_rate_of)
  values (
    restr,
    resp_name,
    comp_name,
    levels,
    size,
    span,
    turnover);
  commit;
  if restr = 'N' then
    insert into non_restr_resp (
      resp_name_of,
      comp_name_respondent)
    values (
      resp_name,
      comp_name);
    inquire_ingres (r_count = rowcount);
    if r_count > 0 then
      commit;
      message 'Data entry accomplished for non-restructured company.'
        with style=popup;
      resume field comp_name;
    else
      message 'System error inputting data for non-restructured company.'
        with style=popup;
      resume field comp_name;
    endif;
  endif;
  insert into restr_resp (
    resp_name_of,
    comp_name_respondent,
    date_begin_rest_for,
    date_end_restr_for,
    nbr_levels_pre_levels_of,
    span_of_ctrl_pre_ctrl_,
    turnover_rate_pre_t_r_of,
    y_n_code_layoff_incen_of,
    y_n_code_eeo_aa_code_of,
    y_n_code_hrg_chg_code_of,
    y_n_code_retraining_of,
    y_n_code_uni_str_of)
  values (
```

```
    resp_name,
    comp_name,
    begin_d,
    end_d,
    plevels,
    pspan,
    pturnover,
    layoff,
    eeo_aa,
    hrg_chg,
    retraining,
    uni_str);
  inquire_ingres (r_count = rowcount);
  if r_count > 0 then
    commit;
    message 'Data entry accomplished for restructured company.'
      with style=popup;
    resume field comp_name;
  else
    message 'System error inputting data for restructured company.'
      with style=popup;
    resume field comp_name;
  endif;
}
'RETRIEVE' = {
  mrstdb_data_entry := select
    size = size_nbr_size_of,
    restr = y_n_code_restr_status_of,
    levels = no_levels_pres_levels_of,
    span = span_of_ctrl_pres_of,
    turnover = turnover_rate_of
  from respondent
  where
    resp_name_of = :resp_name and
    comp_name_respondent = :comp_name;
  if restr = 'Y' then
    mrstdb_data_entry := select
      begin_d = date_begin_rest_for,
      end_d = date_end_restr_for,
      plevels = nbr_levels_pre_levels_of,
      pspan = span_of_ctrl_pre_ctrl_,
      pturnover = turnover_rate_pre_t_r_of,
      layoff = y_n_code_layoff_incen_of,
      eeo_aa = y_n_code_eeo_aa_code_of,
      hrg_chg = y_n_code_hrg_chg_code_of,
      retraining = y_n_code_retraining_of,
      uni_str = y_n_code_uni_str_of
    from restr_resp
    where
      resp_name_of = :resp_name and
      comp_name_respondent = :comp_name;
  endif;
}
'CHANGE' = {
  if restr != 'Y' and restr != 'N' then
    message 'RESTR code must be Y or N.  Transaction rejected.'
      with style=popup;
    resume field restr;
  endif;
  delete from respondent
```

```
  where
    resp_name_of = :resp_name and
    comp_name_respondent = :comp_name;
  delete from non_restr_resp
  where
    resp_name_of = :resp_name and
    comp_name_respondent = :comp_name;
  delete from restr_resp
  where
    resp_name_of = :resp_name and
    comp_name_respondent = :comp_name;
  insert into respondent (
    y_n_code_restr_status_of,
    resp_name_of,
    comp_name_respondent,
    no_levels_pres_levels_of,
    size_nbr_size_of,
    span_of_ctrl_pres_of,
    turnover_rate_of)
  values (
    restr,
    resp_name,
    comp_name,
    levels,
    size,
    span,
    turnover);
  commit;
  if restr = 'N' then
    insert into non_restr_resp (
      resp_name_of,
      comp_name_respondent)
    values (
      resp_name,
      comp_name);
    inquire_ingres (r_count = rowcount);
    if r_count > 0 then
      commit;
      message 'Data entry changed for non-restructured company.'
        with style=popup;
      resume field comp_name;
    else
      message 'System error inputting data for non-restructured company.'
        with style=popup;
      resume field comp_name;
    endif;
  endif;
  insert into restr_resp (
    resp_name_of,
    comp_name_respondent,
    date_begin_rest_for,
    date_end_restr_for,
    nbr_levels_pre_levels_of,
    span_of_ctrl_pre_ctrl_,
    turnover_rate_pre_t_r_of,
    y_n_code_layoff_incen_of,
    y_n_code_eeo_aa_code_of,
    y_n_code_hrg_chg_code_of,
    y_n_code_retraining_of,
    y_n_code_uni_str_of)
```

```
  values (
    resp_name,
    comp_name,
    begin_d,
    end_d,
    plevels,
    pspan,
    pturnover,
    layoff,
    eeo_aa,
    hrg_chg,
    retraining,
   ' uni_str);
  inquire_ingres (r_count = rowcount);
  if r_count > 0 then
    commit;
    message 'Data entry changed for restructured company.'
      with style=popup;
    resume field comp_name;
  else
    message 'System error inputting data for restructured company.'
      with style=popup;
    resume field comp_name;
  endif;
}
'QUESTIONS' = {
  callframe mrstdb_text_entry (
    mrstdb_text_entry.comp_name := comp_name;
    mrstdb_text_entry.resp_name := resp_name)
}
'QUIT' = {
  return;
}
```

MRSTDB_MAIN.OSQ

```
'UPDATE' = {
     callframe MRSTDB_DATA_ENTRY;
}
'QUIT' = {
     return;
}
```

MRSTDB_TEXT_ENTRY.OSQ

```
initialize (
  r_count = integer,
  cmd = vchar(70)) = {
  r_count = 0;
  create table q_text (q_row varchar(70));
  resume field question_id;
}
'INPUT' = {
  delete from q_text;
  unloadtable q_lines
  begin
  , insert into q_text (
      q_row)
    values (
      q_lines.q_line);
  end;
  commit;
  call system 'sql -s <q_text_input. mrstdb';
  cmd := 'rename q_text.txt ' +
    squeeze(comp_name + resp_name + question_id + '.txt');
  call system cmd;
  insert into question (
    question_id_of,
    comp_name_contains,
    resp_name_contains,
    text_answers)
  values (
    question_id,
    comp_name,
    resp_name,
    squeeze(comp_name+resp_name+question_id+'.txt'));
  commit;
  message 'Text successfully input for question.' with style=popup;
  clear field q_lines;
  resume field question_id;
}
'WITHDRAW' = {
  delete from q_text;
  delete from question
  where
    question_id_of = :question_id and
    comp_name_contains = :comp_name and
    resp_name_contains = :resp_name;
  cmd := 'rename ' +
    squeeze(comp_name + resp_name + question_id + '.txt') +
    ' q_text.txt';
  call system cmd;
  commit;
  call system 'sql -s <q_text_withdraw. mrstdb';
  q_lines := select
    q_line = q_row
  from q_text;
  delete from q_text;
  cmd := 'delete q_text.txt;*';
  call system cmd;
  resume field question_id;
}
'QUIT' = {
  drop table q_text;
  return;
```

}

Q_TEXT_INPUT

```
copy q_text (q_row = c0nl) into 'q_text.txt'
\g
\q
```

Q_TEXT_WITHDRAW

```
copy q_text (q_row = c0nl) from 'q_text.txt'
\g
\q
```

APPENDIX B

MRST Survey Questionnaire

TECHNICAL SPECIFICATION FOR THE SANDIA MANAGEMENT RESTRUCTURE STUDY TEAM (MRST) PROTOTYPE INFORMATION SYSTEM

This appendix contains the survey questionnaire that was used to obtain data from external companies. The MRST Prototype Information System information model (presented in Section 2 of this report) was based exclusively on this survey, as all responses were to be accessible through the system. However, the responses to some questions were specifically requested to be available for analysis and comparison purposes. These "priority questions" are represented as entities in the information model, separate from the "TEXT" entity which encompasses all textual responses.

SANDIA NATIONAL LABORATORIES

RESTRUCTURING STUDY: COMPANY SURVEY EXTERNAL

May 8, 1991

1.0 General Information

1. Company Name

   Size

   General Composition of Staff

   Location

   Key Contact

   Telephone No.

   FAX No.

   Type of Business (Producers and Customers)

2. Recently restructured?

   Yes No

   (If "yes," go to Question #3; if "no," or considering restructuring, go to Question #25.

3. When did you restructure?

2.0 Questions Related to Objectives

For those who have restructured:

4. What were your objectives for restructuring?

(In asking this question, first focus on the objectives identified in the previous question that match the following Sandia objectives; then discuss those not mentioned previously.)

5. How did the restructure:

- Affect project execution
- Affect communication
- Affect flexibility, responsiveness
- Affect empowerment
- Affect cost effectiveness
- Affect teamwork

- Affect leadership

- Affect respect for individuals and their talents

- Affect motivation

- Affect creativity and innovation

6. a. What other mechanisms do you have in place for fostering the above items.

   b. What metrics did you establish to measure success in meeting restructuring objectives?

3.0 Questions Relating to Organizational Structure

7. What was your organization structure before restructuring (organization chart if possible)?

   - Number of Levels

- Generic description of responsibilities of each level
- Span of Control (Average)
- How do you address Project vs. Line. vs. Staff vs. Research

8. How did your structure change?
   - Number of Levels
   - Generic description of responsibilities of each level.

- Span of Control
- How do you address Project vs. Line vs. Staff vs. Research

9. Is the new structure uniform over all lines of business?

   Yes No

   If "no," describe the differences and basis for these differences.

10. What systems (internal) changes did you have to make (i.e., information systems, budget systems, financial/accounting systems, etc.)?

11. What staff support is provided to management?

4.0 Questions Relating to Implementing change

4.1 Description of the Change Process

12. Describe the implementation (transition) plan/process (i.e., time, communication plan, feedback, what had to take place before and after restructuring, etc.).

13. Were consultants used? If yes, who. How effective were they?

4.2 Human Resources

14. Did you offer layoff incentives for the level(s) eliminated?

Yes No

15. Did you retrain the employees occupying the eliminated level(s)?

Yes No

If "yes," briefly describe this training.

16. How were these people redeployed?

17. Did this restructuring have an impact on your EEO/AA profile?

Yes No

18. How did you manage reassignments to avoid adversely impacting your EEO/AA program?

19. Did the restructuring require a change in your hiring program?

Yes No

If "yes," describe the changes.

20. Give a brief description of your reward and recognition system and describe any changes you made.

4.3 Evaluating the Process

21. How successful has the plan been?

22. What would you do differently if you had it to do over again?

23. What next, if anything?

5.0 Other

24a. What was your turnover rate before restructuring?

After restructuring?

24b. Is there anything else that is important for us to know about restructuring that we have not discussed?

6.0 Questions Relating to Objectives

For those who have not restructured:

25. How does your structure:

- Achieve excellence in project execution

- Foster good communication

- Foster flexibility, responsiveness

- Promote empowerment of everyone

- Foster cost effectiveness

- Foster teamwork

- Foster leadership

- Foster respect for individuals and their talents

- Foster motivation

- Encourage creativity and innovation

26. What special mechanisms do you have in place for fostering the above items?

27. What metrics do you use to evaluate the above?

7.0 Questions Relating to Organizational Structure

28. What is your organization structure (organization chart if possible)?

- Number of levels
- Generic description of responsibilities of each level

- Span of Control (Average)
- How do you address Project vs. Line vs. Staff vs. Research

29. How would you improve your current management structure?

30. Is there anything else about structure that we did not discuss?

APPENDIX C

MRST Prototype Information System User's Guide

TECHNICAL SPECIFICATION FOR THE SANDIA MANAGEMENT RESTRUCTURE STUDY TEAM (MRST) PROTOTYPE INFORMATION SYSTEM

This appendix contains a user's guide to the primary (restructure analysis) application of the MRST Prototype Information System. Each of the three functional areas (Basic Restructuring Information, Comparison of Respondents, Examination of Survey Information) are briefly described. For each screen, a list of available functions and an explanation of each function's purpose is presented. This is not intended as a tutorial for the use of the system, but rather as a document of operations to supplement the reader's understanding of the screen definitions.

This document describes the functions available in the Management Restructure Study Team (MRST) Prototype Information System. To execute any of the functions discussed, use the mouse to position the cursor on the desired button or menu item, and click the left mouse button.

Main Menu Screen

This screen allows the user to enter any of three functional areas: Basic Restructuring Information, Comparison of Respondents, or Examination of Survey Information. These areas are discussed below. To exit the system, click on Quit.

Basic Restructuring Information Functional Area

This area consists of a single screen that allows the user to access certain basic data from a respondent that has restructured. This information includes the dates that restructuring began and was completed, the company size, as well as the "Yes" or "No" answers to several survey questions dealing with the impact of the restructuring process.

Basic Restructuring Information Screen

SELECT Function: Allows the user to select data for a particular respondent. A menu will appear containing a list of respondent names. To select a name, click on it, then click the SELECT button. The screen will display the data for that respondent.

COMPARE Function: Displays another Basic Restructuring Information screen so that the user can view the data for multiple respondents simultaneously.

CANCEL Function: Return to Main Menu screen.

Comparison of Respondents Functional Area

This area also consists of a single screen that allows the user to display numerical data pertaining to organizational structures of several respondents simultaneously for comparison purposes. It also allows the user to display this data for a single restructured respondent for pre- and post-reconstruction periods in order to analyze the effects of the reconstruction process.

Comparison of Respondents Screen

SELECT Function: Allows the user to select a respondent name from the selection menu. Repeated use of this function will list all selected respondents on the screen for comparison.

NUMBER OF LEVELS
Function: Displays the number of levels from pre- to post-restructured periods in the Previous and Present columns, respectively, for each respondent selected.

SPAN OF CONTROL
Function: Displays the values for span of control.

TURNOVER RATE
Function: Displays the values for turnover rate.

CANCEL Function: Return to Main Menu screen.

Examination of Survey Information Functional Area

This area allows the user to access the actual responses provided by external companies, including text as well as graphical data.

Survey Information Menu Screen

This screen allows the user to select any of the three screens described below. To return to the Main Menu screen, click on Cancel.

Text Examination Screen

SELECT Function: Allows the user to select a respondent name from the selection menu.

KEYWORD Function: Displays a menu of questions indexed by important keywords. When one is selected, the respondent name and question number will be returned to the Text Examination screen.

QUESTION Function: Displays the text corresponding to the respondent name and question number on the screen.

COMPARE Function: Displays another Survey Information Menu screen so that the user can view the responses from multiple companies simultaneously.

CANCEL Function: Return to Survey Information Menu screen.

Chart Examination Screen

SELECT Function: Allows the user to select a respondent name from the selection menu.

PRESENT Function: Displays the organization chart corresponding to the selected respondent's present organization structure.

PREVIOUS Function: Displays the organization chart corresponding to the selected respondent's organization structure prior to restructuring.

COMPARE Function: Displays another Survey Information Menu screen so that the user can examine multiple charts simultaneously.

CANCEL Function: Return to Survey Information Menu screen.

Page Examination Screen

SELECT Function: Allows the user to select a respondent name from the selection menu.

OK Function: Displays the image corresponding to the respondent name and page number on the screen.

COMPARE Function: Displays another Survey Information Menu screen so that the user can view two pages simultaneously.

CANCEL Function: Return to Survey Information Menu screen.

Distribution

1	5	Dan Hartley
1	2800	Arlyn Blackwell
1	2810	Melissa Smartt
1	2816	Karl Wiegandt
1	2816	Irene Thurston
1	2818	John Sharp
1	2818	Olin Bray
1	2818	Larry Claussen
1	2818	Lyle Davis
1	2818	Donna Eaton
1	2818	Paul Flores
10	2818	Ronald Hall
1	2818	E. Janet Klamerus
1	2818	John Orman
1	2818	Gary Rivord
5	2818	Tim Wyatt
1	2830	George Urish
1	2850	Tom Cannon
1	5510	Don McCoy
1	5512	Jim Lenhart
1	7730	Judith Mead
1	7020	P. W. Dean
1	8523-2	Central Technical Files
5	3141	S. Landenberger
8	3145	Document Processing for DOE/OSTI
3	3151	G. C. Claycomb