LOAD DOCUMENT

PHOTOGRAPH THIS SHEET

AD-A228 138

DTIC FILE COPY

LEVEL

INVENTORY

WRDC-TR-90-8026

DOCUMENT IDENTIFICATION

NOV 1990

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

DTIC ELECTE
OCT 25 1990
S E D

DATE ACCESSIONED

ACCESSION FOR

NTIS	GRA&I
DTIC	TRAC
UNANNOUNCED	
JUSTIFICATION	

BY
DISTRIBUTION/
AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL

A-1

DISTRIBUTION STAMP

DATE RETURNED

90

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM 70A
JUN 90

DOCUMENT PROCESSING SHEET

PREVIOUS EDITIONS MAY BE USED UNTIL
STOCK IS EXHAUSTED.

LOAN DOCUMENT

WRDC-TR-90-8026

GEOMETRIC MODELING APPLICATIONS INTERFACE PROGRAM

SCHEMA MANAGER USER'S MANUAL

United Technologies Corporation
Pratt and Whitney
Government Products Division
P.O. Box 9600
West Palm Beach, Florida 33410-9600

NOVEMBER 1990

Final Report For Period August 1985 – March 1989

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.
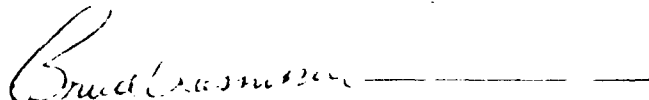
This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Charles Gilman
Project Manager

Walter H. Reimann, Chief
Computer-Integrated Mfg. Branch

FOR THE COMMANDER

BRUCE A. RASMUSSEN
Chief, Integration Technology Division
Manufacturing Technology Directorate

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, WPAFB, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
	   Approved for public release; distribution
	   is unlimited

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
   FR 20358
	5. MONITORING ORGANIZATION REPORT NUMBER(S)
	   WRDC-TR-90-8026

6a. NAME OF PERFORMING ORGANIZATION
   United Technologies Corporation
   Pratt & Whitney
   Government Products Division
	6b. OFFICE SYMBOL
	(If applicable)
	
	(P&W)
		7a. NAME OF MONITORING ORGANIZATION
		Wright Research and Development Center
		Manufacturing Technology Directorate
		(WRDC/MTI)

6c. ADDRESS (City, State and ZIP Code) P.O. Box 9600 West Palm Beach, Florida 33410-9600		7b. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB OH 45433-6533
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-85-C-5122

8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
11. TITLE (Include Security Classification) GEOMETRIC MODELING APPLICATIONS INTERFACE PROGRAM	78011F	MTPI	06	74

12. PERSONAL AUTHOR(S)
   R. Helldoerfer, C. Van Wie, D. Emmerson

13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 1 Aug 85 TO 31 Mar 89	14. DATE OF REPORT (Yr., Mo., Day) November 1990	15. PAGE COUNT 183

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Geometric Modeling Applications Interface Program Product Definition Data Interface Turbine Blades and Disks

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

This User Manual provides a Schema Manager software guide developed under the Computer Integrated Manufacturing program known as GMAP (Geometric Modeling Applications Interface Program), U.S. Air Force Contract F33615-85-C-5122. It includes a description of the menus available to access the create, update, review, and report capabilities generated by the Schema Manager.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED X SAME AS RPT. DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL David Judson	22b. TELEPHONE NUMBER (Include Area Code) (513) 255-7371	22c. OFFICE SYMBOL WRDC/MTI

18. Subject Terms (Continued)

Product Life Cycle
Engineering
Manufacturing
Interface
Exchange Format
CAD
CAM
CIM
IBIS
RFC
System Translator
Schema Manager
Model Access Software
Name/Value Interface

FOREWORD

This User's Manual describes work performed under Air Force Contract F33615-85-C-5122, Geometric Modeling Applications Interface Program (GMAP), covering the period 1 August 1985 to 31 March 1989. This User's Manual provides a Schema Manager software guide developed under this contract which is sponsored by the Computer Integrated Manufacturing Branch, Materials Laboratory, Air Force Systems Command, Wright Air Force Base, Ohio 45433-653? The GMAP Project Manager for the Air Force is Mr. Charles Gilman.

The primary contractor is Pratt & Whitney, an operating unit of United Technologies Corporation. Mr. Richard Lopatka is managing the GMAP project at Pratt & Whitney. Ms. Linda Phillips is the Program Integrator. Mr. John Hamill is the Deputy Program Manager.

McDonnell Aircraft Company is the subcontractor responsible for the Retirement for Cause interface task. Mr. Jerry Weiss is the GMAP Program Manager at McDonnell Aircraft, and Mr. Herb Ryan is the Deputy Program Manager.

NOTE: The number and date in the upper right corner of each page in this document indicate that it has been prepared in accordance to the ICAM CM Life Cycle Documentation requirements for a Configuration Item (CI).

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

R20358/3

SECTION 1

SCOPE

1.1 IDENTIFICATION

This User's Manual provides a Schema Manager software guide for the Geometric Modeling Applications Irterface Program (GMAP), Project 5602, Air Force Contract F33615-85-C-5122.

The PDDI Program, Project 5601, Contract F33615-82-C-5036, provided the foundation for GMAP. The goal of the Product Definition Data Interface Program was to develop computerized communications to replace the traditional engineering drawings used between engineering and manufacturing environments. GMAP refines and extends this product definition data to include data needed to support applications throughout the entire product life cycle. The overall objectives of GMAP are to identify, establish, and demonstrate the use of computerized product definition data in the engineering, manufacturing, and logistics support of complex structural components.

1.2 FUNCTIONAL SUMMARY

The three major functions of the Schema Manager software include the ability to:

- Model a concrete conceptual schema

- Transform a concrete conceptual schema into a physical schema suitable for the Working Form (WF) of the Model Access Software (MAS)

- Generate subschema forms of the physical schema for use by application programs at the time of compiling and/or at the time of running.

1.3 APPROACH

This User's Manual consists of seven sections and two appendices as described below.

Section 1 — Overview of this document.

Section 2 — Related documents and lists applicable to the Schema Manager software.

Section 3 — A high level introduction to the Schema Manager system.

Section 4 — A description of the menus available to access the create, update, review, and report capabilities of the interactive Schema Manager software.

Section 5 — Explanations and examples of reports and files generated by the Schema Manager.

Section 6 — Keywords used to create a new schema or add to an existing schema.

Section 7 — The menus for querying a CAD/CAM model via the Data Dictionary entity definitions.

Appendix A — A description of the routine that locates and returns a copy of a specific entity or class definition.

Appendix B — A definition of the metamodel working form used by the Schema Manager.

SECTION 2

REFERENCES

2.1 REFERENCE DOCUMENTS

The following technical reports, specifications, standards, and other documents have been referred to or are relevant to this Schema Manager User's Manual.

2.1.1 Military

Integrated Computer Aided Manufacturing (ICAM) Architecture, Vol. 4, Function Modeling Manual (IDEF0), USAF Report No. AFWAL-TR-81-4023, June 1981.

Integrated Computer Aided Manufacturing (ICAM) Architecture, Vol. 5, Information Modeling Manual (IDEF1), USAF Report No. AFWAL-TR-81-4023, June 1981.

Integrated Computer Aided Manufacturing (ICAM) Documentation Standards, IDS 150120000C, September 1983.

PDDI System Specification, Product Definition Data Interface (PDDI) Project 5601, Contract F33516-82-5036, July 1984.

PDDI System Specification-Draft Standard, Product Definition Data Interface (PDDI), Project 5601, Contract F33516-82-5036, July 1984.

Information Modeling Manual IDEF-Extended (IDEF1X) Integrated Information Support System (IISS), ICAM Project 6201, Contract F33615-80-C-5155, December 1985.

Interim Technical Report No. 1 (ITR560240001U)
"Geometric Modeling Applications Interface Program" February 1986
(Period 1 August 1985 — 31 October 1985).

Interim Technical Report No. 2 (ITR560240002U)
"Geometric Modeling Applications Interface Program" May 1986
(Period 1 November 1985 — 31 January 1986).

Geometric Modeling Applications Interface Program (GMAP) Scoping Document, CI SD560240001U, May 1986.

Interim Technical Report No. 3 (ITR560240003U)
"Geometric Modeling Applications Interface Program" August 1986
(Period 1 February 1986 — 30 April 1986).

Interim Technical Report No. 4 (ITR560240004U)
"Geometric Modeling Applications Interface Program" November 1986
(Period 1 May 1986 — 31 July 1986).

Geometric Modeling Applications Interface Program (GMAP) Needs Analysis
Document, CI NAD560240001U, November 1986.

Interim Technical Report No. 5 (ITR560240005U)
"Geometric Modeling Applications Interface Program" January 1987
(Period 1 August 1986 — 31 October 1986).

Geometric Modeling Applications Interface Program (GMAP) System
Requirements Document, CI SRD560240001U, February 1987.

Geometric Modeling Applications Interface Program (GMAP) State of the Art
Document, CI SAD560240001U, March 1987.

Interim Technical Report No. 6 (ITR560240006U)
"Geometric Modeling Applications Interface Program" May 1987
(Period 1 November 1986 — 31 January 1987).

Geometric Modeling Applications Interface Program (GMAP) System
Specification (Volumes I-IV), CI SS560240001U, July 1987

Interim Technical Report No. 7 (ITR560240007U)
"Geometric Modeling Applications Interface Program," August 1987
(Period 1 February 1987 — 30 April 1987).

Geometric Modeling Applications Interface Program (GMAP) System Design
Specification, CI SDS560240001U, November 1987.

Geometric Modeling Applications Interface Program (GMAP) to Retirement for
Cause Interface Development Specification, CI DS560240011U, November 1987.

Geometric Modeling Applications Interface Program (GMAP) to Integrated
Blade Inspection System Interface Development Specification, CI
DS560240021U, November 1987.

Geometric Modeling Applications Interface Program (GMAP) to Retirement for
Cause Interface As-designed Product Specification, CI PS560240011U,
December 1987.

Geometric Modeling Applications Interface Program (GMAP) to Retirement for
Cause Interface Unit Test Plan, CI UTP560240011U, December 1987.

Interim Technical Report No. 8 (ITR560240008U)
"Geometric Modeling Applications Interface Program," December 1987
(Period 1 May 1987 — 31 July 1987).

Interim Technical Report No. 9 (ITR560240009U)
"Geometric Modeling Applications Interface Program," March 1988
(Period 1 August 1987 — 31 October 1987).

Geometric Modeling Applications Interface Program (GMAP) System Test Plan, CI STP560240001U, March 1988.

Product Definition Data Interface (PDDI)/Geometric Modeling Applications Interface Program (GMAP) Deliverables Roadmap Document, March 1988.

Geometric Modeling Applications Interface Program (GMAP) to Integrated Blade Inspection System Interface Unit Test Plan, CI UTP560240021U, March 1988.

Geometric Modeling Applications Interface Program (GMAP) to Integrated Blade Inspection System Interface As-designed Product Specification, CI PS560240021U, March 1988.

Geometric Modeling Applications Interface Program (GMAP) System Component As-designed Product Specification, CI PS560240031U, March 1988.

Interim Technical Report No. 10 (ITR560240010U)
"Geometric Modeling Applications Interface Program," August 1988
(Period 1 November 1987 — 31 January 1988).

Interim Technical Report No. 11 (ITR560240010U)
"Geometric Modeling Applications Interface Program," August 1988
(Period 1 February 1988 — 30 April 1988).

Geometric Modeling Applications Interface Program (GMAP) to Retirement for Cause Interface User Operator Manual, CI U/OM560240011U, August 1988.

Interim Technical Report No. 12 (ITR560240012U)
"Geometric Modeling Applications Interface Program" October 1988
(Period 1 May 1988 — 31 July 1988).

Geometric Modeling Applications Interface Program (GMAP) to Retirement for Cause Interface Unit Test Report, CI UTR560240011U, November 1988.

Geometric Modeling Applications Interface Program (GMAP) to Integrated Blade Inspection System Interface Unit Test Report, CI UTR5602421U, November 1988.

Geometric Modeling Applications Interface Program (GMAP) System Translator User Manual, CI UM560240021U, November 1988.

Geometric Modeling Applications Interface Program (GMAP) to Retirement for Cause Interface As-built Product Specification, CI PS560240012U, February 1989.

Geometric Modeling Applications Interface Program (GMAP) to Integrated Blade Inspection System Interface As-built Product Specification, CI PS560240022U, February 1989.

Geometric Modeling Applications Interface Program (GMAP) System Components Operator's Manual, CI OM560240001U, February 1989.

Geometric Modeling Applications Interface Program (GMAP) to Integrated Blade Inspection System Interface User/Operator Manual, CI U/OM560240021U, February 1989.

Geometric Modeling Applications Interface Program (GMAP) Schema Manager User's Manual, CI UM560240011U, February 1989.

Interim Technical Report No. 13 (ITR560240013U)
"Geometric Modeling Applications Interface Program" February 1989
(Period 1 August 1988 — 31 October 1988).

2.1.2 Commercial

A Practical Guide to Splines, C. de Boor, Applied Mathematical Sciences, Vol. 27, Springer-Verlag.

Design of Database Structures, T. J. Teorey and J. P. Fry, Prentice-Hall, Inc., Englewood Cliffs, N.J.

Differential Geometry of Curves and Surfaces, M. P. de Carmo, Prentice-Hall, Inc., 1976.

IDEF1X Readers Reference, D. Appleton Company, December 1985.

Identification of Product Definition Data in a Manufacturing Enterprise — A Case Study. R. Lessard, United Technologies Research Center and R. Disa, Pratt & Whitney, March 1986.

Use of Product Models in a CIM Environment, D. Koziol Emmerson and K. Perlotto, Pratt & Whitney, March 1987.

Technical Issues in Product Data Transfer, Richard Lopatka, Pratt & Whitney, September 1987.

Implementation of GMAP Technologies for Logistic Support Applications, Donald L. Deptowicz, Pratt & Whitney, January 1988.

Barriers to PDES Approval, Anthony Day, Sikorsky, and Richard Lopatka, Pratt & Whitney, April 1988.

PDD: Implementation Issues, Diane Emmerson and Priscilla Blasko, United Technologies Corporation, Proceedings of AUTOFACT '88, October 1988.

2.1.3 Standards Organizations

ANSI Y14.5M, Dimensioning and Tolerancing.

"The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems," **Information Systems**, Vol. 3, pp. 173-191, 1973.

The Second Draft Report of the Ad Hoc Committee on the Content and Methodology of the IGES Version 3 (The Second PDES Report), K. Brauner and D. Briggs, November 1984.

EXPRESS — A Language for Information Modeling, ISO, TC184/SC4/WG1, January 1986.

The STEP File Structure, ISO, TC184/SC4/WG1, January 1987.

Mapping from EXPRESS to Physical File Structure, ISO, TC184/SC4/WG1, January 1987.

2.2 TERMS AND ACRONYMS

A glossary of terms frequently used in GMAP which may be included in this Schema Manager User's Manual is provided below. A list of acronyms and abbreviations used in GMAP is also included in this section.

2.2.1 Terms Used in GMAP

Accept/Reject/Incomplete Notice — A display on the cell computer that indicates the final status of the engine disk.

Accept	= Acceptable within tolerance specified by engine manufacturer
Reject	= Rejected because of flaw(s) outside the range of acceptable tolerances
Incomplete	= Part cannot be inspected

Access Software — A set of routines for creating, managing, and querying an incore Working Form model.

Administrative Data — The set of data which define the part characteristics, materials, approvals, and other miscellaneous information necessary to the production environment.

Angular — An angular size tolerance is used to tolerance the size of an angular feature independent of its angular location along an arc.

Application — A method of producing a specific result.

Application Request — A request initiated by an application program, either through batch or interactive processing, which will interrogate the model through the PDDI Access Software to obtain or operate on specific information regarding the model and its components or elements.

Application Requested Data — The data which fulfill the application's original request and which is in the proper format and readable by the application.

Architecture — A design or orderly arrangement.

ASCII — American Standard Code for Information Interchange.

As-Is — The present condition.

Attribute — A quality of characteristics element of any entity having a name and a value.

B-Spline — A spline defined by a control polygon, B-spline basis functions, and an associated knot vector. A Bezier curve is a special case of a B-spline; a nurb is the most general case of a B-spline.

Bezier Curve — A type of curve defined by a set of vertices called a control polygon and a set of basis functions. The basis functions are known as Bernstein polynomials. K vertices define a curve of order K-1.

Binding — Establishing specific physical references to data structures for an application program; may be performed at compile time or at run time.

Blend — A smooth, continuous transition from one surface to another.

Body of Revolution (BOR) Representation — A topology in which an object is represented as the volume swept by a curve rotated about a line. This is a boundary representation in which the curve represents the surface area of the object.

Boundary Representation — A topology imposed on 3-D geometric entities to yield a general solid model. That model describes an object by describing its boundary area.

Bounded Geometry — Geometry that has limits defined by its mathematical domain or range.

Calibration Block Parameters (Scale Factors) — Nondestructive test parameters used to adjust a specific cell. These parameters are obtained from the calibration blocks located at each cell.

Circumferential — A circumferential tolerance specifies the tolerance zone within which the average diameter of a circular feature must lie. The average diameter is the actual circumference divided by pi (3.14159). A circumferential tolerance is a specific example of a peripheral or perimeter tolerance for a general curve.

Class — A collection of entities that are alike in some manner.

CLIST — IBM Command lists.

Composite Curve — A group of curve segments that are C^0 continuous.

Compound Feature Representation — An enumerative feature representation in which at least one component is itself a feature. For example, a bolt hole circle might be represented as a list of individual hole features.

Concentricity (Generic) — A concentricity tolerance specifies a cylindrical tolerance zone within which the axis of a feature must lie, where the axis of the zone coincides with the axis of the datum.

Conceptual Schema — Formally specified global view that is processing independent, covering information requirements and formulation of independent information structures. A neutral view of data, usually represented in terms of entities and relations.

Conic — A quadratic curve represented in the most general case by the equation:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0.$$

A conic may be a circle, line, ellipse, parabola, or a hyperbola depending on the coefficients, A, B, C, D, E, and F.

Constraints (Generic) — An assertion to explicitly specify data meaning or semantics.[1]

Context-Free Grammar — The syntax of the language gives a precise specification of the data without interpretation of them.

Constituent — A specific instance of an entity that is used in the defination of some other entity.

Curve Segment — A subset of an underlying curve defined by a curve and bounded by two end points.

Data Dictionary — A catalog of all data elements in a design, giving their name, definition, format, source, and usage. May also include data types and value limits.

Defining Airfoil Sections — A planar or conical section that depicts an airfoil profile. Defining airfoil sections are those that meet aerodynamic requirements. Other intermediate sections are added for Manufacturing purposes.

Dimension — A part dimension is a quantifiable value expressing size, form, or location.

Domain — The set of values permissible in a given context.

Dynamic Allocation — The allocation (and deallocation) of memory resources as required by the application. The opposite is **static** allocation where a fixed size segment of memory is available to the application.

Eddy Current Cell — Hardware used to perform an Eddy current inspection operation (surface flaws).

Eddy Current Inspection — An inspection method used to detect internal potential flaws on a disk. It is based on the principle of sending electromagnetic signals to a target area on a part and detecting/interpreting reflection (Eddy current) from the target.

---

[1] T.J. Teorey and J.P. Fry, Design of Database Structures, 1st edition, Prentice-Hall, Inc., Englewood Cliffs, N.J., p. 463.

Eddy Current Scan Plan — An interpreter code program controlling the Eddy current inspection of a particular geometry.

Eddy Current/Ultrasonic Flaw Data Printout — A printout containing size and location information about specific flaw(s) (both critical and noncritical) associated with a particular part.

Entity — A description of a person, place, or thing, about which information is kept.

External Reference — A reference to some quantity of data that exists somewhere outside the scope of the immediate body of information.

Feature — A part feature in the dimensioning and tolerancing context is a feature in the sense of ANSI Y14.5M, that is, a physical component portion of a part, such as a surface, hole, slot, and so on, that is used in a tolerancing situation. In the dimensioning and tolerancing context, a feature consists of individual or groups of basic shape elements used to define the physical shape of an item. This general dimensioning and tolerancing use of features is to be distinguished from Features. The word "features" alone implies dimensioning and tolerancing features. The term "form feature" is described below.

Feature Pattern — A geometric pattern of occurrences of similar form features, for example, a circular pattern of scallops, a rectangular array of holes.

Feature Representation (Generic) — A description of a form feature within the context of a geometric model.

Feature Type — A name applied to a form feature that is suggestive of its shape and size, for example, hole, slot, web.

Feature of Size (Generic) — A feature of size provides a geometric location capable of being referenced for use with datums and tolerances. A feature of size can be a GMAP feature, or other referenceable shape elements of a part model that are symmetric about a point, line, plane, axis, curve, and so on. When a feature of size is used in a relationship with a tolerance or datum, its feature of symmetry is the implied reference.

Flat Pattern Representation (Extrusion Representation) — A topology in which an object is represented as the volume swept by a planar polygon moving in a direction normal to its plane. The polygon may have internal polygon represent the surface area of the object.

Flaw Characteristics — Location, length, width, depth, and nondestructive test parameters associated with a specific flaw.

Flaw Data Packet — Packet containing nonevaluated flaw data. Note that the packet can contain zero flaws.

Flaw Orientation — The direction of the major characteristic of the flaw with respect of the part coordinate system.

Flaw Suspect Location — The coordinate location of a possible flaw detected during a survey mode inspection (six-axis position of ultrasonic cell, seven-axis position of Eddy current cell).

Form Feature — A portion of a part's geometry that is useful to regard as an entity. In a boundary representation context, this is a subset of the part's surface area.

Form Tolerance — Form tolerances are used to control the form of model features. A form tolerance specifies the amount that an actual features form may vary from nominal. Form tolerance include straightness tolerance, flatness tolerance, roundness/circularity tolerance, cylindricity tolerance, perpendicularity tolerance, parallelism tolerance, angularity tolerance, profile-of-a-line tolerance, profile-of-a-surface tolerance, circular-runout tolerance, true-direction tolerance, and mismatch tolerance.

Functionality — (1) To show that the configuration item has fulfilled the specified requirements. (2) The receiving and sending systems can operate on the entity in the same manner with the same results within a pre-defined tolerance.

Function Modeling — A description of a system in terms of a hierarchy of functions or activities, each level decomposing higher ones into greater detail. Functions are named by verbs; nouns related are declared as inputs, controls, outputs, and mechanisms.

Geometric Element (Generic) — An instance of a geometric entity.

Geometric Group — A group of geometric entities with a name.

Geometric Model — A part description in terms of its underlying geometric elements. The model may be a wireframe, surface, or solid model.

Geometric Pattern — A circular or rectangular pattern of geometric entities.

GMAP Feature — A GMAP feature consists of a useful collection (for some application) of one or more part faces that, because of utility, are regarded as a single thing. The faces do not necessarily appear in the geometric model of the part; the feature may be described (represented) without being realized in the geometric model. GMAP features are analogous to form features. (Refer to **Features** data needs, above.)

Group Technology Code — An alphanumeric string identifying significant characteristics of a product, enabling group technology applications. Also known as Part Classification Code.

Include File — PASCAL source code from another file or library included on the compilation of a PASCAL source file.

Input Data — That information which the application needs to supply in order to interrogate or operate on the model. These data may assume only these forms prescribed by the PDDI Access Software specification.

Inspection Cycle — A period for which nondestructive testing inspection requirements are defined.

Inspection Cycle Zone — An entity that is composed of a unique combination of zone and inspection cycle.

Inspection Module Operator — Refers to personnel operating RFC cell(s).

Instrument Setting Adjustments — Nondestructive testing parameter adjustments automatically accomplished via pre- and post-calibration operations. These adjustments have to be accomplished within a predetermined tolerance.

Internal Flaw — A subsurface anomaly.

Internal Flaw Major Characteristic — A vector determined by an agreed upon method.

> Example (1): The vector of greatest magnitude from the centroid to a boundary of the anomaly.

> Example (2): A vector representing the major axis of the minimum ellipsoidal envelope encompassing the anomaly.

Internal Flaw Tolerance — A unique combination of:

    (a)   Internal flaw orientation range.
    (b)   Serviceable internal flaw tolerance limits.
    (c)   Repairable internal flaw tolerance limits.

Internal Flaw Tolerance Limit — A unique combination of:

    (a)   Maximum diameter.
    (b)   Maximum depth below surface.
    (c)   Maximum thickness.

Interpreted Request — Input data which have been appropriately modified to conform to the PDDI Access Software's internal data representation so that they may be further processed.

Key Attribute — An attribute or combination of attributes having values that uniquely identify each entity instance.[2]

Laminates Representation (Generic) — A topology in which an object is represented as layers of flat material of known thickness.

Location Tolerance — Location tolerances specify the allowable variation in position of model features. Location tolerances include various forms of position tolerancing conventions. These are (true) position, concentricity, alignment, rectilinear location, and angular location.

Logistics Support — The function of procuring, distributing, maintaining, replacing, and repairing material in support of a delivered product.

Machine Coordinate Positions — The probe location with respect to machine coordinates.

---

[2] *Integrated Computer Aided Manufacturing (ICAM) Architecture*, Vol. 5, Information Modeling Manual (IDEF1), USAF Report NO. AFWAL-TR-81-4023, June 1981, p. 212.

Machine Preset Data — Machine coordinate adjustments automatically accomplished via pre- and post-calibration operations. These adjustments have to be accomplished within predetermined tolerance.

Metadata — Data about data. Defines the physical schema and record formats of the part data.

Metamodel — A body of data that defines the characteristics of a data model or structure.

Model — A collection of PDD that is transferable, displayable, accessible, and equivalent to a Part. The internal representation of the application data, as initiated and organized by the user. The model is also referred to as the Working Form.

Model Network Definition — The set of rules and definitions which outline in detail the data structure whereby higher order entities may be composed of lower order entities, or constituents, and the lower order entities may be constituents of one or more higher order entities.

Native System — The PDD and applications in a format that is unique to the database of a CAD system.

Nonconstructive Feature Representation (Explicit Feature Representation) — A feature representation that at least partially depends on a declaration that a face, or portion of a face, is "in" the feature.

Nondestructive Testing Parameters — Parameters used by the Eddy current and ultrasonic instruments (examples: amplitude, phase angle, gain, threshold, and so on).

Nondestructive Testing Personnel — Personnel responsible for the generation of scan plans and derivation of applicable nondestructive testing instrument settings used in the scan plans.

Nonshape Data — Produce definition data that cannot be represented by shape elements.

Normal Forms — Conditions reflecting the degree of refinement and control over the relationships and entities in an information model.

Numerical Control Program (Complete and Proposed) — Set of program instructions used to generate a probe path.

Orientation Range — An envelope in which the major flaw characteristic must lie.

Parse — The process of analyzing input strings (records) to identify fields and to verify that the data have a valid format.

Part Blueprint — A blueprint provided by the engine manufacturer of a particular F100 engine disk.

Physical Schema — Internal representation of data; the computer view that includes stored record format and physical ordering of stored records.

Polynomial Spline — A parametric spline of order 1, 2, or 3 defined by a set of N+1 points. The spline is CX, CY, or CZ continuous and defined by coefficients such that:

$$x(i) = AX_{(i)} + BX_{(i)} * S + CX_{(i)} * S**2 + DX_{(i)} * S**2$$

$$y(i) = AY_{(i)} + BY_{(i)} * S + CY_{(i)} * S**2 + DY_{(i)} * S**2$$

$$z(i) = AZ_{(i)} + BZ_{(i)} * S + CZ_{(i)} * S**2 + DZ_{(i)} * S**2$$

and a parameter space $(T_0, T_1, ... T_n)$

where

$$T_{(i)} < = u < = T_{(i+1)}$$

$$S = u - T_{(i)}$$

Position Tolerance — A position tolerance (true position) specifies a tolerance zone within which the feature may vary in any direction.

Post-processor — A phase of the translator where data are received from the Exchange Format and are converted to the Working Form.

Pre-processor — A phase of the translator where data are taken from the Working Form and are converted to the Exchange Format.

Primitive Constructive Feature Representation (Generic) — A constructive representation that is noncompound and that does not incorporate another feature. Such a representation must consist solely of overt construction information. Representation of a through hole by centerline and diameter is an example.

Probe Blueprint — Blueprint of Eddy current probe supplied by the probe manufacturer.

Product Definition Data — Those data "explicitly representing all required concepts, attributes, and relationships" normally communicated from Design throughout Manufacturing and Logistics Support. The data include both shape and nonshape information required to fully represent a component or assembly so that it can be analyzed, manufactured, inspected, and supported. They enable downstream applications, but do not include process instructions. These data are not always finalized at the design release; the manufacturing process can also add to the product model or generate derived manufacturing product models.

Product Life Cycle — Includes design, analysis, manufacturing, inspection, and product and logistics support of a product.

Product Model — A computer representation of a product.

Product Support — The function that interprets customer requests for information and can provide the technical responses to the customer in the form of technical orders and instructions.

R30044/3

Proprietary Part Flaw Data — Formatted dataset containing proprietary data defining size(s), maximums, and location(s) of critical flaw(s) (dimensional and locational tolerance).

RAW.0 File — A data file that uses a bi-cubic patch surface representation to define the surfaces of an airfoil.

Ready Status — Go/No-Go decision.

Relation — A logical association between entities.[3]

Remount Decision — Decision to remount an engine disk.

Replicate Feature Representation (Generic) — A description of a feature as being identical to another feature except for location. Mathematically, a replicate feature representation consists of the identification of another (necessarily constructive) feature plus a transformation.

Robot Initialization Parameters — A set of nondestructive testing parameters used to initialize the robot on an Eddy current or ultrasonic cell.

Rotational Sweep — A sweep in which the swept curve is rotated about a line (the "centerline" of the sweep).

Ruled Surface (Generic) — A surface defined by a linear blend of two curves.

Run System — The Translator subpackage which provides the communication interface between the user and the pre/post-processors.

Run-Time Subschema — A subset of the data dictionary information used at run-time by the access software to provide field data and check data.

Scan Plan — Instructions that drive an inspection; these include inspection area geometry, ordered inspection path points, inspection probe selection, inspection path for each probe, mechanical commands that allow mechanical manipulator positioning, instrument setting, and all the variables needed for signal processing and flaw data acquisition during inspection.

Scan Plan Specifications — Standards and procedures used in creating Eddy current and ultrasonic scan plans for the RFC system.

Schema — Formal definition of information structure. See Conceptual Schema, Physical Schema, Run-time Schema.

Shape — The physical geometry of a mechanical part, as distinguished from a computer description of that geometry. Where the difference is significant, the attitude is taken that shape is nominal or basic, with shape variations of tolerances grafted thereon

Shape Data — Include the geometric, topological description of a product along with the associated dimensional tolerances and feature descriptions.

---

[3] Ibid., p. 214.

Shape Representation — A computer description of shape. The description may be partial in the sense that not all aspects of part shape are indicated. For example, a body of revolution representative of a turned part may not describe the nonaxisymmetric aspects of part geometry. A solid model must be complete and unambiguous in the sense that it describes a single volume in 3-D space.

Single Spatial Probe/Transducer Path — The starting and ending location of a single probe movement.

Size Tolerance — Size tolerances specify the allowable variation in size-of-model features, independent of location. Size tolerances include circumferential, rectilinear size, and angular size.

Solid Geometric Model (Shape Representation) — A computer description of shape. The description may be partial in the sense that not all aspects of part shape are indicated. For example, a body of revolution representation of a turned part may not describe the nonaxisymmetric[4] aspects of part geometry. A solid model must be complete and unambiguous in the sense that it describes a single volume in 3-D space.

Solid Modeling — The creation of an unambiguous and complete representation of the size and shape of an object.

Source Code — A computer program written in some language which is processed to produce machine code.

Spline — A piecewise polynomial of order K, having continuity up to order K-1 at the segment joints.

Squirter Blueprint — Blueprint of the squirter head that houses the ultrasonic transducer.

Subface — A subface is a bounded portion of a face. It is defined by an underlying face, exactly one periphery closed curve and zero, one, or more internal closed curves that represent cutouts or holes in the region. The internal closed curve must not touch or intersect each other or the periphery closed curve and must be entirely contained within the periphery closed curve.

Surface Flaw — A surface anomaly.

Surface Flaw Major Characteristic — A vector determined by an agreed upon method.

> Example: A vector representing the major axis of the minimum elliptical envelope encompassing the anomaly in the plane of the surface.

Surface Flaw Tolerance — A unique combination of:

(a) Surface flaw orientation range.
(b) Serviceable surface flaw tolerance limits.

---

[4] Ibid., p. 211.

(c) Repairable surface flaw tolerance limits.

Surface Flaw Tolerance Limit — A unique combination of:

(a) Maximum length.
(b) Maximum width.
(c) Maximum depth.

Sweep Surface — Surfaces formed by extruding or revolving a planar profile in space.

Syntax — Grammar: A set of rules for forming meaningful phrases and sentences from words in a vocabulary.

System Computer — VAX 11/780 and supporting peripheral hardware.

System Constraints — Those hardware and software environmental constraints which will be imposed upon the PDDI Access Software that will limit its implementaiton and application. An example of such constraints might be the particular compiler used to compile the PDDI Access Software package.

To-Be — The future condition possible, given a proposed capability.

Tolerance (Generic) — The total amount by which something may vary. For mechanical product definition, tolerances can be shape tolerances, weight tolerances, finish tolerances and so on. In the context of GMAP, the term "tolerance" used alone implies shape tolerance. Other forms of tolerance (nonshape) are explicitly stated, for example, "finish tolerance." In a GMAP product model, tolerances occur without dimensions. As in the Product Definition Data Interface Program, model dimensions are implicit in the model geometry. Therefore, application of a tolerance implies a specific underlying dimension or geometric condition.

Topology — A data structure that assembles geometric entities (points, curves, surfaces) into a solid geometric model.

Transducer Blueprint — Blueprint of ultrasonic transducer supplied by the transducer manufacturer.

Transfer Data — The data required to make an exchange of data between systems (i.e., delimiters, record counts, record length, entity counts, numeric percision).

Translator — A software MECHANISM that is used for passing data between the Exchange Format and Working Form of the PDD.

Ultrasonic Cell — Hardware used to perform ultrasonic inspection operation (internal flaws).

Ultrasonic Inspection — An inspection method used to detect surface flaws on a disk. It uses ultrasonic waves through a stream of water to send and collect signals concerning an area targeted for inspection.

Ultrasonic Scan Plan — Interpreter code program controlling the ultrasonic inspection of a particular geometry.

Unbounded Geometry — Geometry represented parametrically, without limits, usually by coefficients to a defining equation.

Unigraphics (UG) — A computer graphics system.

User Function (UFUNC) — An interface to the UG database.

Working Form — Product definition data information in machine-dependent data formats; a memory resident network model.

Zone — A physical area of the disk composed of zone components.

Zone Component — A subface, face, or feature that constitutes a zone or element of a zone.

2.2.2 Acronyms Used In GMAP

ADB	— Application Data Block (also referred to as Attribute Data Block).
AIMS	— Automated IDEF Methodology System.
ANSI	— American National Standards Institute.
ANT	— Abstract of New Technology.
APT	— Automatically Programmed Tools.
ATP	— Automation Technology Products
BOM	— Bill of Materials.
BOR	— Body of Revolution.
BPI	— Bits per Inch.
BREP	— Boundary Representation.
CAD	— Computer Aided Design.
CAE	— Computer Aided Engineering.
CAEDS	— Computer Aided Engineering Design System.
CALS	— Computer-aided Acquisition and Logistics.
CAM	— Computer Aided Manufacturing.
CAM-I	— Computer Aided Manufacturing--International.
CAPP	— Computer Aided Process Planning.
CAS	— Cooled Airfoil System.
CDM	— Common Data Model.
CDR	— Critical Design Review.
CDT	— Component Design Technology.
CFSR	— Contract Fund Status Report.
CI	— Configuration Item.
CIM	— Computer Integrated Manufacturing.
CLIST	— IBM Command List.
CM	— Configuration Management.
CMM	— Coordinate Measuring Machine.
C/SSR	— Cost/Schedule Status Report.
CWBS	— Contract Work Breakdown Structure.
DBMS	— Data Base Management System.
DCL	— DEC Command Language.
DDL	— Data Definition Language.
DEA	— Digital Equipment Automation.

DEC	— Digital Equipment Corporation.
DES0	— (ICAM) Architecture of Design.
DJR	— Design Job Request; Drafting Job Request.
DoD	— Department of Defense.
DS	— Design Specification.
DSM	— Design Substantiation Memo.
EBCDIC	— Extended Binary Coded Decimal Interchange Code (IBM Character Set).
EC	— Eddy Current.
ECO	— Engineering Change Order.
EDM	— Electrical Discharge Machining.
EF	— Exchange Format.
EII	— Engineering Information Index.
EMD	— Engineering Master Drawing.
EPCS	— Engine Product Configuration Support.
ESA	— Engineering Source Approval.
ESP	— Experimental Solids Proposal.
FEDD	— For Early Domestic Dissemination.
FEM	— Finite-Element Modeling.
FOF	— Factory of the Future.
FOS	— Feature of Size.
FPIM	— Fluorescent Penetrant Inspection Module.
FSCM	— Federal Supply Code for Manufacturers.
GE	— General Electric.
GMAP	— Geometric Modeling Applications Interface Program.
GSE	— Ground Support Equipment.
HCF	— High-Cycle Fatigue.
IBIS	— Integrated Blade Inspection System.
IBM	— International Business Machines.
ICAM	— Integrated Computer Aided Manufacturing.
ICOM	— Input/Control/Output/Mechanism.
ICS	— Information Computer System.
IDEF	— ICAM Definition.
IDEF0	— IDEF Function Modeling.
IDEF1	— IDEF Information Modeling.
IDEF1X	— IDEF Extended Information Modeling.
IDEF2	— IDEF Dynamics Modeling.
IDSS	— Integrated Decision Support System.
IEEE	— Institute of Electrical and Electronics Engineers.
IEN	— Internal Engineering Notice.
IFS	— Interface Specification.
IGES	— Initial Graphics Exchange Specification.
IISS	— Integrated Information Support System.
ILC	— Improved Life Core.
IMS	— Information Management System.
IPGS	— (IBIS) Inspection Plan Generation Subsystem.
IRB	— Industry Review Board.
IRIM	— Infrared Inspection Module.
ISO	— International Standards Organization.

R3244/3

ITA	— Intelligent Task Automation.
ITI	— International TechneGroup Incorporated.
ITR	— Interim Technical Report.
LCF	— Low-Cycle Fatigue.
MAS	— Model Access Software.
MCAIR	— McDonnell Douglas Corporation/McDonnell Aircraft Company.
MFG0	— (ICAM) Architecture of Manufacturing.
MRP	— Materials Requirements Planning.
NAD	— Needs Analysis Document.
NBS	— National Bureau of Standards.
N/C	— Numerical Control.
NDE	— Nondestructive Evaluation.
NDML	— Neutral Data Manipulation Language.
NDT	— Nondestructive Test.
NTSB	— National Transportation Safety Board.
NVI	— Name/Value Interface.
OGP	— Optical Gaging Products, Inc.
PA/QA	— Product Assurance/Quality Assurance.
PD	— Product Data.
PDD	— Product Definition Data.
PDDI	— Product Definition Data Interface Program.
PDES	— Product Data Exchange Specification.
PDL	— Program Design Language.
PED	— Preliminary Engine Design.
PI	— Principal Investigator.
PID	— PDDI Interim Database.
PIES	— Product Information Exchange System.
PMP/PMS	— Program Management Plan/Project Master Schedule.
PROCAP	— Process Capability.
PS	— Product Specification.
RFC	— Retirement for Cause Disk Inspection System.
RPM	— Revolutions per Minute.
SA-ALC	— San Antonio-Air Logistics Center.
SAD	— State-of-the-Art Document.
SD	— Scoping Document.
SDL	— Source Data List.
SDS	— System Design Specification.
SL	— Salvage Layout.
SML	— Source Material Log.
SOA	— State-of-the-Art (Survey).
SOR	— Surface of Revolution.
SPC	— Statistical Process Control.
SPF	— System Panel Facility.
SQA	— Software Quality Assurance.
SQAP	— Software Quality Assurance Plan.
SRD	— System Requirements Document.
SRL	— Systems Research Laboratories.
SS	— System Specification.
STEP	— Standard for the Exchange of Product Model Data.

STP	—System Test Plan.
TCTO	—Time Compliance Technical Order.
TD	—Technical Data.
TDCR	—Turbine Design Cost Reduction.
TDR	—Tool Design Request.
TechMod	—Technology Modernization.
TO	—Technical Order.
TOP	—Technical and Office Protocol.
TSO	—Time-Sharing Option (IBM term).
UFUNC	—User Function.
UG	—Unigraphics.
UGFM	—Unigraphics File Manager.
USA	—Unified System for Airfoils.
USAF	—United States Air Force.
UTC	—United Technologies Corporation.
UTP	—Unit Test Plan.
UTR	—Unit Test Report.
UTRC	—United Technologies Research Center.
VAX	—Virtual Architecture Extended.
VMS	—Virtual Memory System.
WBS	—Work Breakdown Structure.
WF	—Working Form.
WPAFB	—Wright-Patterson Air Force Base.
XIM	—X-Ray Inspection Module.

SECTION 3

SYSTEM OVERVIEW

The three major functions of the Schema Manager software include the ability to:

- Model a concrete conceptual schema

- Transform a concrete conceptual schema into a physical schema suitable for the Working Form (WF) of the Model Access Software (MAS)

- Generate subschema forms of the physical schema for use by application programs at the time of compiling and/or at the time of running.

3.1 PURPOSE OF THE SCHEMA MANAGER SOFTWARE

The concrete conceptual schema is the logical view of data called the "Conceptual Model" in the ANSI/SPARC three-schema approach and the "Schema" in the CODASYL 78 approach. It contains an unambiguous set of requirements for the data objects (as opposed to an abstract conceptual schema which may be incomplete or contain ambiguity).

The physical schema is the implementation view of data called the "Internal Model" in the ANSI/SPARC three-schema approach and the "Storage Schema" in the CODASYL 78 approach. It contains the resolution of the requirements for the data objects (such as size, boundary alignment, and location).

The subschema forms of the physical schema are the application views of data called the "External Model" in the ANSI/SPARC three-schema approach and the "Sub-Schema" in the CODASYL 78 approach. The forms of the physical schema generated by the schema manager include:

- A PASCAL language "Include" file of types and constants
- A report of the physical definitions for general use
- A GMAP/PDDI Data Dictionary
- A Run-Time Subschema.

These forms of the physical schema are explained in detail in Section 4.5.

The content of the conceptual schema is based on the EXPRESS information modeling language, and is an extension of the GMAP/PDDI metamodel. This is further explained in Section 3.2.

3.2 THE METAMODEL

There are three distinct phases, or aspects of data modeling:

- Definition of the metamodel, or model of the data
- Definition of entities within the context of the metamodel

- Creation of instances of entities using those definitions.

A metamodel is simply data about data. It is the framework within which specific entity definitions can be made. The second and third phases of data modeling can take place only when a metamodel framework has been defined.

The metamodel is used to establish the content and structure of the schema. It is also used in mapping the conceptual schema into implementation. This task is the primary function of the Schema Manager Software.

Figure 3-1 presents the metamodel used by the Schema Manager Software. It is an extension of the metamodel developed for the GMAP/PDDI Project sponsored by the Computer Integrated Manufacturing Branch of the Air Force Systems Command. The metamodel is based on commonly used models, specifically, the relational and network models. The elements used in the metamodel are described in Table 3-1.

3.3 PHYSICAL REPRESENTATION

The physical representation for an entity in the MAS WF depends upon the attribute data types: pointers are placed in a special data structure called a constituent list; the other primitive data types and the one complex primitive data type are grouped into a data structure called an application data block (ADB).

The aggregation data types are located according to the primitive data types of which they are composed. The order of attributes within each of the data structures is called the physical schema order.
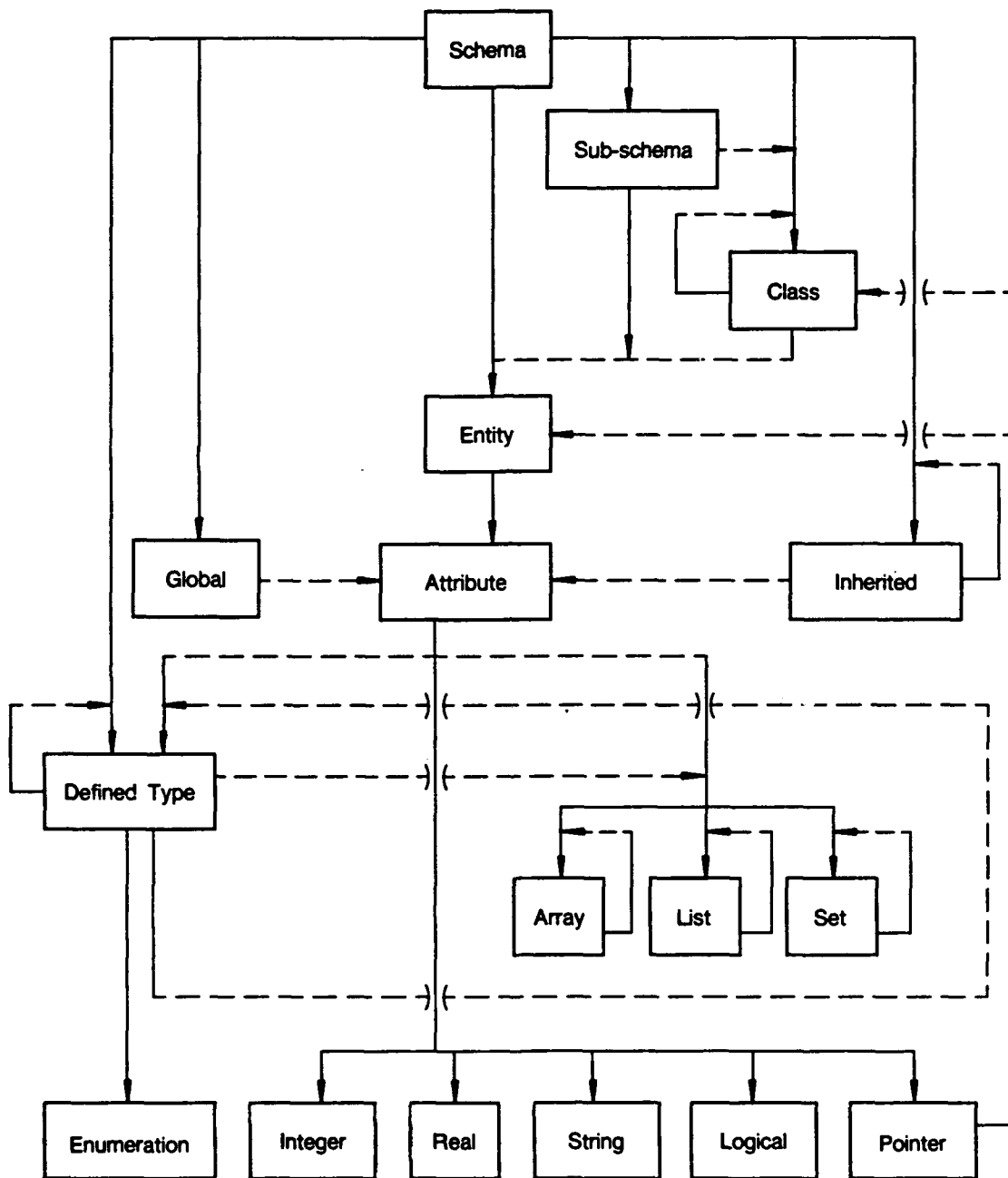
FDA 357846

Figure 3-1. Metamodel for the Schema Manager

TABLE 3-1

ELEMENTS OF THE METAMODEL USED

BY THE SCHEMA MANAGER SOFTWARE

ENTITY	A collection of facts having meaning or purpose. An entity is defined by one or more attributes: global attributes apply to all entities in the schema; inherited attributes apply to the entities that are a SUBTYPE OF that inheritance group; local attributes are unique to an entity.
ATTRIBUTE	A defining fact of an entity. A required attribute must have a valid value in every instance of the entity. An optional attribute may not have a valid value in a specific instance of the entity, which in itself is meaningful. Each attribute is defined in terms of a specific data type. The data type may be a primitive (integer, real, string, logical, pointer), complex primitive (enumeration), or aggregation (array, list, set). The data types are defined below:
PRIMITIVE	
INTEGER	A whole number where the natural domain of values is implementation dependent. The precision is the number of decimal digits.
REAL	A floating point number defined by an exponent and a mantissa. The precision is the number of decimal digits in the mantissa.
STRING	A fixed size array of characters. The character set is implementation dependent (i.e., EBCDIC for IBM), as is the collating sequence used in comparisons. Each string is defined by the number of characters it can hold.
LOGICAL	A binary set of TRUE or FALSE.
POINTER	A reference to another entity. The entity pointed to is called a constituent. The entity containing the attribute of pointer data type is called a user. A pointer must specify the entities, or classes of entities, for which it is valid. Any instance of these entities can be pointed to.
COMPLEX PRIMITIVE	
ENUMERATION	An ordered list of possible values. The number of values is its cardinality. The position of a specific value is its ordinality.
AGGREGATION	
ARRAY	An ordered collection of elements of a single primitive data type, which is fixed in size. Each dimension of an array is defined by a lower bound and an upper bound. These specify the number of elements in that dimension, and map to the first and last positions.
LIST	An ordered collection of elements of a single primitive data type, which is variable in size. The apparent size of the list is bounded by minimum and maximum occurrences.
SET	An unordered collection of elements of a single primitive data type, which is variable in size. The apparent size of the list is bounded by minimum and maximum occurrences. (This data type should not be confused with the keyword "set" in the PASCAL language which defines a construct that can be used only to inquire about the presence or absence of a member.)
DEFINED TYPE	An alias for a primitive, complex primitive or aggregation data type.
CLASS	A collection of entities, or classes of entities.
SUB-SCHEMA	A collection of entities, or classes of entities, which defines the scope of a physical schema. (Note: to ensure a complete definition, when a class is a member of a sub-schema, its members are implicitly included, and when an entity is a member of a sub-schema, its constituents are implicitly included.)
SCHEMA	All data in the schema model.

R20A6A/6

R20A6A/6

The location of an attribute within the appropriate data structure depends upon the order of attributes in the entity definition and, for those in the ADB, upon the data type requirements for size and boundary alignment. The order of attributes within the entity definition is called the conceptual schema order. It is determined by whether the attribute is global, inherited or local; and by its entry order within one of these groups.

Global attributes apply to all entities in the schema. They are defined in a single group which appears first for all entities. Global attributes are optional.

Inherited attributes are defined in one or more groups, and the groups may be organized in a hierarchy. They apply only to the entities that are a SUBTYPE OF that inheritance group (an entity can be a subtype of only one inheritance group, but the inheritance group can itself be a subtype of another inheritance group). Inherited attributes appear after the global attributes, in their top-down hierarchical order. Inherited attributes are optional.

Local attributes are unique to a single entity and appear as the last group. At least one local attribute is required.

Pointers represent constituent relationships between entities. Each entry in a constituent list will be automatically matched, by the MAS, with an entry in a user list to complete the binary relationship. The pointer data type has a single physical definition. Pointers are assigned to locations in the constituent list in the conceptual schema order.

The remaining data types, located in the ADB, have various size and boundary alignment requirements. They are assigned to locations within a group (i.e., global, inherited, local) as described below:

- Attributes that are aligned on double-word boundaries are assigned first (i.e., eight-byte REAL's); within the boundary alignment class, they are assigned to locations by the entry order within the group.

- Attributes that are aligned on full-word boundaries are assigned second (i.e., four-byte REAL's and four-byte INTEGER's); within the boundary alignment class, they are assigned to locations by the entry order within the group.

- Attributes that are aligned on half-word boundaries are assigned third (i.e., two-byte INTEGER's); within the boundary alignment class, they are assigned to locations by the entry order within the group

- Attributes that are aligned on bytes are assigned fourth (i.e., one-byte INTEGER's, LOGICAL's, ENUMERATION's, STRING's); within the boundary alignment class, they are assigned to locations by the entry order within the group.

The physical schema order of attributes in the ADB is summarized in Table 3-2.

3-5

TABLE 3-2

PHYSICAL SCHEMA ORDER OF ATTRIBUTES IN THE ADB

MAS required	:	KIND	
		LENGTH	
		SYSUSE	
global	:	double words	: entry order
		full words	: entry order
		half words	: entry order
		bytes	: entry order
top level inherited	:	double words	: entry order
		full words	: entry order
		half words	: entry order
		bytes	: entry order
bottom level inherited	:	double words	: entry order
		full words	: entry order
		half words	: entry order
		bytes	: entry order
local	:	double words	: entry order
		full words	: entry order
		half words	: entry order
		bytes	: entry order

R20864/8

The physical schema order of attributes is determined as described above unless the
position override is used when defining one or more of the attributes in a group (i.e., global,
inherited, local). In this case, the attributes with a position override are assigned to locations
within the group in position number order, before considering the attributes without a position
number (or with a position number of zero). The attributes without a position override are
assigned locations according to their boundary alignment requirements, and their entry order
within the group. Since the groups of attributes are shared in a hierarchical manner, the position
override affects all entities sharing the groups, and therefore, physical compatibility is assured.
The use of the position override can result in an inefficient allocation of storage for the ADB, and
is intended only as a last resort for mapping onto data that was created prior to the entity
definition.

The physical representation of the primitive data types is summarized in Table 3-3.

TABLE 3-3

PHYSICAL REPRESENTATION OF THE PRIMITIVE DATA TYPES

Data Type	Digits	Bytes	Implemented as
INTEGER	1-2	1	IBM Packed 0..255, VAX [BYTE] 0..255
	3-4	2	IBM Packed 0..65535, VAX [WORD] 0..65535
	5-9	4	Integer
REAL	1-7	4	IBM shortreal, VAX real
	8-16	8	IBM real, VAX double
STRING(n)		n	packed array (1..n) of char
LOGICAL		1	Boolean
POINTER		4	MAS ENTKEY
ENUMERATION		1	for a cardinality of m, packed 0..(m-1)

The physical representation of an array of pointers makes use of a pseudo-entity called the ARRAY ENTITY. Whereas an attribute with a data type of pointer is represented by an entry in the constituent list referring directly to the constituent, for an array of pointers the entry in the constituent list refers to an ARRAY ENTITY whose "constituent list" is the array of direct references to the constituents. For a two-dimensional array, a first level ARRAY ENTITY refers to a second level group of ARRAY ENTITIES, one for each "row" of the array. For an n-dimensional array there are n-levels of ARRAY ENTITIES, with the n-th level containing the direct references to the constituents. This approach allows applications programs to navigate multi-dimensional arrays of constituents with the same MAS functions used for navigating constituents not in an array.

Please note that the pseudo-entity ARRAY ENTITY (KIND = 1100) should be used only for the purpose of representing arrays of pointers. It is usually handled differently from true entity definitions, and should, therefore, not be used for any other purpose. The physical representation of the ARRAY ENTITY is automatically generated by the Schema Manager software. The list of eligible KINDS defined for the ARRAY ENTITY in the GMAP/PDDI Data Dictionary includes all of the entity KINDS and class KINDS referred to in any definition of an attribute that is an "array of pointers."

List and set have been temporarily implemented with the same physical representation as the array. Currently, the only effect of using these data types is that the "minimum occurrences" will appear in the GMAP/PDDI Data Dictionary entry for the attribute.

3.4 INVOKING THE SCHEMA MANAGER SOFTWARE

After the Schema Manager software has been properly installed as specified in the Operator's Manual (OM560240001U), the INTERACTIVE and/or BATCH INTERFACE of the software can be invoked by executing the following clists:

EXEC /CAD5.DBMA.SCE.CLIST(SCHEMA)'

1. The user is prompted for a qualifier name for the schema model files. If /HELP' is entered, a brief explanation is provided.

    If there is no schema model file for the specified name, a message specifying that fact is displayed and confirmation is requested that a new schema model is intended.

2. For existing schema model files, last access information is displayed.

3. The master file and output files are allocated.

4. A prompt is given for a source file for batch input. To input a batch file, enter the file name in quotes. The file, if complete and correct, is modelled and filed to disk. The batch input report can be printed and the interactive mode can be entered.

5. The dialog manager files are allocated and the Schema Manager Main Menu is displayed.

The MODEL QUERY UTILITY can be invoked by executing:

EXEC /CAD5.DBMA.MQU.CLIST(MQUERY)'

3.4.1 Interactive Interface Invocation Example

The interactive interface invocation example is listed below.

Input: EXEC /CAD5.DBMA.SCE.CLIST(SCHEMA)'

Result: ENTER THE QUALIFIER NAME FOR THE SCHEMA MODEL
          FILES (BLANK TO QUIT, /HELP' FOR AN EXPLANATION):

Input: SCETEST2

Result: CAD5.DBMA.SCETEST2.SCHEMA DOES NOT EXIST.
          DID YOU WANT TO CREATE A NEW FILE (/Y',/N'):

3-8

Input: Y

Result: ALLOCATING MASTER FILE FOR THE SCETEST SCHEMA
          MODEL...
          ALLOCATING OUTPUT FILES FOR THE SCETEST SCHEMA
          MODEL...
          ENTER THE SOURCE FILE FOR BATCH INPUT (BLANK IF
          NONE):

Input: <enter>

Result: ALLOCATING DIALOG MANAGER FILES...
          ***

Input: <enter>

Result: menu: MAIN MENU

          .
          .
          .
          .

3.4.2 Batch Interface Invocation Example

The Batch Interface Invocation example is listed below.

Input: EXEC /CAD5.DBMA.SCE.CLIST(SCHEMA)'

Result: ENTER THE QUALIFIER NAME FOR THE SCHEMA MODEL
          FILES (BLANK TO QUIT, /HELP' FOR AN EXPLANATION):

Input: SCETEST2

Result: ALLOCATING MASTER FILE FOR THE SCETEST SCHEMA
          MODEL...

R20356/6

LAST ACCESS: <userid> <date> <time>
ALLOCATING OUTPUT FILES FOR THE SCETEST SCHEMA
MODEL...
ENTER THE SOURCE FILE FOR BATCH INPUT (BLANK IF
NONE):

Input: /CAD5.DBMA.SCETEST2.CSR'

Result: PROCESSING BATCH INPUT...
          THE MODEL HAS BEEN FILED.
          PRINT THE BATCH INPUT REPORT (/Y',/N'):

Input: Y

Result: ENTER THE INTERACTIVE MODE (/Y',/N'):

Input: N

Result: PRINT OUTPUT FILES (/Y',/N'):

Input: N

Result: READY

3.4.3 Model Query Utility Invocation Example

The Model Query Utility Invocation example is shown below.

Input: EXEC /CAD5.DBMA.MQU.CLIST(MQUERY)'

Result: menu: MODEL QUERY MAIN MENU

SECTION 4

INTERACTIVE INTERFACE

This section identifies and explains the menus available within the interactive interface of the Schema Manager software.

The commands available from the main menu allow a user to retrieve a model from disk, create, update, review, generate reports, file a model to disk, and exit the Schema Manager.

```
-------------------------------MAIN MENU----------------------------

COMMAND ===> (a)



            COMMANDS:
                    1    : EXIT SCHEMA MANAGER
                    2    : RETRIEVE MODEL FROM DISK
                    3    : CREATE
                    4    : UPDATE
                    5    : REVIEW
                    6    : GENERATE REPORTS
                    7    : FILE MODEL TO DISK

            ENTER A COMMAND (ABOVE).
```

Figure 4-1. Schema Manager Main Menu

Menu Description:

(a) Select command 1 to exit from the Schema Manager. The REPLACE OPTION MENU is displayed.
Select command 2 to retrieve a model from a disk file into the working form.
Select command 3 to display the CREATE MENU.
Select command 4 to display the UPDATE MENU.
Select command 5 to display the REVIEW MENU.
Select command 6 to display the REPORT MENU.
Select command 7 to file a model from the working form into a disk file.

4.1 RETRIEVE MODEL FROM DISK

If command 2, **Retrieve Model From Disk,** is selected from the MAIN MENU, the Schema Manager displays one of the following messages in the top right corner of the display screen:

- Model retrieved
- No Model retrieved.

If command 3, **Create,** is selected from the MAIN MENU without having previously retrieved a model, the user is asked, "SHOULD A NEW MODEL BE CREATED?"

```
---------------------------CREATING A NEW MODEL-----------------------



    WARNING :  A MODEL HAS NOT BEEN RETRIEVED FROM
               DISK INTO WORKING FORM.

    SHOULD A NEW MODEL BE CREATED? (a)
          (ENTER 'Y' OR 'N')
```
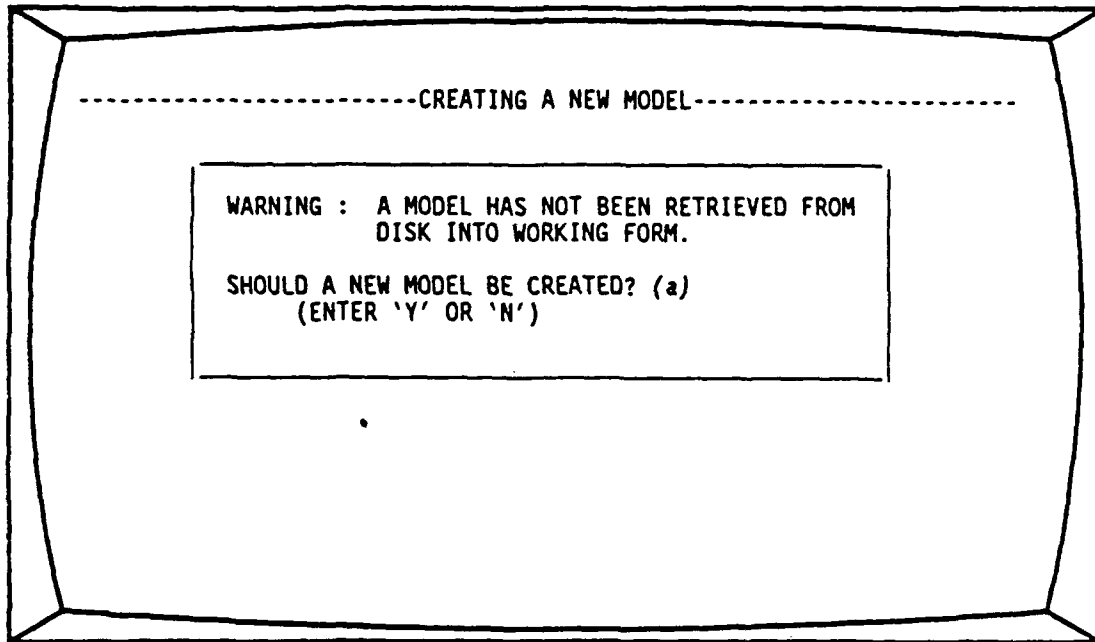
Figure 4-2. Creating a New Model

Menu Description:

(a) Enter /Y' to display the CREATE MENU.
Enter /N' to redisplay the MAIN MENU.

4.2 CREATE OPERATIONS

The CREATE functions within the Schema Manager software can be accessed from interactive menus. There are six functions available directly from the main CREATE menu. Other supporting menus provide the remaining capabilities.

The create menus are listed in Table 4-1 along with a section reference to the detailed definitions.

TABLE 4-1

INTERACTIVE INTERFACE

CREATE Menus	Section
CREATE ENTITY	4.2.1
CREATE DEFINED TYPE	4.2.2
CREATE SUPERTYPE	4.2.3
CREATE GLOBAL ATTRIBUTE	4.2.4
CREATE SUBSCHEMA	4.2.5
CREATE CLASS	4.2.6
OTHER SUPPORTING MENUS	4.2.7
CREATE ARRAY	4.2.7.1
CREATE ATTRIBUTE	4.2.7.2
CREATE CLASS MEMBERS	4.2.7.3
CREATE ENUMERATION	4.2.7.4
CREATE INTEGER	4.2.7.5
CREATE LIST	4.2.7.6
CREATE POINTER	4.2.7.7
CREATE REAL	4.2.7.8
CREATE SET	4.2.7.9
CREATE STRING	4.2.7.10
CREATE/REFERENCE SUPERTYPE	4.2.7.11
DISPLAY LIST	4.2.7.12
LIST MEMBERS	4.2.7.13

R20358/7

Create Menu

The commands available from the CREATE MENU allow a user to create an entity, a defined type, a supertype, a global attribute, a subschema, or a class.

```
-------------------------------CREATE MENU-------------------------------

COMMAND ===> (a)



                   COMMANDS:
                      1    : RETURN TO MAIN MENU
                      2    : CREATE ENTITY
                      3    : CREATE DEFINED TYPE
                      4    : CREATE SUPERTYPE
                      5    : CREATE GLOBAL ATTRIBUTE
                      6    : CREATE SUBSCHEMA
                      7    : CREATE CLASS

                   ENTER A COMMAND (ABOVE).
```

Figure 4-3. Create Menu

Menu Description:

(a) Select command 1 to redisplay the MAIN MENU.
Select command 2 to display the CREATE ENTITY MENU.
Select command 3 to display the CREATE DEFINED TYPE MENU.
Select command 4 to display the CREATE SUPERTYPE MENU.
Select command 5 to display the CREATE GLOBAL ATTRIBUTE MENU.
Select command 6 to display the CREATE SUBSCHEMA MENU.
Select command 7 to display the CREATE CLASS NAME AND KIND NUMBER MENU.

4.2.1 Create Entity

The CREATE ENTITY menu allows a user to define the characteristics of an entity.


```
------------------------------CREATE ENTITY------------------------------

COMMAND ===> (a)


                    COMMANDS:
                         1     : RETURN TO PREVIOUS MENU
                         2     : EXIT TO CREATE MENU

                    ENTER EITHER A COMMAND (ABOVE) OR
                    ENTITY DATA (BELOW).


                    NAME                          ===> (b)
                    KIND NUMBER                   ===> (c)
                    REFERENCE SUPERTYPE (Y/N)? ===> (d)

                    COMMENTS:
                    >>   (e)
                    >>
                    >>
```

Figure 4-4. Create Entity

Menu Description:

(a) Select command 1 to redisplay the previous menu.
     Select command 2 to redisplay the CREATE MENU.

(b) Enter the name of the entity to be added to the model.

(c) Enter the entity kind number.

(d) Enter /Y' to display the CREATE/REFERENCE SUPERTYPE MENU.
     Enter /N' if this entity is not a SUBTYPE.
     After the entity name, kind number, and supertype reference have been
     entered, the CREATE ATTRIBUTE MENU will be displayed.

(e) Enter remarks as appropriate (maximum 150 characters). These will be
     saved and will appear on the Conceptual Schema Report and the PASCAL
     Include File.

4-5

4.2.2 Create Defined Type

The CREATE DEFINED TYPE menu allows a user to define the characteristics of a defined type.

```
 ------------------------------CREATE DEFINED TYPE------------------------------
COMMAND ===> (a)

         +--------------------------------------------------------+
         | COMMANDS:                                              |
         |         1      : RETURN TO PREVIOUS MENU               |
         |         2      : EXIT TO CREATE MENU                   |
         |         3      : SELECT FROM CURRENT DEFINED           |
         |                    TYPES                               |
         |                                                        |
         | ENTER EITHER A COMMAND (ABOVE) OR A DEFINED            |
         | TYPE NAME AND TYPE NUMBER (BELOW).                     |
         +--------------------------------------------------------+

           TYPE NAME    ===> (b)
           TYPE NUMBER  ===> (c)
     +-------------------------------------------------------------+
     |                    TYPE NUMBERS                             |
     | 1 : INTEGER   4 : LOGICAL   7 : SET        10 : ENUMERATION |
     | 2 : REAL      5 : ARRAY     8 : POINTER                     |
     | 3 : STRING    6 : LIST      9 : DEFINED TYPE                |
     +-------------------------------------------------------------+
```

Figure 4-5. Create Defined Type

Menu Description:

(a) Select command 1 to redisplay the previous menu.
Select command 2 to redisplay the CREATE MENU.
Select command 3 to display the modeled defined types, one of which may be selected. In choosing this command a new defined type is not created. The DISPLAY LIST MENU is displayed.

(b) Enter the defined type name.

(c) Enter the defined type number (1-10).
After a defined type name and number are accepted, the appropriate create type menu is displayed. Possible create menus to be displayed include:

CREATE INTEGER	CREATE ARRAY	CREATE POINTER
CREATE REAL	CREATE LIST	CREATE DEFINED TYPE
CREATE STRING	CREATE SET	CREATE ENUMERATION

4.2.3 Create Supertype

The CREATE SUPERTYPE menu allows a user to define the characteristics of a supertype.

```
 -------------------------------CREATE SUPERTYPE-------------------------------

  COMMAND ===> (a)


                 ┌──────────────────────────────────────────────┐
                 │  COMMANDS:                                     │
                 │          1    : RETURN TO PREVIOUS MENU        │
                 │          2    : EXIT TO CREATE MENU            │
                 │                                                │
                 │  ENTER EITHER A COMMAND (ABOVE) OR SUPERTYPE   │
                 │  DATA (BELOW).                                 │
                 └──────────────────────────────────────────────┘

                 NAME                         ===> (b)
                 REFERENCE SUPERTYPE (Y/N)? ===> (c)
```

Figure 4-6. Create Supertype

Menu Description:

(a) Select command 1 to redisplay the previous menu.
Select command 2 to redisplay the CREATE MENU.

(b) Enter the name of the supertype to be added to the model.

(c) Enter /Y' to display the CREATE/REFERENCE SUPERTYPE MENU.
Enter /N' if this supertype does not refer to a parent supertype.
After the supertype name and supertype reference have been entered, the
CREATE ATTRIBUTE MENU will be displayed.

4-7

R20464/7

4.2.4 Create Global Attribute

The CREATE GLOBAL ATTRIBUTE menu allows a user to define the characteristics of a global attribute.

```
------------------------------CREATE GLOBAL ATTRIBUTE------------------------
COMMAND ===> (a)

   COMMANDS:
      1  :  EXIT TO CREATE MENU
   ENTER EITHER A COMMAND (ABOVE) OR ATTRIBUTE DATA (BELOW).


   GLOBAL ATTRIBUTE
      NAME                        ===> (b)
      REQUIRED/OPTIONAL (R/O)     ===> (c)
      POSITION (OPTIONAL INTEGER) ===> (d)
      TYPE NUMBER                 ===> (e)
      COMMENT>> (f)

               TYPE NUMBERS
         1 : INTEGER   4 : LOGICAL    7 : SET
         2 : REAL      5 : ARRAY      8 : DEFINED TYPE
         3 : STRING    6 : LIST
```

Figure 4-7. Create Global Attribute

Menu Description:

(a) Select command 1 to redisplay the CREATE MENU.

(b) Enter the name of the global attribute to be included in the model.

(c) Enter /R' to select "required" or /O' to select "optional". The software will default to /R'.

(d) Enter the user specified physical position number of the global attribute in the ADB. (This entry is optional.)

(e) Enter the attribute type number (1-8).
     After the global attribute data is accepted, the appropriate create type menu is displayed. Possible create menus to be displayed include:

     CREATE INTEGER   CREATE ARRAY   CREATE DEFINED TYPE
     CREATE REAL   CREATE LIST
     CREATE STRING   CREATE SET

4-8

(f)   Enter remarks as appropriate (maximum 50 characters). The comments will
be saved and will appear on the Conceptual Schema Report and the
PASCAL Include File.

4.2.5 Create Subschema

The CREATE SUBSCHEMA menu allows a user to define the characteristics of a
subschema.

```
-------------------------CREATE SUBSCHEMA------------------------
COMMAND ===> (a)

        COMMANDS:
             1   : EXIT TO CREATE MENU
             2   : SELECT A MEMBER KIND NUMBER FROM THE
                     EXISTING CLASSES AND ENTITIES
             3   : LIST CURRENT MEMBERS IN THE SUBSCHEMA
             4   : SAVE SUBSCHEMA CREATED

        ENTER EITHER A COMMAND (ABOVE) OR A SUBSCHEMA NAME
        AND MEMBER KIND NUMBERS TO BE INCLUDED IN THE SUB-
        SCHEMA (BELOW).

        SUBSCHEMA NAME      ===> (b)
        COMMENTS:
        >>
        >>    (c)
        >>
        MEMBER KIND NUMBER ===> (d)
```

Figure 4-8.   Create Subschema

Menu Description:

(a)   Select command 1 to redisplay the CREATE MENU.
Select command 2 to display the eligible entities and classes which may be
selected to be members of the subschema. The DISPLAY LIST MENU is
displayed.
Select command 3 to display the current entity and class members of the
subschema. The LIST MEMBERS MENU is displayed.
Select command 4 to model the created subschema. The CREATE MENU
is redisplayed.

(b)   Enter the name of the subschema to be added to the model.

(c)   Enter remarks as appropriate (maximum 150 characters). The comments
will be saved and will appear on the Conceptual Schema Report and the
PASCAL Include File.

4-9

(d) Enter the kind number of a modeled entity or class that is to be made a member of the subschema. As each kind number is accepted, the CREATE SUBSCHEMA MENU is redisplayed.

NOTE: Once a subschema name is accepted, it cannot be altered in this menu.

R2044/7

4.2.6 Create Class Name and Kind Number

The CREATE CLASS NAME AND KIND NUMBER Menu allows a user to define the characteristics of a class.

```
------------------CREATE CLASS NAME AND KIND NUMBER------------------

COMMAND ---> (a)


                    ┌──────────────────────────────────────┐
                    │ COMMANDS:                            │
                    │     1     : RETURN TO PREVIOUS MENU  │
                    │     2     : EXIT TO CREATE MENU      │
                    │                                      │
                    │ ENTER EITHER A COMMAND (ABOVE) OR    │
                    │ A CLASS NAME AND KIND NUMBER (BELOW) │
                    └──────────────────────────────────────┘

                    NAME        ---> (b)
                    KIND NUMBER ---> (c)
                    COMMENTS:
                    >>   (d)
                    >>
```

Figure 4-9. Create Class Name and Kind Number

Menu Description:

(a) Select command 1 to redisplay the previous menu.
     Select command 2 to redisplay the CREATE MENU.

(b) Enter the class name.

(c) Enter the class kind number.

(d) Enter remarks as appropriate (maximum 150 characters). The comments will be saved and will appear on the Conceptual Schema Report and the PASCAL Include File.

After the class name, kind number, and comments have been accepted, the CREATE CLASS MEMBERS MENU is displayed.

4.2.7 Other Supporting Menus

The supporting menus available within the CREATE function include (alphabetically):

— Create Array
— Create Attribute
— Create Class Members
— Create Enumeration
— Create Integer
— Create List
— Create Pointer
— Create Real
— Create Set
— Create String
— Create/Reference Supertype
— Display List*
— List Members*.

Some of the menus listed above are also referred to from the REVIEW and UPDATE sections.

4.2.7.1 Create Array

The CREATE ARRAY menu allows a user to define the characteristics of an array.

```
 ----------------------------CREATE ARRAY----------------------------
 COMMAND ===> (a)

        ┌──────────────────────────────────────────────────┐
        │ COMMANDS:                                         │
        │       1    : RETURN TO PREVIOUS MENU              │
        │       2    : EXIT TO CREATE MENU                  │
        │                                                   │
        │  ENTER EITHER A COMMAND (ABOVE) OR ARRAY          │
        │  DATA (BELOW).                                    │
        └──────────────────────────────────────────────────┘

        LOW BOUND    ===> (b)
        HIGH BOUND   ===> (c)
        TYPE NUMBER  ===> (d)

        ┌──────────────────────────────────────────────────┐
        │                  TYPE NUMBERS                     │
        │  1 : INTEGER    4 : LOGICAL    7 : SET            │
        │  2 : REAL       5 : ARRAY      8 : POINTER        │
        │  3 : STRING     6 : LIST       9 : DEFINED TYPE   │
        └──────────────────────────────────────────────────┘
```

Figure 4-10. Create Array

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the CREATE MENU.

(b) Enter the array lower bound.

(c) Enter the array upper bound.

(d) Enter the array type number (1-9).
     After the array data is accepted, the appropriate create type menu is
     displayed. Possible create menus to be displayed include:

     CREATE INTEGER   CREATE ARRAY   CREATE POINTER
     CREATE REAL   CREATE LIST   CREATE DEFINED TYPE
     CREATE STRING   CREATE SET

4-13

4.2.7.2 Create Attribute

The CREATE ATTRIBUTE menu allows a user to define the characteristics of an attribute.

```
----------------------------CREATE ATTRIBUTE----------------------------
COMMAND ===> (a)

         COMMANDS:
             1  :  EXIT TO CREATE MENU
             2  :  DISPLAY ATTRIBUTES LOCAL TO (b)
             3  :  SAVE ATTRIBUTES CREATED
         ENTER EITHER A COMMAND (ABOVE) OR ATTRIBUTE
         DATA (BELOW).

         NAME                           ===> (c)
         REQUIRED/OPTIONAL (R/O)        ===> (d)
         POSITION (OPTIONAL INTEGER)    ===> (e)
         TYPE NUMBER                    ===> (f)
         COMMENT>>  (g)

                          TYPE NUMBERS
             1 : INTEGER    4 : LOGICAL     7 : SET
             2 : REAL       5 : ARRAY       8 : POINTER
             3 : STRING     6 : LIST        9 : DEFINED TYPE
```

Figure 4-11. Create Attribute

Menu Description:

(a) Select option 1 to redisplay the CREATE MENU.
Select option 2 to display the local attributes currently defined. The LIST MEMBERS MENU is displayed.
Select option 3 to model the entity created.

(b) Entity or Supertype Keyword displayed.

(c) Enter the name of the attribute to be included in the entity.

(d) Enter /R' to select "required" or /O' to select "optional." The software will default to /R'.

(e) Enter the user specified physical position number of the attribute in the ADB. (This entry is optional.)

R20458/7

(f) Enter the attribute type number (1-9).

After the attribute data is accepted, the appropriate create type menu is displayed. Possible create menus to be displayed include:

CREATE INTEGER CREATE ARRAY CREATE POINTER
CREATE REAL   CREATE LIST   CREATE DEFINED TYPE
CREATE STRING   CREATE SET

(g) Enter remarks as appropriate (maximum 50 characters). The comments will be saved and will appear on the Conceptual Schema Report and the PASCAL Include File.

R20654/7

4.2.7.3 Create Class Members

The CREATE CLASS MEMBERS menu allows a user to define the members of a class.

```
------------------------------CREATE CLASS MEMBERS----------------------

COMMAND ===> (a)


          COMMANDS:
              1  : EXIT TO CREATE MENU
              2  : DISPLAY CURRENT MEMBERS IN THE CLASS
              3  : SELECT A MEMBER KIND NUMBER FROM THE
                      EXISTING CLASSES AND ENTITIES
              4  : CREATE A CLASS WITHIN THE CLASS
              5  : CREATE AN ENTITY WITHIN THE CLASS
              6  : SAVE CLASS CREATED

          ENTER EITHER A COMMAND (ABOVE) OR THE KIND
          NUMBER OF AN EXISTING MEMBER THAT IS TO BE
          INCLUDED IN THE CLASS (BELOW).


          MEMBER KIND NUMBER ===> (b)
```

Figure 4-12.   Create Class Members

Menu Description:

(a)   Select option 1 to redisp¹   the CREATE MENU.
Select option 2 to display tue current entity and class members of the class.
The LIST MEMBERS MENU is displayed.
Select option 3 to display the eligible entities and classes which may be
selected to be members of the class. The DISPLAY LIST MENU is
displayed.
Select option 4 to create a class member within the current class. The
CREATE CLASS NAME AND KIND NUMBER MENU is displayed.
Select option 5 to create an entity member within the current class. The
CREATE ENTITY MENU is displayed.
Select option 6 to model the class created. The CREATE MENU is
redisplayed.

(b)   Enter the kind number of a modeled class or entity that is to be made a
member of the class.

R2035A/7

4.2.7.4 Create Enumeration

The CREATE ENUMERATION menu allows a user to create the values of an enumeration.

```
  -------------------------CREATE ENUMERATION-----------------------

  COMMAND ===> (a)


          ┌─────────────────────────────────────────────────┐
          │ COMMANDS:                                        │
          │      1    : RETURN TO PREVIOUS MENU              │
          │      2    : EXIT TO CREATE MENU                  │
          │      3    : DISPLAY CURRENT MEMBERS OF ENUMERATION│
          │      4    : SAVE ENUMERATION CREATED             │
          │                                                  │
          │ ENTER EITHER A COMMAND (ABOVE) OR ENUMERATION    │
          │ MEMBER NAME(S) (BELOW).  NOTE:  AS EACH MEMBER IS│
          │ ACCEPTED IT VANISHES FROM THE SCREEN.            │
          └─────────────────────────────────────────────────┘

  MEMBER NAME ===> (b)
```

Figure 4-13. Create Enumeration

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the CREATE MENU.
Select option 3 to display the current members of the enumeration. The LIST MEMBERS MENU is displayed.
Select option 4 to model the created enumeration. The previous create menu is displayed.

(b) Enter the name of the enumeration items to be included in the enumeration. As each name is entered, the CREATE ENUMERATION MENU is redisplayed.

4.2.7.5 Create Integer

The CREATE INTEGER menu allows a user to define the precision of an integer.

```
------------------------------------CREATE INTEGER---------------------------

COMMAND ===> (a)



                    ┌─────────────────────────────────────┐
                    │ COMMANDS:                            │
                    │      1     : RETURN TO PREVIOUS MENU │
                    │      2     : EXIT TO CREATE MENU     │
                    │      3     : ACCEPT DEFAULT PRECISION│
                    │                                      │
                    │ ENTER EITHER A COMMAND (ABOVE OF     │
                    │ THE PRECISION OF THE INTEGER (BELOW).│
                    │                                      │
                    │ ACCEPTABLE PRECISION VALUES:  1-9    │
                    │ DEFAULT PRECISION VALUE:        9    │
                    └─────────────────────────────────────┘

                    ENTER PRECISION IN DECIMAL DIGITS ===> (b)
```

Figure 4-14. Create Integer

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the CREATE MENU.
Select option 3 to accept the default precision. The previous menu is redisplayed.

(b) Enter the integer precision in decimal digits.

4.2.7.6 Create List

The CREATE LIST menu allows a user to define the characteristics of a list.

```
-------------------------------CREATE LIST----------------------------
COMMAND ===> (a)

            COMMANDS:
                    1    : RETURN TO PREVIOUS MENU
                    2    : EXIT TO CREATE MENU

            ENTER EITHER A COMMAND (ABOVE) OR LIST
            DATA (BELOW).

            MINIMUM OCCURRENCES  ===> (b)
            MAXIMUM OCCURRENCES  ===> (c)
                    TYPE NUMBER  ===> (d)

                         TYPE NUMBERS
            1 : INTEGER    4 : LOGICAL    7 : SET
            2 : REAL       5 : ARRAY      8 : POINTER
            3 : STRING     6 : LIST       9 : DEFINED TYPE
```

Figure 4-15. Create List

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the CREATE MENU.

(b) Enter the minimum number of occurrences for the list.

(c) Enter the maximum number of occurrences for the list.

(d) Enter the list type number (1-9).

After the list data is accepted, the appropriate create type menu is displayed.
Possible create menus to be displayed include:

CREATE INTEGER   CREATE ARRAY   CREATE POINTER
CREATE REAL   CREATE LIST   CREATE DEFINED TYPE
CREATE STRING   CREATE SET

4.2.7.7 Create Pointer

The CREATE POINTER menu allows a user to define the members of a pointer.

```
------------------------------CREATE POINTER--------------------------

COMMAND ===> (a)


        +----------------------------------------------------+
        | COMMANDS:                                          |
        |     1    : RETURN TO PREVIOUS MENU                 |
        |     2    : EXIT TO CREATE MENU                     |
        |     3    : SELECT A KIND NUMBER FROM THE EXISTING  |
        |             CLASSES AND ENTITIES                   |
        |     4    : DISPLAY CURRENT MEMBERS IN THE POINTER  |
        |     5    : SAVE MEMBERS CREATED                    |
        |                                                    |
        | ENTER EITHER A COMMAND (ABOVE) OR MEMBER KIND      |
        | NUMBER (BELOW).  NOTE:  AS EACH NUMBER IS ACCEPTED |
        | IT VANISHES FROM THE SCREEN.                       |
        +----------------------------------------------------+

        KIND NUMBER ===> (b)
```

Figure 4-16. Create Pointer

Menu Description:

(a)   Select option 1 to redisplay the previous menu.

Select option 2 to redisplay the CREATE MENU.

Select option 3 to display the eligible entities and classes which may be selected to be referenced by the pointer. The DISPLAY LIST MENU is displayed.

Select option 4 to display the current member entities and classes referenced by the pointer. The LIST MEMBERS MENU is displayed.

Select option 5 to model the pointer created. The previous menu is redisplayed.

(b)   Enter the kind number of the entity or class that is to be referenced by the pointer.

4.2.7.8 Create Real

The CREATE REAL menu allows a user to define the precision of a real.


```
-----------------------------------CREATE REAL-----------------------------------

COMMAND ===> (a)



                    ┌─────────────────────────────────────────┐
                    │ COMMANDS:                                │
                    │      1     : RETURN TO PREVIOUS MENU      │
                    │      2     : EXIT TO CREATE MENU          │
                    │      3     : ACCEPT DEFAULT PRECISION     │
                    │                                          │
                    │ ENTER EITHER A COMMAND (ABOVE) OR         │
                    │ THE PRECISION OF THE REAL (BELOW).        │
                    │                                          │
                    │ ACCEPTABLE PRECISION VALUES:  1-16        │
                    │ DEFAULT PRECISION VALUE:        16        │
                    └─────────────────────────────────────────┘

                    ENTER PRECISION IN DECIMAL DIGITS ===> (b)
```

Figure 4-17. Create Real

Menu Description:

(a)   Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the CREATE MENU.
Select option 3 to accept default precision. The previous menu is redisplayed.

(b)   Enter the real precision in decimal digits.

4.2.7.9 Create Set

The CREATE SET menu allows a user to define the characteristics of a set.

```
-----------------------------------CREATE SET-----------------------------------
COMMAND ===> (a)

        ┌─────────────────────────────────────────────────┐
        │ COMMANDS:                                        │
        │      1    : RETURN TO PREVIOUS MENU              │
        │      2    : EXIT TO CREATE MENU                  │
        │                                                  │
        │ ENTER EITHER A COMMAND (ABOVE) OR SET            │
        │ DATA (BELOW).                                    │
        └─────────────────────────────────────────────────┘

        MINIMUM OCCURRENCES   ===> (b)
        MAXIMUM OCCURRENCES   ===> (c)
                 TYPE NUMBER  ===> (d)

        ┌─────────────────────────────────────────────────┐
        │                  TYPE NUMBERS                    │
        │  1 : INTEGER   4 : LOGICAL   7 : SET             │
        │  2 : REAL      5 : ARRAY     8 : POINTER         │
        │  3 : STRING    6 : LIST      9 : DEFINED TYPE    │
        └─────────────────────────────────────────────────┘
```

Figure 4-18. Create Set

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the CREATE MENU.

(b) Enter the minimum number of occurrences for the set.

(c) Enter the maximum number of occurrences for the set.

(d) Enter the list type number (1-9).

After the set data is accepted, the appropriate create type menu is displayed.
Possible create menus to be displayed include:

CREATE INTEGER	CREATE ARRAY	CREATE POINTER
CREATE REAL	CREATE LIST	CREATE DEFINED TYPE
CREATE STRING	CREATE SET	

R2085A/7

4.2.7.10 Create String

The CREATE STRING menu allows a user to define the length of a string.

```
 ----------------------------------CREATE STRING----------------------------------

COMMAND ===> (a)



                    |-----------------------------------------|
                    | COMMANDS:                               |
                    |        1      : RETURN TO PREVIOUS MENU  |
                    |        2      : EXIT TO CREATE MENU      |
                    |        3      : ACCEPT DEFAULT LENGTH    |
                    |                                         |
                    | ENTER EITHER A COMMAND (ABOVE) OR        |
                    | THE STRING LENGTH (BELOW).               |
                    |                                         |
                    | ACCEPTABLE LENGTH VALUES:  1-1000        |
                    | DEFAULT LENGTH VALUE:        1           |
                    |-----------------------------------------|

            ENTER LENGTH IN DECIMAL DIGITS ===> (b)
```

Figure 4-19. Create String

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the CREATE MENU.
Select option 3 to accept the default precision. The previous menu is redisplayed.

(b) Enter the string length in decimal digits.

4-23

4.2.7.11 Create/Reference Supertype

The CREATE/REFERENCE SUPERTYPE menu allows a user to either define the characteristics of a new supertype or to reference an existing supertype.

```
---------------------CREATE/REFERENCE SUPERTYPE---------------------

COMMAND ===> (a)


        +--------------------------------------------------+
        | COMMANDS:                                        |
        |       1     : RETURN TO PREVIOUS MENU            |
        |       2     : EXIT TO CREATE MENU                |
        |       3     : REFERENCE AN EXISTING SUPERTYPE    |
        |                                                  |
        | ENTER EITHER A COMMAND (ABOVE) OR SUPERTYPE      |
        | DATA (BELOW).                                    |
        +--------------------------------------------------+

        NAME                      ===> (b)
        REFERENCE SUPERTYPE (Y/N)? ===> (c)
```

Figure 4-20. Create/Reference Supertype

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the CREATE MENU.
     Select option 3 to display the DISPLAY LIST Menu.

(b) Enter the name of the supertype to be added to the model.

(c) Enter /Y' to display the CREATE/REFERENCE SUPERTYPE Menu.
     Enter /N' if this supertype does not refer to a parent supertype.

After the supertype name and supertype reference have been entered, the CREATE ATTRIBUTE MENU will be displayed.

4.2.7.12 Display List

The DISPLAY LIST menu allows a user to make a single selection from a list of displayed "entities".

```
 ---------------------------------DISPLAY LIST----------------------------

 COMMAND ===> (a)                                       SCROLL ===>


          +-----------------------------------------------------------+
          | COMMANDS:                                                 |
          |     1    : RETURN TO PREVIOUS MENU                        |
          |     2    : EXIT TO EDIT MENU                              |
          |                                                           |
          | OPTION:                                                   |
          |   SELECT AN ENTITY BY ENTERING ANY CHARACTER NEXT         |
          |   TO THE DESIRED MEMBER (AND BENEATH THE ASTERISK).       |
          |                                                           |
          | ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).      |
          +-----------------------------------------------------------+

          ENTITIES:         *
                           (b)
```

Figure 4-21. Display List

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the appropriate edit menu.

(b) A list of "entities" is displayed. Select an "entity" by entering any character next to the desired "entity's" name.

4.2.7.13 List Members

The LIST MEMBERS menu displays a list of "entities" for user review.

```
------------------------ LIST MEMBERS ------------------------

COMMAND ===> (a)                                    SCROLL ===>

          ┌─────────────────────────────────────────────┐
          │  COMMANDS:                                   │
          │       1     : RETURN TO PREVIOUS MENU        │
          │       2     : EXIT TO EDIT MENU              │
          │                                              │
          │  ENTER A COMMAND (ABOVE).                    │
          └─────────────────────────────────────────────┘

          MEMBERS:

                  (b)
```
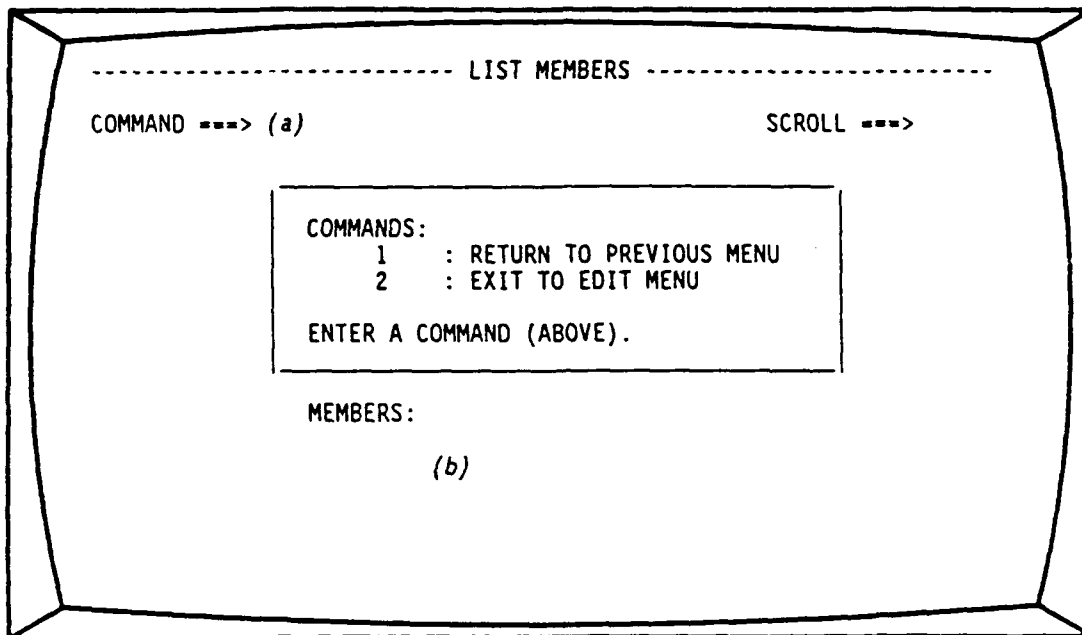
Figure 4-22. List Members

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the appropriate edit menu.

(b) A list of members is displayed.

4.3 UPDATE OPERATIONS

The UPDATE functions within the Schema Manager software are accessed from interactive menus. There are six update functions available directly from the main UPDATE MENU. Other supporting menus provide the remaining capabilities.

The update menus are listed in Table 4-2 along with a section reference to the detailed definitions.

TABLE 4-2

INTERACTIVE INTERFACE

UPDATE MENUS	SECTION
UPDATE ENTITY	4.3.1
UPDATE DEFINED TYPE	4.3.2
UPDATE SUPERTYPE	4.3.3
UPDATE GLOBAL ATTRIBUTE	4.3.4
UPDATE SUBSCHEMA	4.3.5
UPDATE CLASS	4.3.6
OTHER SUPPORTING MENUS	4.3.7
ADD ATTRIBUTE	4.3.7.1
ADD ENUMERATION ITEM	4.3.7.2
CREATE ARRAY	4.2.7.1
CREATE DEFINED TYPE	4.2.2
CREATE ENUMERATION	4.2.7.4
CREATE INTEGER	4.2.7.5
CREATE LIST	4.2.7.6
CREATE POINTER	4.2.7.7
CREATE REAL	4.2.7.8
CREATE SET	4.2.7.9
CREATE STRING	4.2.7.10
CREATE/REFERENCE SUPERTYPE	4.2.7.11
DISPLAY ATTRIBUTES	4.4.7.1
DISPLAY LIST	4.2.7.12
LIST MEMBERS	4.2.7.13
REVIEW ARRAY	4.4.7.2
REVIEW ATTRIBUTE	4.4.4
REVIEW CLASS	4.4.6
REVIEW DEFINED TYPE	4.4.2
REVIEW ENTITY	4.4.1
REVIEW ENUMERATION	4.4.7.3
REVIEW INTEGER	4.4.7.4
REVIEW LIST	4.4.7.5
REVIEW POINTER	4.4.7.6
REVIEW REAL	4.4.7.7
REVIEW SET	4.4.7.8
REVIEW STRING	4.4.7.9
REVIEW SUPERTYPE	4.4.3
UPDATE APPROVAL	4.3.7.3
UPDATE ARRAY	4.3.7.4
UPDATE ATTRIBUTE	4.3.7.5
UPDATE ATTRIBUTE DATA	4.3.7.6
UPDATE CLASS MEMBERS	4.3.7.7
UPDATE ENUMERATION	4.3.7.8
UPDATE INTEGER	4.3.7.9
UPDATE LIST	4.3.7.10
UPDATE POINTER	4.3.7.11
UPDATE REAL	4.3.7.12
UPDATE SET	4.3.7.13
UPDATE STRING	4.3.7.14
UPDATE SUBSCHEMA MEMBERS	4.3.7.15

R20856/7

Update Menu

The commands available from the UPDATE MENU allow a user to update an entity, defined type, supertype, global attribute, subschema, and class.

```
-------------------------------UPDATE MENU-----------------------------

COMMAND ===> (a)



             COMMANDS:
                  1    : RETURN TO MAIN MENU
                  2    : UPDATE ENTITY
                  3    : UPDATE DEFINED TYPE
                  4    : UPDATE SUPERTYPE
                  5    : UPDATE GLOBAL ATTRIBUTE
                  6    : UPDATE SUBSCHEMA
                  7    : UPDATE CLASS

             ENTER A COMMAND (ABOVE).
```

Figure 4-23. Update Menu

Menu Description:

(a) Select command 1 to redisplay the MAIN MENU.
Select command 2 to display the DISPLAY LIST MENU listing the modeled entities.
Select command 3 to display the DISPLAY LIST MENU listing the modeled defined types.
Select command 4 to display the DISPLAY LIST MENU listing the modeled supertypes.
Select command 5 to display the DISPLAY LIST MENU listing the modeled global attributes.
Select command 6 to display the DISPLAY LIST MENU listing the modeled subschemas.
Select command 7 to display the DISPLAY LIST MENU listing the modeled classes.

4.3.1 Update Entity

The UPDATE ENTITY menu allows a user to update the characteristics of an entity.

```
-----------------------------------UPDATE ENTITY-----------------------------------

COMMAND ===> (a)

                 ┌─────────────────────────────────────┐
                 │ COMMANDS:                            │
                 │      1    : RETURN TO PREVIOUS MENU   │
                 │      2    : EXIT TO UPDATE MENU       │
                 │      3    : DELETE THE ENTITY         │
                 │      4    : UPDATE LOCAL ATTRIBUTES   │
                 │      5    : UPDATE SUPERTYPE          │
                 │      6    : ADD SUPERTYPE REFERENCE   │
                 │   ENTER EITHER A COMMAND (ABOVE) OR   │
                 │   A CHANGE TO THE ENTITY DATA (BELOW).│
                 └─────────────────────────────────────┘

        NAME          : (b)                   ===> (c)
        KIND NUMBER : (d)                     ===> (e)

        COMMENTS:
        >>
        >>  (f)
        >>
```

Figure 4-24. Update Entity

Menu Description:

(a) Select command 1 to redisplay the previous menu.
Select command 2 to redisplay the UPDATE MENU.
Select command 3 to delete the entity from the model. The DISPLAY LIST
MENU will be redisplayed listing the remaining modeled entities.
Select command 4 to display the UPDATE ATTRIBUTE MENU.
Select command 5 to display the UPDATE SUPERTYPE MENU.

(b) The current entity name is displayed.

(c) The entity name may be updated.

(d) The current entity kind number is displayed.

(e) The entity kind number may be updated.

(f) Update remarks as appropriate (maximum 150 characters). These remarks,
if saved, will appear on the Conceptual Schema Report and the PASCAL
Include File.

4-30

4.3.2 Update Defined Type

The UPDATE DEFINED TYPE menu allows a user to update the characteristics of a defined type.

```
----------------------UPDATE DEFINED TYPE----------------------
COMMAND ===> (a)
                                                          •
        COMMANDS:
                1    : RETURN TO PREVIOUS MENU
                2    : EXIT TO UPDATE MENU
                3    : SAVE CHANGES MADE
                4    : UPDATE THE CURRENT TYPE : (b)
                5    : REVIEW THE CURRENT TYPE : (b)
        OPTION:
           ENTER EITHER A COMMAND (ABOVE) OR A CHANGE TO THE
           DEFINED TYPE DATA (BELOW).

        NAME   : (c)                ===> (d)
        TYPE   : (e)                ===> (f)

                          TYPE NUMBERS
        1 : INTEGER   4 : LOGICAL   7 : SET        10 : ENUMERATION
        2 : REAL      5 : ARRAY     8 : POINTER
        3 : STRING    6 : LIST      9 : DEFINED TYPE
```

Figure 4-25. Update Defined Type

Menu Description:

(a) Select command 1 to redisplay the previous menu.
Select command 2 to redisplay the UPDATE MENU.
Select command 3 to accept any changes made to the defined type.
Select command 4 to display the appropriate update type menu.
Possible update menus to be displayed include:

UPDATE INTEGER   UPDATE ARRAY   UPDATE POINTER
UPDATE REAL   UPDATE LIST   UPDATE DEFINED TYPE
UPDATE STRING   UPDATE SET   UPDATE ENUMERATION

Select command 5 to display the appropriate review type menu. Possible review menus to be displayed include:

REVIEW INTEGER   REVIEW ARRAY   REVIEW POINTER
REVIEW REAL   REVIEW LIST   REVIEW DEFINED TYPE
REVIEW STRING   REVIEW SET   REVIEW ENUMERATION

(b) The current defined type type is displayed.

4-31

(c) The current defined type name is displayed.

(d) The defined type name may be updated

(e) The current defined type type is displayed.

(f) The defined type type may be updated (1-9). If a new defined type type is entered, the appropriate create type menu is displayed. Possible create menus to be displayed include:

CREATE INTEGER MENU   CREATE ARRAY MENU   CREATE POINTER MENU
CREATE REAL MENU   CREATE LIST MENU   CREATE DEFINED TYPE MENU
CREATE STRING MENU   CREATE SET MENU   CREATE ENUMERATION MENU

4.3.3 Update Supertype

The UPDATE SUPERTYPE menu allows a user to update the characteristics of a supertype.

```
,-------------------------------------------------------------------.
|   ,-------------------------------------------------------------. |
|  /------------------------------UPDATE SUPERTYPE-----------------\|
| /                                                                 |
|/  COMMAND ===> (a)                                                |
||                                                                  |
||          ,-----------------------------------------.            |
||          | COMMANDS:                               |            |
||          |      1    : RETURN TO PREVIOUS MENU     |            |
||          |      2    : EXIT TO UPDATE MENU         |            |
||          |      3    : DELETE THE SUPERTYPE        |            |
||          |      4    : UPDATE LOCAL ATTRIBUTES     |            |
||          |      5    : UPDATE SUPERTYPE            |            |
||          |      6    : ADD SUPERTYPE REFERENCE     |            |
||          | OPTION:                                 |            |
||          |   ENTER EITHER A COMMAND (ABOVE) OR A   |            |
||          |   CHANGE TO THE SUPERTYPE NAME (BELOW). |            |
||          `-----------------------------------------'            |
||                                                                  |
|\     NAME        : (b)                    ===> (c)               /|
| \                                                               / |
|  `-------------------------------------------------------------'  |
`-------------------------------------------------------------------'
```

Figure 4-26.   Update Supertype

Menu Description:

(a)   Select command 1 to redisplay the previous menu.
Select command 2 to redisplay the UPDATE MENU.
Select command 3 to delete the entity from the model. The DISPLAY LIST MENU will be redisplayed listing the remaining modeled entities.
Select command 4 to display the UPDATE ATTRIBUTE MENU.
Select command 5 to display the UPDATE SUPERTYPE MENU.
Select command 6 to display the CREATE/REFERENCE SUPERTYPE MENU.

(b)   The current supertype name is displayed.

(c)   The supertype name may be updated.

4.3.4 Update Global Attribute

The UPDATE GLOBAL ATTRIBUTE menu allows a user to update the characteristics of a global attribute.

```
-------------------UPDATE GLOBAL ATTRIBUTE-------------------
COMMAND ===> (a)

       COMMANDS:
           1    : RETURN TO PREVIOUS MENU
           2    : EXIT TO UPDATE MENU
           3    : SAVE THE CHANGES MADE
           4    : UPDATE THE CURRENT TYPE : (b)
           5    : REVIEW THE CURRENT TYPE : (b)
           6    : DELETE THE GLOBAL ATTRIBUTE
       ENTER A COMMAND (ABOVE) OR A CHANGE TO THE ATTRIBUTE
       DATA (BELOW).
              GLOBAL
       NAME                    : (c)           ===> (d)
       REQUIRED/OPTIONAL       : (e)           ===> (f)    (R/O)
       POSITION                : (g)           ===> (h)    (INTEGER)
       TYPE NUMBER             : (i)           ===> (j)
       COMMENT>>  (k)

       TYPE      1 : INTEGER    4 : LOGICAL    7 : SET
       NUMBERS:  2 : REAL       5 : ARRAY      8 : DEFINED TYPE
                 3 : STRING     6 : LIST
```

Figure 4-27. Update Global Attribute

Menu Description:

(a) Select command 1 to redisplay the previous menu.

Select command 2 to redisplay the UPDATE MENU.

Select command 3 to accept any changes made to the global attribute.

Select command 4 to display the appropriate update type menu.

Possible update menus to be displayed include:

UPDATE INTEGER MENU   UPDATE ARRAY MENU   UPDATE DEFINED TYPE MENU
UPDATE REAL MENU   UPDATE LIST MENU
UPDATE STRING MENU   UPDATE SET MENU

R20858/7

Select command 5 to display the appropriate review type menu. Possible review menus to be displayed include:

REVIEW INTEGER MENU   REVIEW ARRAY MENU   REVIEW DEFINED TYPE MENU
REVIEW REAL MENU   REVIEW LIST MENU
REVIEW STRING MENU   REVIEW SET MENU

Select command 6 to delete the global attribute from the model. The DISPLAY LIST MENU will be redisplayed listing the remaining modeled global attributes.

(b) The current global attribute type is displayed.

(c) The current global attribute name is displayed.

(d) The global attribute name may be updated.

(e) The current required/optional characteristic of the global attribute is displayed.

(f) The required/optional characteristic of the global attribute may be updated.

(g) The current user specified physical position of the global attribute in the ADB is displayed.

(h) The user specified physical position number of the global attribute in the ADB may be updated.

(i) The current global attribute type is displayed.

(j) The global attribute type may be updated (1-9). If a new global type is entered, the appropriate create type menu is displayed. Possible create menus to be displayed include:

CREATE INTEGER MENU   CREATE ARRAY MENU   CREATE DEFINED TYPE MENU
CREATE REAL MENU   CREATE LIST MENU
CREATE STRING MENU   CREATE SET MENU

(k) Update remarks as appropriate (maximum 50 characters). These remarks, if saved, will appear on the Conceptual Schema Report and the PASCAL Include File.

4-35

4.3.5 Update Subschema

The UPDATE SUBSCHEMA menu allows a user to update the characteristics of a subschema.

```
------------------------------UPDATE SUBSCHEMA------------------------------

COMMAND ===> (a)


              +------------------------------------------------+
              | COMMANDS:                                      |
              |    1     : RETURN TO PREVIOUS MENU             |
              |    2     : EXIT TO UPDATE MENU                 |
              |    3     : DELETE THE SUBSCHEMA                |
              |    4     : REVIEW/ADD/DELETE MEMBERS           |
              | OPTION:                                        |
              |   ENTER EITHER A COMMAND (ABOVE) OR A          |
              |   CHANGE TO THE SUBSCHEMA NAME (BELOW).        |
              +------------------------------------------------+

              NAME : (b)                    ===> (c)
              COMMENTS:
              >>
              >>   (d)
              >>
```

Figure 4-28. Update Subschema

Menu Description:

(a) Select command 1 to redisplay the DISPLAY LIST MENU listing the modeled subschemas.
Select command 2 to redisplay the UPDATE MENU.
Select command 3 to delete the subschema from the model.
Select command 4 to review, add, or delete entity and class members of the subschema. The UPDATE SUBSCHEMA MEMBERS Menu is displayed.

(b) The current subschema name is displayed.

(c) The subschema name may be updated.

(d) Update remarks as appropriate (maximum 150 characters). These remarks, if saved, will appear on the Conceptual Schema Report and the PASCAL Include File.

4.3.6 Update Class

The UPDATE CLASS menu allows a user to update the characteristics of a class.

```
------------------------------UPDATE CLASS------------------------------
COMMAND ===> (a)

      COMMANDS:
            1    : RETURN TO PREVIOUS MENU
            2    : EXIT TO UPDATE MENU
            3    : DELETE THE CLASS
            4    : UPDATE/REVIEW/ADD/DELETE MEMBERS
      OPTION:
        ENTER EITHER A COMMAND (ABOVE) OR A CHANGE TO THE
        CLASS DATA (BELOW).

      CLASS : (b)

            NAME : (c)                    ===> (d)
            KIND : (e)                    ===> (f)

            COMMENTS:
            >>
            >>    (g)
            >>
```

Figure 4-29.   Update Class

Menu Description:

(a)   Select command 1 to redisplay the previous menu.
Select command 2 to redisplay the UPDATE MENU.
Select command 3 to delete the class from the model. The DISPLAY LIST
MENU is redisplayed listing the modeled classes.
Select command 4 to update, review, add, or delete a class member. The
UPDATE CLASS MEMBERS Menu is displayed.

(b)   The current class name is displayed.

(c)   The current class name is displayed.

(d)   The class name may be updated.

(e)   The current class kind number is displayed.

(f)   The class kind number may be updated.

(g)   Update remarks as appropriate (maximum 150 characters). These remarks,
if saved, will appear on the Conceptual Schema Report and the PASCAL
Include File.

4-37

4.3.7 Other Supporting Menus

The supporting menus listed below (alphabetically) are described in this section except as noted:

- Add Attribute
- Add Enumeration Item
- Create Array*
- Create Defined Type*
- Create Enumeration*
- Create Integer*
- Create List*
- Create Pointer*
- Create Real*
- Create Set*
- Create String*
- Create/Reference Supertype*
- Display Attributes**
- Display List*
- List Members*
- Review Array**
- Review Attribute**
- Review Class**
- Review Defined Type**
- Review Entity**
- Review Enumeration**

- Review Integer**
- Review List**
- Review Pointer**
- Review Real**
- Review Set**
- Review String**
- Review Supertype**
- Update Approval
- Update Array
- Update Attribute
- Update Attribute Data
- Update Class Members
- Update Enumeration
- Update Integer
- Update List
- Update Pointer
- Update Real
- Update Set
- Update String
- Update Subschema Members

*Described in the CREATE section.
**Described in the REVIEW section.

R2D85A/7

4.3.7.1 Add Attribute

The ADD ATTRIBUTE menu allows a user to define the characteristics of an attribute to be added to the model.

```
-------------------------ADD ATTRIBUTE------------------------
COMMAND ===> (a)

     ┌──────────────────────────────────────────────────────┐
     │ COMMAND:                                             │
     │     1     :   RETURN TO PREVIOUS MENU                │
     │                                                      │
     │ ENTER EITHER A COMMAND (ABOVE) OR ATTRIBUTE DATA     │
     │ (BELOW).                                             │
     └──────────────────────────────────────────────────────┘

       NAME                          ===> (b)
       REQUIRED/OPTIONAL (R/O)       ===> (c)
       POSITION (OPTIONAL INTEGER)   ===> (d)
       TYPE NUMBER                   ===> (e)
       COMMENT>>  (f)

       ┌────────────────────────────────────────────────────┐
       │                  TYPE NUMBERS                       │
       │   1 : INTEGER     4 : LOGICAL    7 : SET            │
       │   2 : REAL        5 : ARRAY      8 : POINTER        │
       │   3 : STRING      6 : LIST       9 : DEFINED TYPE   │
       └────────────────────────────────────────────────────┘
```

Figure 4-30. Add Attribute

Menu Description:

(a) Select option 1 to redisplay the previous menu.

(b) Enter the name of an attribute to be added to the entity.

(c) Enter "R" (required) or "O" (optional).

(d) Enter the user specified physical position number of the attribute in the ADB. (This entry is optional.)

(e) Enter the attribute type number (1-9). Possible create menus to be displayed include:

CREATE INTEGER   CREATE ARRAY   CREATE POINTER
CREATE REAL   CREATE LIST   CREATE DEFINED TYPE
CREATE STRING   CREATE SET

(f) Enter remarks as appropriate (maximum 50 characters). The comments will be saved and will appear on the Conceptual Schema Report and the PASCAL Include File.
After the attribute data are accepted, the appropriate create type menu is displayed.

4.3.7.2 Add Enumeration Item

The ADD ENUMERATION ITEM menu allows a user to add a value to an enumeration.

```
------------------------ADD ENUMERATION ITEM------------------------

COMMAND ---> (a)


        ┌─────────────────────────────────────────────────┐
        │ COMMANDS:                                        │
        │      1    : RETURN TO PREVIOUS MENU              │
        │      2    : EXIT TO UPDATE MENU                  │
        │                                                  │
        │ ENTER EITHER A COMMAND (ABOVE) OR AN ENUMERATION │
        │ ITEM NAME (BELOW).                               │
        └─────────────────────────────────────────────────┘

        ENUMERATION ITEM NAME ---> (b)
```

Figure 4-31. Add Enumeration Item

Menu Description:

(a) Select command 1 to redisplay the previous menu.
Select command 2 to redisplay the UPDATE MENU.

(b) Enter the names of the enumeration items to be added to the enumeration.

4.3.7.3 Update Approval

If a change has been made to an "entity" that would alter previously performed physicalization, the UPDATE APPROVAL menu is displayed.

```
---------------------------UPDATE APPROVAL-------------------------

COMMAND ===> (a)

        (b)  WAS PREVIOUSLY PHYSICALIZED.
      ┌─────────────────────────────────────────────────┐
      │                                                 │
      │  ANY CHANGES MADE MAY ALTER PHYSICALIZATION.    │
      │                                                 │
      └─────────────────────────────────────────────────┘


          ┌──────────────────────────────────────────┐
          │                                          │
          │  COMMANDS:                               │
          │    <ENTER>  : ACCEPT CHANGES MADE        │
          │        1    : REJECT CHANGES MADE        │
          │                                          │
          │  ENTER A COMMAND (ABOVE).                │
          │                                          │
          └──────────────────────────────────────────┘
```

Figure 4-32. Update Approval

Menu Description:

(a) Press the <ENTER> key to accept the changes made.
Select option 1 to reject the changes made.

(b) Specifies what has been previously physicalized.

4.3.7.4 Update Array

The UPDATE ARRAY menu allows a user to update the characteristics of an array.

```
----------------------------------UPDATE ARRAY----------------------------------
COMMAND ===> (a)

    ┌──────────────────────────────────────────────────────────────────┐
    │ COMMANDS:                                                          │
    │     1     : RETURN TO PREVIOUS MENU                                │
    │     2     : EXIT TO UPDATE MENU                                    │
    │     3     : SAVE CHANGES MADE TO THE ARRAY                         │
    │     4     : UPDATE THE CURRENT TYPE : (b)                          │
    │     5     : REVIEW THE CURRENT TYPE : (b)                          │
    │ OPTION:                                                            │
    │     ENTER EITHER A COMMAND (ABOVE) OR A CHANGE TO THE              │
    │     ARRAY DATA (BELOW).                                            │
    └──────────────────────────────────────────────────────────────────┘
    ARRAY :
                    LOW BOUND : (c)          ===> (d)
                    HIGH BOUND: (e)          ===> (f)
                    TYPE      : (g)          ===> (h)
    ┌──────────────────────────────────────────────────────────────────┐
    │                        TYPE NUMBERS                                │
    │     1 : INTEGER    4 : LOGICAL     7 : SET                         │
    │     2 : REAL       5 : ARRAY       8 : POINTER                     │
    │     3 : STRING     6 : LIST        9 : DEFINED TYPE                │
    └──────────────────────────────────────────────────────────────────┘
```

Figure 4-33. Update Array

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the UPDATE MENU.
Select option 3 to accept any changes made to the array.
Select option 4 to display the appropriate update type menu. Possible
update menus to be displayed include:

UPDATE INTEGER MENU	UPDATE ARRAY MENU	UPDATE POINTER MENU
UPDATE REAL MENU	UPDATE LIST MENU	UPDATE DEFINED TYPE MENU
UPDATE STRING MENU	UPDATE SET MENU	

R20434/7

Select option 5 to display the appropriate review type menu. Possible review menus to be displayed include:

REVIEW INTEGER MENU   REVIEW ARRAY MENU   REVIEW POINTER MENU
REVIEW REAL MENU   REVIEW LIST MENU   REVIEW DEFINED TYPE MENU
REVIEW STRING MENU   REVIEW SET MENU

(b) The current array type is displayed.

(c) The current array lower bound is displayed.

(d) The array lower bound may be updated.

(e) The current array upper bound is displayed.

(f) The array upper bound may be updated.

(g) The current array type is displayed.

(h) The array type may be changed (1-9). If a new array type is entered, the appropriate create type menu is displayed. Possible create menus to be displayed include:

CREATE INTEGER MENU   CREATE ARRAY MENU   CREATE POINTER MENU
CREATE REAL MENU   CREATE LIST MENU   CREATE DEFINED TYPE MENU
CREATE STRING MENU   CREATE SET MENU

4.3.7.5 Update Attribute Menu

The UPDATE ATTRIBUTE menu allows a user to add an attribute to the model or to select an attribute to update.

```
-----------------------------UPDATE ATTRIBUTE--------------------------

COMMAND ===> (a)                                    SCROLL ===>


      +--------------------------------------------------+
      | COMMANDS:                                        |
      |       1     : RETURN TO PREVIOUS MENU            |
      |       2     : EXIT TO UPDATE MENU                |
      |       3     : ADD AN ATTRIBUTE TO THE (b)        |
      |       4     : SAVE CHANGES MADE TO THE (b)       |
      | OPTION:                                          |
      |   ENTER A CHARACTER CODE NEXT TO THE DESIRED MEMBER |
      |   (AND BENEATH THE ASTERISK).                    |
      |   'D' DELETE, 'R' REVIEW, OR 'U' UPDATE THE ATTRIBUTE. |
      |                                                  |
      |   ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW). |
      +--------------------------------------------------+
NAME:    (c)                            KIND NUMBER:    (d)
                  *
          (e)
```

Figure 4-34. Update Attribute

Menu Description:

(a) Select option 1 to redisplay the previous menu.
    Select option 2 to redisplay the UPDATE MENU.
    Select option 3 to add an attribute to the entity. The ADD ATTRIBUTE
    MENU is displayed.
    Select option 4 to accept any changes made to the entity. The DISPLAY
    LIST MENU will be redisplayed listing the modeled entities.

(b) ENTITY or SUPERTYPE as appropriate.

(c) The entity or supertype name.

(d) The entity kind number.

(e) The entity attributes are displayed.
    Enter /D' next to an attribute that is to be deleted from the model.
    Enter /R' next to an attribute that is to be reviewed. The REVIEW
    ATTRIBUTE MENU is displayed.
    Enter /U' next to an attribute that is to be updated. The UPDATE
    ATTRIBUTE DATA MENU is displayed.

4.3.7.6 Update Attribute Data

The UPDATE ATTRIBUTE DATA menu allows a user to update the characteristics of an attribute.

```
----------------------UPDATE ATTRIBUTE DATA----------------------
COMMAND ===> (a)

        COMMANDS:
            1   : RETURN TO PREVIOUS MENU
            2   : EXIT TO UPDATE MENU
            3   : SAVE THE CHANGES MADE
            4   : UPDATE THE CURRENT TYPE : (b)
            5   : REVIEW THE CURRENT TYPE : (b)
        OPTION:
    ENTER A COMMAND (ABOVE) OR A CHANGE TO THE ATTRIBUTE DATA
    (BELOW).

        ENTITY ATTRIBUTE
    NAME                    : (c)            ===> (d)
    REQUIRED/OPTIONAL       : (e)            ===> (f)     (R/O)
    POSITION                : (g)            ===> (h)     (INTEGER)
    TYPE NUMBER             : (i)            ===> (j)

    TYPE        1 : INTEGER    4 : LOGICAL    7 : SET
    NUMBERS:    2 : REAL       5 : ARRAY      8 : POINTER
                3 : STRING     6 : LIST       9 : DEFINED TYPE
```

Figure 4-35. Update Attribute Date

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the UPDATE MENU.
Select option 3 to accept any changes made to the attribute.
Select option 4 to display the appropriate update type menu. Possible update menus to be displayed include:

UPDATE INTEGER   UPDATE ARRAY   UPDATE POINTER
UPDATE REAL   UPDATE LIST   UPDATE DEFINED TYPE
UPDATE STRING   UPDATE SET

Select option 5 to display the appropriate review type menu. Possible review menus to be displayed include:

REVIEW INTEGER   REVIEW ARRAY   REVIEW POINTER
REVIEW REAL   REVIEW LIST   REVIEW DEFINED TYPE
REVIEW STRING   REVIEW SET

(b) The current attribute type is displayed.

4-45

(c) The current attribute name is displayed.

(d) The attribute name may be updated.

(e) The current required/optional characteristic of the attribute is displayed.

(f) The required/optional characteristic of the attribute may be updated.

(g) The current user specified physical position of the attribute in the ADB is displayed.

(h) The user specified physical position of the attribute in the ADB may be updated.

(i) The current attribute type is displayed.

(j) The attribute type may be updated (1-9). If a new attribute type is entered, the appropriate create type menu is displayed. Possible create menus to be displayed include:

CREATE INTEGER MENU	CREATE ARRAY MENU	CREATE POINTER MENU
CREATE REAL MENU	CREATE LIST MENU	CREATE DEFINED TYPE MENU
CREATE STRING MENU	CREATE SET MENU	

4.3.7.7 Update Class Members

The UPDATE CLASS MEMBERS menu allows a user to update the members of a class.

```
-------------------------UPDATE CLASS MEMBERS---------------------
COMMAND ===> (a)                                     SCROLL ===>

    ┌──────────────────────────────────────────────────────┐
    │ COMMANDS:                                            │
    │     1    : RETURN TO PREVIOUS MENU                   │
    │     2    : EXIT TO UPDATE MENU                       │
    │     3    : ADD A MEMBER TO THE CLASS                 │
    │     4    : SAVE CHANGES MADE TO THE CLASS            │
    │                                                      │
    │ OPTION:                                              │
    │    ENTER A CHARACTER CODE NEXT TO THE DESIRED MEMBER │
    │    (AND BENEATH THE ASTERISK).                       │
    │     'D' DELETE OR 'R' REVIEW THE MEMBER              │
    │                                                      │
    │ ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW). │
    └──────────────────────────────────────────────────────┘

    NAME: (b)                    KIND NUMBER: (c)
    MEMBERS :                *
                            (d)
```

Figure 4-36. Update Class

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the UPDATE MENU.
     Select option 3 to add a member to the class. The DISPLAY LIST MENU
     is displayed listing the eligible entities and classes.
     Select option 4 to accept any changes made to the class. The DISPLAY
     LIST MENU is redisplayed listing the modeled classes.

(b) The current class name is displayed.

(c) The current class kind number is displayed.

(d) The class members are displayed.
     Enter /D' next to a member that is to be deleted from the model.
     Enter /R' next to a member that is to be reviewed. The REVIEW ENTITY
     or REVIEW CLASS MENU is displayed.

4.3.7.8 Update Enumeration

The UPDATE ENUMERATION menu allows a user to update the values of an enumeration.

```
 --------------------------UPDATE ENUMERATION--------------------------

  COMMAND ===> (a)                                        SCROLL ===>


          ┌─────────────────────────────────────────────────────────┐
          │  COMMANDS:                                                │
          │      1    : RETURN TO PREVIOUS MENU                       │
          │      2    : EXIT TO UPDATE MENU                           │
          │      3    : ADD ITEM TO THE ENUMERATION                   │
          │      4    : SAVE CHANGES MADE TO THE ENUMERATION          │
          │                                                           │
          │  OPTION:                                                  │
          │    ENTER A CHARACTER CODE NEXT TO THE DESIRED MEMBER      │
          │    (AND BENEATH THE ASTERISK).                            │
          │                  'D' TO DELETE THE MEMBER                 │
          │  ENTER A COMMAND (ABOVE).                                 │
          └─────────────────────────────────────────────────────────┘

          ENUMERATION ITEMS:    *
                               (b)
```

Figure 4-37.   Update Enumeration

Menu Description:

(a)   Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the UPDATE MENU.
Select option 3 to add an enumeration item to the enumeration. The ADD ENUMERATION ITEM MENU will be displayed.
Select option 4 to accept any changes made to the enumeration.

(b)   The enumeration items are displayed.
Enter /D' next to an item that is to be deleted from the model.

4.3.7.9 Update Integer

The UPDATE INTEGER menu allows a user to update the precision of an integer.

```
-------------------------------UPDATE INTEGER-------------------------------

COMMAND ===> (a)


        COMMANDS:
            1     : RETURN TO PREVIOUS MENU
            2     : EXIT TO UPDATE MENU
            3     : ACCEPT DEFAULT PRECISION

        OPTION:
          ENTER EITHER A COMMAND (ABOVE) OR A CHANGE TO THE
          INTEGER PRECISION (BELOW).
          ACCEPTABLE PRECISION VALUES:  1-9
          DEFAULT PRECISION VALUE:         9


        PRECISION IN DECIMAL DIGITS: (b)        ===> (c)
```

Figure 4-38. Update Integer

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the UPDATE MENU.
Select option 3 to accept the default precision.

(b) The current integer precision in decimal digits is displayed.

(c) The integer precision may be updated.

4.3.7.10 Update List

The UPDATE LIST menu allows a user to update the characteristics of a list.

```
-----------------------------------UPDATE LIST-----------------------------------

 COMMAND ===> (a)

        COMMANDS:
              1    : RETURN TO PREVIOUS MENU
              2    : EXIT TO UPDATE MENU
              3    : SAVE CHANGES MADE TO THE LIST
              4    : UPDATE THE CURRENT TYPE : (b)
              5    : REVIEW THE CURRENT TYPE : (b)
        OPTION:
           ENTER EITHER A COMMAND (ABOVE) OR A CHANGE TO THE
           LIST DATA (BELOW).

        LIST :
           MINIMUM OCCURRENCES : (c)          ===> (d)
           MAXIMUM OCCURRENCES : (e)          ===> (f)
                         TYPE : (g)           ===> (h)

                           TYPE NUMBERS
           1 : INTEGER    4 : LOGICAL    7 : SET
           2 : REAL       5 : ARRAY      8 : POINTER
           3 : STRING     6 : LIST       9 : DEFINED TYPE
```

Figure 4-39. Update List

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the UPDATE MENU.
Select option 3 to accept the changes made to the list.
Select option 4 to display the appropriate update type menu. Possible update menus to be displayed include:

UPDATE INTEGER MENU	UPDATE ARRAY MENU	UPDATE POINTER MENU
UPDATE REAL MENU	UPDATE LIST MENU	UPDATE DEFINED TYPE MENU
UPDATE STRING MENU	UPDATE SET MENU	

Select option 5 to display the appropriate review type menu. Possible review menus to be displayed include:

REVIEW INTEGER MENU	REVIEW ARRAY MENU	REVIEW POINTER MENU
REVIEW REAL MENU	REVIEW LIST MENU	REVIEW DEFINED TYPE MENU
REVIEW STRING MENU	REVIEW SET MENU	

(b) The current list type is displayed.

(c) The current list minimum occurrences is displayed.

(d) The list minimum occurrences may be updated.

(e) The current list maximum occurrences is displayed.

(f) The list maximum occurrences may be updated.

(g) The current list type is displayed.

(h) The list type may be changed (1-9). If a new list type is entered, the appropriate create type menu is displayed. Possible create menus to be displayed include:

CREATE INTEGER MENU   CREATE ARRAY MENU   CREATE POINTER MENU
CREATE REAL MENU   CREATE LIST MENU   CREATE DEFINED TYPE MENU
CREATE STRING MENU   CREATE SET MENU

4.3.7.11 Update Pointer

The UPDATE POINTER menu allows a user to update the members of a pointer.

```
-----------------------------UPDATE POINTER----------------------------
COMMAND ===> (a)                                      SCROLL ===>

     COMMANDS:
          1    : RETURN TO PREVIOUS MENU
          2    : EXIT TO UPDATE MENU
          3    : ADD A MEMBER TO THE POINTER
          4    : SAVE CHANGES MADE TO THE POINTER

     OPTION:
       ENTER A CHARACTER CODE NEXT TO THE DESIRED MEMBER
       (AND BENEATH THE ASTERISK).
                   'D' TO DELETE THE MEMBER
                   'R' TO REVIEW THE MEMBER

     ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).


       MEMBERS:          *
                        (b)
```

Figure 4-40. Update Pointer

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the UPDATE MENU.
Select option 3 to add a reference to an entity or class. The DISPLAY LIST MENU is displayed listing the eligible entities and classes.
Select option 4 to accept any changes made to the pointer.

(b) The entities and classes referenced by the pointer are displayed.
Enter /D' next to a member that is to be deleted from the model.
Enter /R' next to a member that is to be reviewed. The REVIEW ENTITY or REVIEW CLASS MENU is displayed.

R30844/7

4.3.7.12 Update Real

The UPDATE REAL menu allows a user to update the precision of a real.

```
---------------------------------UPDATE REAL---------------------------

COMMAND ===> (a)



        COMMANDS:
            1    : RETURN TO PREVIOUS MENU
            2    : EXIT TO UPDATE MENU
            3    : ACCEPT DEFAULT PRECISION
        OPTION:
          ENTER EITHER A COMMAND (ABOVE) OR A CHANGE TO THE
          REAL PRECISION (BELOW).
          ACCEPTABLE PRECISION VALUES:  1-16
          DEFAULT PRECISION VALUE:       16


    PRECISION IN DECIMAL DIGITS : (b)        ===> (c)
```

Figure 4-41. Update Real

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the UPDATE MENU.
Select option 3 to accept the default precision.

(b) The current real precision in decimal digits is displayed.

(c) The real precision may be updated.

4.3.7.13 Update Set

The UPDATE SET Menu allows a user to update the characteristics of a set.

```
---------------------------------UPDATE SET----------------------------------
COMMAND ===> (a)
    ┌──────────────────────────────────────────────────────────────┐
    │ COMMANDS:                                                      │
    │      1    : RETURN TO PREVIOUS MENU                            │
    │      2    : EXIT TO UPDATE MENU                                │
    │      3    : SAVE CHANGES MADE TO THE SET                       │
    │      4    : UPDATE THE CURRENT TYPE :  (b)                     │
    │      5    : REVIEW THE CURRENT TYPE :  (b)                     │
    │ OPTION:                                                        │
    │    ENTER EITHER A COMMAND (ABOVE) OR A CHANGE TO THE           │
    │    LIST DATA (BELOW).                                          │
    └──────────────────────────────────────────────────────────────┘
    SET :
      MINIMUM OCCURRENCES : (c)          ===> (d)
      MAXIMUM OCCURRENCES : (e)          ===> (f)
                     TYPE : (g)          ===> (h)
      ┌──────────────────────────────────────────────────────┐
      │                   TYPE NUMBERS                        │
      │    1 : INTEGER    4 : LOGICAL    7 : SET              │
      │    2 : REAL       5 : ARRAY      8 : POINTER          │
      │    3 : STRING     6 : LIST       9 : DEFINED TYPE     │
      └──────────────────────────────────────────────────────┘
```

Figure 4-42. Update Set

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the UPDATE MENU.
Select option 3 to accept the changes made to the set.
Select option 4 to display the appropriate update type menu. Possible update menus to be displayed include:

UPDATE INTEGER MENU	UPDATE ARRAY MENU	UPDATE POINTER MENU
UPDATE REAL MENU	UPDATE LIST MENU	UPDATE DEFINED TYPE MENU
UPDATE STRING MENU	UPDATE SET MENU	

Select option 5 to display the appropriate review type menu. Possible review menus to be displayed include:

REVIEW INTEGER MENU	REVIEW ARRAY MENU	REVIEW POINTER MENU
REVIEW REAL MENU	REVIEW LIST MENU	REVIEW DEFINED TYPE MENU
REVIEW STRING MENU	REVIEW SET MENU	

(b) The current set type is displayed.

(c) The current set minimum occurrences is displayed.

(d) The set minimum occurrences may be updated.

(e) The current set maximum occurrences is displayed.

(f) The set maximum occurrences may be updated.

(g) The current set type is displayed.

(h) The set type may be changed (1-9). If a new set type is entered, the appropriate create type menu is displayed. Possible create menus to be displayed include:

CREATE INTEGER MENU   CREATE ARRAY MENU   CREATE POINTER MENU
CREATE REAL MENU   CREATE LIST MENU   CREATE DEFINED TYPE MENU
CREATE STRING MENU   CREATE SET MENU

4.3.7.14 Update String

The UPDATE STRING menu allows a user to update the length of a string.

```
-------------------------------UPDATE STRING------------------------------

COMMAND ===> (a)


        ┌─────────────────────────────────────────────────────────┐
        │ COMMANDS:                                                │
        │     1     : RETURN TO PREVIOUS MENU                      │
        │     2     : EXIT TO UPDATE MENU                          │
        │     3     : ACCEPT DEFAULT PRECISION                     │
        │                                                          │
        │ OPTION:                                                  │
        │   ENTER EITHER A COMMAND (ABOVE) OR A CHANGE TO THE      │
        │   STRING LENGTH (BELOW).                                 │
        │   ACCEPTABLE LENGTH VALUES:  1-1000                      │
        │   DEFAULT LENGTH VALUE:        1                         │
        └─────────────────────────────────────────────────────────┘

        LENGTH IN DECIMAL DIGITS : (b)        ===> (c)
```

Figure 4-43. Update String

Menu Description:

(a)   Select option 1 to redisplay the previous menu.
      Select option 2 to redisplay the UPDATE MENU.
      Select option 3 to accept the default precision.

(b)   The current string length in decimal digits is displayed.

(c)   The string length may be updated.

4.3.7.15 Update Subschema Members

The UPDATE SUBSCHEMA MEMBERS menu allows a user to update the members of a subschema.

```
-------------------------UPDATE SUBSCHEMA MEMBERS--------------------

COMMAND ===> (a)                                          SCROLL ===>

        COMMANDS:
             1      : RETURN TO PREVIOUS MENU
             2      : EXIT TO UPDATE MENU
             3      : ADD A MEMBER TO THE SUBSCHEMA
             4      : SAVE CHANGES MADE TO THE SUBSCHEMA

        OPTION:
          ENTER A CHARACTER CODE NEXT TO THE DESIRED MEMBER
          (AND BENEATH THE ASTERISK).
          'D' TO DELETE OR  'R' TO REVIEW THE MEMBER

        ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).


        NAME    : (b)
        MEMBERS :              *
                            (c)
```

Figure 4-44. Update Subschema Members

Menu Description:

   (a) Select option 1 to redisplay the previous menu.
       Select option 2 to redisplay the UPDATE MENU.
       Select option 3 to add an entity or class to the subschema. The DISPLAY LIST MENU is displayed listing the eligible entities and classes.
       Select option 4 to accept any changes made to the subschema. The DISPLAY LIST MENU will be redisplayed listing the modeled subschemas.

   (b) The subschema name will be displayed.

   (c) The entity and class members of the subschema will be displayed. Enter /D' next to a member that is to be deleted from the model. Enter /R' next to a member that is to be reviewed. The REVIEW ENTITY or REVIEW CLASS MENU is displayed.

4.4 REVIEW OPERATIONS

The REVIEW functions within the Schema Manager software are accessed from interactive menus. There are six functions available directly from the main REVIEW menu. Other supporting menus provide the remaining capabilities.

The review menus are listed in Table 4-3 along with a section reference to the detailed definitions.

TABLE 4-3

INTERACTIVE INTERFACE REVIEW MENUS

	Section
REVIEW ENTITY	4.4.1
REVIEW DEFINED TYPE	4.4.2
REVIEW SUPERTYPE	4.4.3
REVIEW GLOBAL ATTRIBUTE	4.4.4
REVIEW SUBSCHEMA	4.4.5
REVIEW CLASS	4.4.6
OTHER SUPPORTING MENUS	4.4.7
DISPLAY ATTRIBUTES	4.4.7.1
DISPLAY LIST	4.2.7.12
REVIEW ARRAY	4.4.7.2
REVIEW ATTRIBUTE	4.4.4
REVIEW ENUMERATION	4.4.7.3
REVIEW INTEGER	4.4.7.4
REVIEW LIST	4.4.7.5
REVIEW POINTER	4.4.7.6
REVIEW REAL	4.4.7.7
REVIEW SET	4.4.7.8
REVIEW STRING	4.4.7.9

R2035A/7

R2035A/7

Review Menu

The commands available from the REVIEW MENU allow a user to review an entity, defined type, supertype, global attribute, subschema, or class.

```
-----------------------------REVIEW MENU----------------------------

COMMAND ===> (a)


                  COMMANDS:
                     1    : RETURN TO MAIN MENU
                     2    : REVIEW ENTITY
                     3    : REVIEW DEFINED TYPE
                     4    : REVIEW SUPERTYPE
                     5    : REVIEW GLOBAL ATTRIBUTE
                     6    : REVIEW SUBSCHEMA
                     7    : REVIEW CLASS

                  ENTER A COMMAND (ABOVE).
```

Figure 4-45. Review Menu

Menu Description:

(a) Select option 1 to redisplay the MAIN MENU.
Select option 2 to display the DISPLAY LIST MENU listing the modeled entities.
Select option 3 to display the DISPLAY LIST MENU listing the modeled defined types.
Select option 4 to display the DISPLAY LIST MENU listing the modeled supertypes.
Select option 5 to display the DISPLAY LIST MENU listing the modeled global attributes.
Select option 6 to display the DISPLAY LIST MENU listing the modeled subschemas.
Select option 7 to display the DISPLAY LIST MENU listing the modeled classes.

4.4.1 Review Entity

The REVIEW ENTITY menu allows a user to review the characteristics of an entity.

```
-----------------------------REVIEW ENTITY-----------------------------

COMMAND ===> (a)



          ┌──────────────────────────────────────┐
          │ COMMANDS:                             │
          │     1     : RETURN TO PREVIOUS MENU   │
          │     2     : EXIT TO EDIT MENU         │
          │     3     : REVIEW LOCAL ATTRIBUTES   │
          │     4     : REVIEW SUPERTYPE          │
          │                                       │
          │ ENTER EITHER A COMMAND (ABOVE).       │
          └──────────────────────────────────────┘

          NAME : (b)                  KIND:  (c)
          COMMENTS:
          >>
          >>  (d)
          >>
```

Figure 4-46. Review Entity

Menu Description:

    (a)   Select option 1 to redisplay the previous menu.
            Select option 2 to redisplay the appropriate edit menu.
            Select option 3 to display the DISPLAY ATTRIBUTES MENU.
            Select option 4 to display the REVIEW SUPERTYPE MENU.

    (b)   The entity name will be displayed.

    (c)   The entity kind number will be displayed.

    (d)   The entity remarks will be displayed.

4.4.2 Review Defined Type

The REVIEW DEFINED TYPE menu allows a user to review the characteristics of a defined type.

```
-----------------------REVIEW DEFINED TYPE-----------------------

OPTION ---> (a)


           COMMANDS:
                1     : RETURN TO PREVIOUS MENU
                2     : EXIT TO REVIEW MENU
                3     : REVIEW THE TYPE

           ENTER A COMMAND (ABOVE).


           DEFINED TYPE:

                          NAME: (b)
                          TYPE: (c)
```

Figure 4-47. Review Defined Type

Menu Description

    (a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the appropriate edit menu.
Select option 3 to display the appropriate review type menu. Possible review menus to be displayed include:

REVIEW INTEGER MENU   REVIEW ARRAY MENU   REVIEW POINTER MENU
REVIEW REAL MENU   REVIEW LIST MENU   REVIEW DEFINED TYPE MENU
REVIEW STRING MENU   REVIEW SET MENU   REVIEW ENUMERATION MENU

    (b) The defined type name is displayed.

    (c) The defined type type is displayed.

4.4.3 Review Supertype

The REVIEW SUPERTYPE menu allows a user to review the characteristics of a supertype.


```
------------------------------REVIEW SUPERTYPE------------------------------

COMMAND ---> (a)


        ┌─────────────────────────────────────────────┐
        │ COMMANDS:                                     │
        │       1    : RETURN TO PREVIOUS MENU          │
        │       2    : EXIT TO EDIT MENU                │
        │       3    : REVIEW LOCAL ATTRIBUTES          │
        │       4    : REVIEW SUPERTYPE                 │
        │                                               │
        │ ENTER EITHER A COMMAND (ABOVE).               │
        └─────────────────────────────────────────────┘

        NAME : (b)
```

Figure 4-48.   Review Supertype

Menu Description:

(a)   Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the appropriate edit menu.
Select option 3 to display the DISPLAY ATTRIBUTES MENU.
Select option 2 to display the REVIEW SUPERTYPE MENU.

(b)   The supertype name will be displayed.

4-62

4.4.4 REVIEW GLOBAL ATTRIBUTE

The REVIEW GLOBAL ATTRIBUTE menu allows a user to review the characteristics of a global attribute.

```
------------------------------REVIEW ATTRIBUTE----------------------------

 OPTION ===> (a)


              COMMANDS:
                   1      : RETURN TO PREVIOUS MENU
                   2      : EXIT TO REVIEW MENU
                   3      : REVIEW THE TYPE

              ENTER A COMMAND (ABOVE).


                (b)
                   ATTRIBUTE:

                   NAME               : (c)
                   POSITION           : (d)
                   REQUIRED/OPTIONAL  : (e)
                   TYPE               : (f)
                   COMMENT >>         : (g)
```

Figure 4-49. Review Attribute

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the appropriate edit menu.
Select option 3 to display the appropriate review type menu.

Possible review menus to be displayed include:

REVIEW INTEGER MENU   REVIEW ARRAY MENU   REVIEW DEFINED TYPE MENU
REVIEW REAL MENU   REVIEW LIST MENU
REVIEW STRING MENU   REVIEW SET MENU

(b) The "global" or "entity" keyword is displayed.

(c) The attribute name will be displayed.

(d) The user specified physical position number of the attribute in the ADB is displayed.

(e) The attribute required/optional characteristic is displayed.

4-63

R2065A/7

(f)   The attribute type is displayed.

(g)   The attribute remarks are displayed.

4.4.5 Review Subschema

The REVIEW SUBSCHEMA menu allows a user to review the characteristics of a subschema.

```
--------------------------REVIEW SUBSCHEMA --------------------------

COMMAND ===> (a)                                 SCROLL ===>


        ┌─────────────────────────────────────────────────────────┐
        │ COMMANDS:                                                │
        │     1      : RETURN TO PREVIOUS MENU                      │
        │     2      : EXIT TO REVIEW MENU                          │
        │ OPTION:                                                   │
        │    REVIEW A SUBSCHEMA MEMBER BY ENTERING ANY CHARACTER    │
        │    NEXT TO THE DESIRED MEMBER (AND BENEATH THE ASTERISK). │
        │ ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).      │
        └─────────────────────────────────────────────────────────┘
          COMMENTS:
          >>
          >>   (b)
          >>
          MEMBERS :                   *
                                    (c)
```

Figure 4-50. Review Subschema

Menu Description:

(a)   Select option 1 to redisplay the previous menu.
      Select option 2 to redisplay the review menu.

(b)   The subschema remarks are displayed.

(c)   A list of the modeled subschema members is displayed. A subschema member may be reviewed by entering any character next to the desired member's name. The REVIEW ENTITY or REVIEW CLASS MENU is displayed.

4.4.6 Review Class

The REVIEW CLASS menu allows a user to review the characteristics of a class.

```
-------------------------------REVIEW CLASS----------------------------

COMMAND ===> (a)                                    SCROLL ===>


    ┌───────────────────────────────────────────────────────────┐
    │ COMMANDS:                                                   │
    │      1    : RETURN TO PREVIOUS MENU                          │
    │      2    : EXIT TO REVIEW MENU                              │
    │ OPTION:                                                      │
    │   REVIEW A CLASS MEMBER BY ENTERING ANY CHARACTER NEXT       │
    │   TO THE DESIRED MEMBER (AND BENEATH THE ASTERISK).          │
    │ ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).         │
    └───────────────────────────────────────────────────────────┘
    CLASS: (b)                            KIND: (c)
    COMMENTS:
    >>
    >>    (d)
    >>
    MEMBERS:                  *
                             (e)
```

Figure 4-51.   Review Class

Menu Description:

(a)   Select option 1 to redisplay the previous menu.
      Select option 2 to redisplay the appropriate edit menu.

(b)   The class name is displayed.

(c)   The class kind number is displayed.

(d)   The class remarks are displayed.

(e)   A list of the modeled class members is displayed. A class member may be
      reviewed by entering any character next to the desired members's name.
      The REVIEW ENTITY MENU or REVIEW CLASS MENU is displayed.

4.4.7 Other Supporting Menus

The supporting menus alphabetically listed below are described in this section except as noted:

- Display Attributes
- Display List*
- Review Array
- Review Enumeration
- Review Integer
- Review List
- Review Pointer
- Review Real
- Review Set
- Review String.

*Described in the CREATE section.

4.4.7.1 Display Attributes

The DISPLAY ATTRIBUTES menu allows a user to make a single selection from a list of displayed attributes.

```
-----------------------------DISPLAY ATTRIBUTES-----------------------

COMMAND ===> (a)


    ┌─────────────────────────────────────────────────────────────┐
    │ COMMANDS:                                                     │
    │     1    : RETURN TO PREVIOUS MENU                            │
    │     2    : EXIT TO REVIEW MENU                                │
    │                                                               │
    │ OPTION:                                                       │
    │   REVIEW AN ATTRIBUTE BY ENTERING ANY CHARACTER NEXT          │
    │   TO THE DESIRED ATTRIBUTE·(AND BENEATH THE ASTERISK).        │
    │                                                               │
    │ ENTER EITHER A COMMAND (ABOVE) OR SELECT AN OPTION (BELOW).   │
    └─────────────────────────────────────────────────────────────┘

    MEMBERS:
            (b)
```

Figure 4-52. Display Attributes

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the appropriate edit menu.

(b) A list of the attributes will be displayed. An attribute may be reviewed by entering any character next to the desired attribute's name. The REVIEW ATTRIBUTE Menu is displayed.

4-67

4.4.7.2 Review Array

The REVIEW ARRAY menu allows a user to review the characteristics of an array.

```
-----------------------------REVIEW ARRAY-----------------------------

OPTION ===> (a)


        +---------------------------------------+
        | COMMANDS:                             |
        |     1      : RETURN TO PREVIOUS MENU  |
        |     2      : EXIT TO REVIEW MENU      |
        |     3      : REVIEW THE TYPE          |
        |                                       |
        | ENTER A COMMAND (ABOVE).              |
        +---------------------------------------+


        ARRAY DATA:

                    LOW BOUND:   (b)
                    HIGH BOUND:  (c)
                    TYPE:        (d)
```

Figure 4-53. Review Array

Menu Description:

    (a)   Select option 1 to redisplay the previous menu.
           Select option 2 to redisplay the appropriate edit menu.
           Select option 3 to display the appropriate review type menu. Possible review
           menus to be displayed include:

REVIEW INTEGER MENU   REVIEW ARRAY MENU   REVIEW POINTER MENU
REVIEW REAL MENU   REVIEW LIST MENU   REVIEW DEFINED TYPE MENU
REVIEW STRING MENU   REVIEW SET MENU

    (b)   The array low bound is displayed.

    (c)   The array high bound is displayed.

    (d)   The array type is displayed.

4.4.7.3 Review Enumeration

The REVIEW ENUMERATION menu allows a user to review the values of an enumeration.

```
--------------------------REVIEW ENUMERATION------------------------

COMMAND ===> (a)                                SCROLL ===>

         .
         +--------------------------------------------------+
         |                                                  |
         |  COMMANDS:                                       |
         |       1     : RETURN TO PREVIOUS MENU            |
         |       2     : EXIT TO REVIEW MENU                |
         |                                                  |
         |  ENTER A COMMAND (ABOVE).                        |
         +--------------------------------------------------+


         MEMBERS:
                   (b)
```

Figure 4-54. Review Enumeration

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the appropriate edit menu.

(b) A list of the enumeration items is displayed.

4.4.7.4 Review Integer

The REVIEW INTEGER menu allows a user to review the precision of an integer.

```
------------------------------REVIEW INTEGER-------------------------------

COMMAND ===> (a)



            ┌─────────────────────────────────────────┐
            │ COMMANDS:                                │
            │      1      : RETURN TO PREVIOUS MENU     │
            │      2      : EXIT TO REVIEW MENU         │
            │                                          │
            │ ENTER A COMMAND (ABOVE).                  │
            └─────────────────────────────────────────┘

            PRECISION IN DECIMAL DIGITS : (b)
```

Figure 4-55. Review Integer

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the appropriate edit menu.

(b) The integer precision in decimal digits is displayed.

4.4.7.5 Review List

The REVIEW LIST menu allows a user to review the characteristics of a list.

```
-------------------------------REVIEW LIST-----------------------------

OPTION ===> (a)



            ┌─────────────────────────────────────────┐
            │ COMMANDS:                                │
            │     1      : RETURN TO PREVIOUS MENU      │
            │     2      : EXIT TO REVIEW MENU          │
            │     3      : REVIEW THE TYPE              │
            │                                          │
            │ ENTER A COMMAND (ABOVE).                  │
            └─────────────────────────────────────────┘

            LIST DATA:

                    MINIMUM OCCURRENCES: (b)
                    MAXIMUM OCCURRENCES: (c)
                                  TYPE: (d)
```

Figure 4-56. Review List

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the appropriate edit menu.
Select option 3 to display the appropriate review type menu:

REVIEW INTEGER MENU	REVIEW ARRAY MENU	REVIEW POINTER MENU
REVIEW REAL MENU	REVIEW LIST MENU	REVIEW DEFINED TYPE MENU
REVIEW STRING MENU	REVIEW SET MENU	

(b) The list minimum occurrences is displayed.

(c) The list maximum occurrences is displayed.

(d) The list type is displayed.

4.4.7.6 Review Pointer

The REVIEW POINTER menu allows a user to review the members of a pointer.

```
---------------------------REVIEW POINTER---------------------------

COMMAND ===> (a)                                    SCROLL ===>

        +--------------------------------------------------------+
        | COMMANDS:                                              |
        |     1    : RETURN TO REVIEW MENU                       |
        |     2    : EXIT TO EDIT MENU                           |
        |                                                        |
        | OPTION:                                                |
        |   REVIEW A POINTER BY ENTERING ANY CHARACTER NEXT TO   |
        |   THE DESIRED MEMBER (AND BENEATH THE ASTERISK).       |
        |                                                        |
        | ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).   |
        +--------------------------------------------------------+

        MEMBERS :                  *
                                  (b)
```

Figure 4-57. Review Pointer

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the appropriate edit menu.

(b) A list of the pointer references (entities and classes) is displayed. Review a
pointer reference by entering any character next to the desired member's
name. The REVIEW ENTITY or REVIEW CLASS MENU is displayed.

4.4.7.7 Review Real

The REVIEW REAL menu allows a user to review the precision of a real.

```
+----------------------------------------------------------+
|                                                          |
|  -------------------------REVIEW REAL------------------------  |
|                                                          |
|  COMMAND ===> (a)                                        |
|                                                          |
|            +--------------------------------------+      |
|            | COMMANDS:                             |      |
|            |    1      : RETURN TO PREVIOUS MENU   |      |
|            |    2      : EXIT TO REVIEW MENU       |      |
|            |                                       |      |
|            | ENTER A COMMAND (ABOVE).              |      |
|            +--------------------------------------+      |
|                                                          |
|            PRECISION IN DECIMAL DIGITS : (b)             |
|                                                          |
|                                                          |
+----------------------------------------------------------+
```

Figure 4-58. Review Real

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the appropriate edit menu.

(b) The real precision in decimal digits is displayed.

4.4.7.8 Review Set

The REVIEW SET menu allows a user to review the characteristics of a set.

```
-----------------------------REVIEW SET-----------------------------

OPTION ===> (a)


            ┌─────────────────────────────────────────┐
            │ COMMANDS:                                │
            │     1     : RETURN TO PREVIOUS MENU      │
            │     2     : EXIT TO REVIEW MENU          │
            │     3     : REVIEW THE TYPE              │
            │                                          │
            │ ENTER A COMMAND (ABOVE).                 │
            └─────────────────────────────────────────┘

            SET DATA:

                    MINIMUM OCCURRENCES: (b)
                    MAXIMUM OCCURRENCES: (c)
                                   TYPE: (d)
```

Figure 4-59. Review Set

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to redisplay the appropriate edit menu.
Select option 3 to display the appropriate review type menu. Possible review menus to be displayed include:

REVIEW INTEGER MENU   REVIEW ARRAY MENU   REVIEW POINTER MENU
REVIEW REAL MENU   REVIEW LIST MENU   REVIEW DEFINED TYPE MENU
REVIEW STRING MENU   REVIEW SET MENU

(b) The set minimum occurrences is displayed.

(c) The set maximum occurrences is displayed.

(d) The set type is displayed.

4-74

4.4.7.9 Review String

The REVIEW STRING menu allows a user to review the length of a string.

```
-------------------------------REVIEW STRING----------------------------

COMMAND ===> (a)


            ┌─────────────────────────────────────────┐
            │ COMMANDS:                                │
            │      1      : RETURN TO PREVIOUS MENU     │
            │      2      : EXIT TO REVIEW MENU         │
            │                                          │
            │ ENTER A COMMAND (ABOVE).                 │
            └─────────────────────────────────────────┘

            LENGTH IN DECIMAL DIGITS : (b)
```

Figure 4-60. Review String

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to redisplay the appropriate edit menu.

(b) The string length in decimal digits is displayed.

4.5 GENERATE REPORTS

The REPORT functions within the Schema Manager software can be accessed from interactive menus. There are six options available directly from the main REPORT menu. Other supporting menus provide the remaining capabilities.

The report menus are listed in Table 4-4 with a section reference to the detailed definitions.

TABLE 4-4

INTERACTIVE INTERFACE REPORT MENUS

	Section
PASCAL INCLUDE FILES	4.5.1
PHYSICAL SUBSCHEMA REPORT	4.5.2
RUNTIME SUBSCHEMA REPORT	4.5.3
DATA DICTIONARY	4.5.4
CROSS REFERENCE MAIN MENU	4.5.5
DATA TYPE SPECIFICATION	4.5.5.1
PRECISION SPECIFICATION	4.5.5.2
LENGTH SPECIFICATION	4.5.5.3
SPECIFICATIONS	4.5.5.4
CROSS REFERENCE REPORT	4.5.5.5

R20358/7

The commands available from the REPORT MENU allow a user to generate any of the following: Conceptual Schema Report, PASCAL Include File, Physical Subschema Report, Runtime Subschema File, Data Dictionary, and Cross Reference capabilities.

```
 -------------------------------REPORT MENU---------------------------

COMMAND ===> (a)



        COMMANDS:
              1    : RETURN TO MAIN MENU
              2    : CONCEPTUAL SCHEMA REPORT
              3    : PASCAL INCLUDE FILE
              4    : PHYSICAL SUBSCHEMA REPORT
              5    : RUNTIME SUBSCHEMA FILE
              6    : DATA DICTIONARY
              7    : CROSS REFERENCE

        ENTER A COMMAND (ABOVE).
```
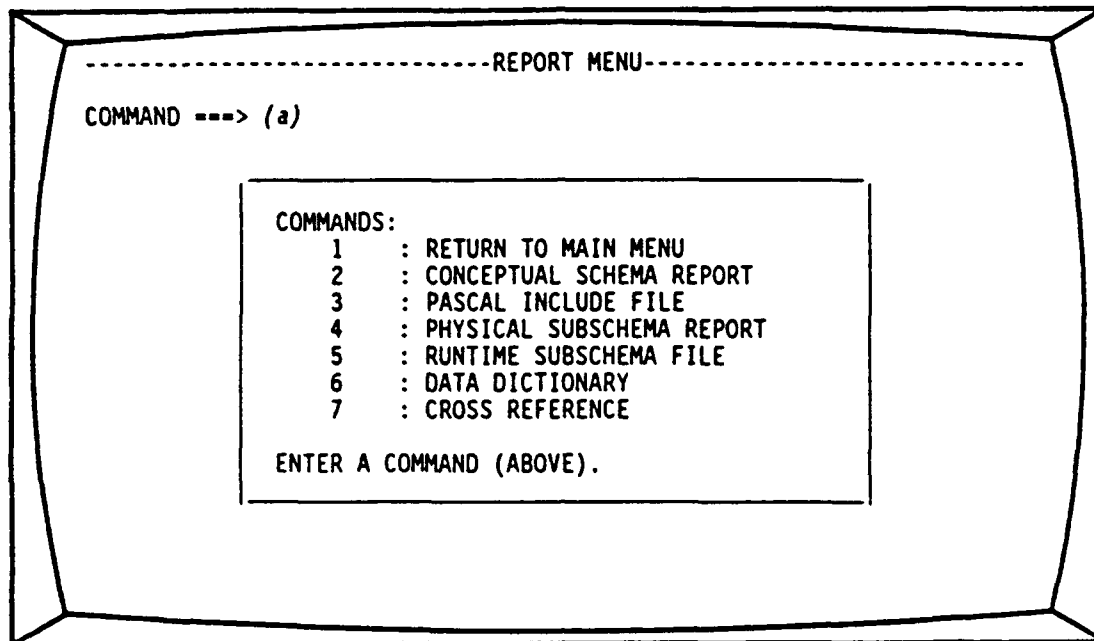
Figure 4-61. Report Menu

Menu Description:

(a) Select option 1 to redisplay the MAIN MENU.

Select option 2 to generate a Conceptual Schema Report. After the report is generated, the message /FILE CREATED' is displayed in the upper right hand corner.

Select option 3 to display the PASCAL INCLUDE FILE MENU.

Select option 4 to display the PHYSICAL SUBSCHEMA REPORT MENU.

Select option 5 to display the RUNTIME SUBSCHEMA REPORT MENU.

Select option 6 to display the DATA DICTIONARY MENU.

Select option 7 to display the CROSS REFERENCE MAIN MENU.

R20354/7

4.5.1 PASCAL Include File

The PASCAL INCLUDE FILES menu allows a user to select an existing subschema for which a file is to be generated. After the file is generated, the message /FILE CREATED' is displayed in the upper right hand corner of the screen.

```
------------------------- PASCAL INCLUDE FILES ------------------------

COMMAND ===> (a)                                    SCROLL ===>

        +------------------------------------------------------+
        | COMMANDS:                                            |
        |     1     : RETURN TO PREVIOUS MENU                  |
        |                                                      |
        | OPTION:                                              |
        |   CHOOSE THE SUBSCHEMA FOR WHICH THE PASCAL INCLUDE  |
        |   FILES ARE TO BE GENERATED.                         |
        |                                                      |
        | ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW). |
        +------------------------------------------------------+

        MEMBERS :                    *
                                    (b)
```

Figure 4-62. PASCAL Include File

Menu Description:

(a)   Select option 1 to redisplay the REPORT MENU.

(b)   A list of the modeled subschemas is displayed. Generate a PASCAL Include File for a subschema by entering any character next to the desired subschema's name.

4.5.2 Physical Subschema Report

The PHYSICAL SUBSCHEMA REPORT menu allows a user to select an existing subschema for which a report is to be generated. After the report is generated, the message /FILE CREATED' is displayed in the upper right hand corner of the screen.

```
-------------------- PHYSICAL SUBSCHEMA REPORT --------------------

COMMAND ===> (a)                                        SCROLL ===>


          +----------------------------------------------------+
          |  COMMANDS:                                         |
          |      1      : RETURN TO PREVIOUS MENU              |
          |                                                    |
          |  OPTION:                                           |
          |    CHOOSE THE SUBSCHEMA FOR WHICH THE PHYSICAL SUB-|
          |    SCHEMA REPORT IS TO BE GENERATED.               |
          |                                                    |
          |  ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).|
          +----------------------------------------------------+

          MEMBERS :                  *
                                    (b)
```

Figure 4-63. Physical Subschema Report

Menu Description:

(a) Select option 1 to redisplay the REPORT MENU.

(b) A list of the modeled subschemas is displayed. Generate a Physical Subschema Report for a subschema by entering any character next to the desired subschema's name.

4-79

4.5.3 Runtime Subschema File

The RUNTIME SUBSCHEMA REPORT menu allows a user to select an existing subschema for which a report is to be generated. After the report is generated, the message /FILE CREATED' is displayed in the upper right hand corner of the screen.

```
------------------ RUNTIME SUBSCHEMA REPORT -----------------------

COMMAND ===> (a)                                    SCROLL ===>


        COMMANDS:
              1    : RETURN TO PREVIOUS MENU

        OPTION:
          CHOOSE THE SUBSCHEMA FOR WHICH THE RUNTIME SUB-
          SCHEMA FILE IS TO BE GENERATED.

        ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).


        MEMBERS :              *
                             (b)
```

Figure 4-64. Runtime Subschema Report

Menu Description:

(a) Select option 1 to redisplay the REPORT MENU.

(b) A list of the modeled subschemas is displayed. Generate a Runtime Subschema Report for a subschema by entering any character next to the desired subschema's name.

4.5.4 Data Dictionary

The DATA DICTIONARY menu allows a user to select an existing subschema for which a file is to be generated. After the file is generated, the message /FILE CREATED' is displayed in the upper right hand corner of the screen.

```
------------------------DATA DICTIONARY------------------------

COMMAND ===> (a)                                    SCROLL ===>

     ┌─────────────────────────────────────────────────────┐
     │  COMMANDS:                                           │
     │      1     : RETURN TO PREVIOUS MENU                 │
     │                                                      │
     │  OPTION:                                             │
     │    CHOOSE THE SUBSCHEMA FOR WHICH THE DATA DICTIONARY │
     │    IS TO BE GENERATED.                               │
     │                                                      │
     │  ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW). │
     └─────────────────────────────────────────────────────┘

     MEMBERS :                    *
                                 (b)
```

Figure 4-65. Data Dictionary

Menu Description:

(a) Select option 1 to redisplay the REPORT MENU.

(b) A list of the modeled subschemas is displayed. Generate a Data Dictionary for a subschema by entering any character next to the desired subschema's name.

4.5.5 Cross Reference Main Menu

The CROSS REFERENCE MAIN MENU allows a user to query an existing model.

```
--------------------CROSS REFERENCE MAIN MENU--------------------

COMMAND ---->  (a)


    COMMANDS :  (ENTER ONE ABOVE)
        1  : RETURN TO PREVIOUS MENU
        2  : LIST ALL ATTRIBUTES OF A PARTICULAR DATA TYPE
        3  : LIST ALL ATTRIBUTES OF A SPECIFIED DEFINED TYPE
        4  : LIST ALL ENTITIES HAVING AN ATTRIBUTE WITH A PARTICULAR
             NAME
        5  : LIST ALL CLASSES CONTAINING A PARTICULAR ENTITY
        6  : LIST ALL SUBSCHEMAS CONTAINING A PARTICULAR ENTITY
        7  : LIST ALL SUBSCHEMAS CONTAINING A PARTICULAR CLASS
        8  : LIST ALL EXISTING PRECISIONS FOR THE INTEGER DATA TYPE
        9  : LIST ALL EXISTING PRECISIONS FOR THE REAL DATA TYPE
       10  : LIST ALL EXISTING LENGTHS FOR THE STRING DATA TYPE
```

Figure 4-66. Cross Reference Main Menu

Menu Description:

(a) Select option 1 to redisplay the REPORT MENU.
     Select option 2 to display the DATA TYPE SPECIFICATION MENU.
     Select option 3 to display the SPECIFICATION MENU.
     Select option 4 to display the SPECIFICATION MENU.
     Select option 5 to display the SPECIFICATION MENU.
     Select option 6 to display the SPECIFICATION MENU.
     Select option 7 to display the SPECIFICATION MENU.
     Select option 8 to display the CROSS REFERENCE REPORT MENU.
     Select option 9 to display the CROSS REFERENCE REPORT MENU.
     Select option 10 to display the CROSS REFERENCE REPORT MENU.

The Cross Reference Report Function accesses five menus within the Schema Manager. These five menus, described in the following sections, include:

- Data Type Specification Menu
- Precision Specification Menu
- Length Specification Menu
- Specification Menu
- Cross Reference Report Menu.

4.5.5.1 Data Type Specification

The DATA TYPE SPECIFICATION menu allows a user to specify the data type for which a search is to be conducted.

```
 ------------------------DATA TYPE SPECIFICATION----------------------
OPTION ===> (a)

                     ATTRIBUTES OF A PARTICULAR DATA TYPE

                   ┌─────────────────────────────────┐
                   │         SPECIFY DATA TYPE        │
                   │        (ENTER ONE ABOVE)         │
                   │                                  │
                   │         1.  RETURN               │
                   │         2.  INTEGER              │
                   │         3.  REAL                 │
                   │         4.  STRING               │
                   │         5.  LOGICAL              │
                   │         6.  ENUMERATION          │
                   │         7.  POINTER              │
                   │         8.  DEFINED TYPE         │
                   │         9.  ARRAY                │
                   │        10.  LIST                 │
                   │        11.  SET                  │
                   │                                  │
                   └─────────────────────────────────┘
```

Figure 4-67. Data Type Specification

Menu Description:

(a) Select option 1 to redisplay the CROSS REFERENCE MAIN MENU.
Select option 2 to display the PRECISION SPECIFICATION MENU.
Select option 3 to display the PRECISION SPECIFICATION MENU.
Select option 4 to display the LENGTH SPECIFICATION MENU.
Select option 5 to display the CROSS REFERENCE REPORT MENU.
Select option 6 to display the CROSS REFERENCE REPORT MENU.
Select option 7 to display the CROSS REFERENCE REPORT MENU.
Select option 8 to display the CROSS REFERENCE REPORT MENU.
Select option 9 to display the CROSS REFERENCE REPORT MENU.
Select option 10 to display the CROSS REFERENCE REPORT MENU.
Select option 11 to display the CROSS REFERENCE REPORT MENU.

R20446/7

4.5.5.2 Precision Specification Menu

The PRECISION SPECIFICATION menu allows a user to specify the precision for which a search is to be conducted.

```
-------------------------PRECISION SPECIFICATION----------------------

COMMAND ===> (a)

                                    (b)

                            SPECIFY PRECISION

              ┌──────────────────────────────────────────┐
              │ COMMANDS:                                 │
              │  <ENTER>  : ALL PRECISIONS                │
              │     1     : RETURN TO PREVIOUS MENU       │
              │                                           │
              │  ENTER EITHER A COMMAND (ABOVE) OR        │
              │  SPECIFY AN INTEGER PRECISION (BELOW)     │
              └──────────────────────────────────────────┘

          SPECIFY PRECISION IN DECIMAL DIGITS ===> (c)
```

Figure 4-68.   Precision Specification

Menu Description:

    (a)   Press the <ENTER> key to display all existing integer or real precisions. Select option 1 to redisplay the previous menu.

    (b)   A description of the type of precision (integer or real).

    (c)   Specify a precision in integer digits. If the request is satisfied, the CROSS REFERENCE REPORT MENU will be displayed.

4.5.5.3 Length Specification

The LENGTH SPECIFICATION menu allows a user to specify the string length for which a search is to be conducted.

```
-----------------------------LENGTH SPECIFICATION-----------------------------

COMMAND ===> (a)

                     LIST ALL ATTRIBUTES OF TYPE STRING

                              SPECIFY LENGTH

              +--------------------------------------------+
              | COMMANDS:                                  |
              |  <ENTER>  : ALL LENGTHS                    |
              |     1     : RETURN TO PREVIOUS MENU        |
              |                                            |
              | ENTER EITHER A COMMAND (ABOVE) OR          |
              | SPECIFY A STRING LENGTH (BELOW)            |
              +--------------------------------------------+

              ENTER STRING LENGTH ===> (b)
```

Figure 4-69. Length Specification

Menu Description:

(a) Press the <ENTER> key to display all existing string lengths. Select option 1 to redisplay the previous menu.

(b) Specify a string length in integer digits. If the request is satisfied, the CROSS REFERENCE REPORT MENU will be displayed.

4.5.5.4 Specification

The SPECIFICATION menu allows a user to specify a name or kind number for which a search is to be conducted.

```
----------------------------------SPECIFICATION----------------------------

COMMAND ===>   (a)


              ┌─────────────────────────────────────────┐
              │ COMMANDS:                                │
              │      1    : RETURN TO PREVIOUS MENU      │
              │                                          │
              │ ENTER EITHER A COMMAND (ABOVE) OR DATA   │
              │ (BELOW).                                 │
              └─────────────────────────────────────────┘

              (b)

              (c)
              >> (d)
```

Figure 4-70. Specification

Menu Description:

(a) Select option 1 to redisplay the previous menu.

(b) A description of the request selected from the previous menu.

(c) A description of the name or kind number to be entered.

(d) Specify a name or kind number for which a search is to be conducted. If the request is satisfied, the CROSS REFERENCE REPORT Menu will be displayed.

4.5.5.5 Cross Reference Report

The CROSS REFERENCE REPORT menu displays for user review the results of the query performed.

```
-----------------------CROSS REFERENCE REPORT----------------------

COMMAND ===> (a)                                    SCROLL ===>

        ┌──────────────────────────────────────────────────┐
        │ COMMANDS:                                         │
        │     1    : RETURN TO SPECIFICATION MENU           │
        │     2    : EXIT TO CROSS REFERENCE MAIN MENU      │
        │                                                   │
        │ ENTER A COMMAND (ABOVE).                          │
        └──────────────────────────────────────────────────┘

        (b)


                              (c)
```

Figure 4-71. Cross Reference Report

Menu Description:

(a) Select option 1 to redisplay the SPECIFICATION Menu.
     Select option 2 to redisplay the CROSS REFERENCE MAIN MENU.

(b) A description of the list created.

(c) Appropriate headings and the results satisfying the request are displayed.

4.6 FILE MODEL TO DISK

If command 7, **File Model to Disk,** is selected from the MAIN MENU, the model is filed to disk. The message /MODEL FILED' is displayed in the upper right corner of the MAIN MENU.

4.7 EXIT SCHEMA MANAGER

If command 1, **Exit Schema Manager,** is selected from the MAIN MENU, the REPLACE OPTION MENU is displayed. This menu allows a user to replace the model stored on disk with the current working model.

```
------------------------------REPLACE OPTION MENU-------------------------


        SHOULD THE CURRENT WORKING MODEL REPLACE THE
        MODEL STORED ON DISK?  (a)  (ENTER 'Y' or 'N')
```

Figure 4-72. Replace Option Menu

Menu Description:

(a) Enter /Y' to file a model in working form to the disk. After the model is filed, an option is given to print reports. Enter /N' for an option to print reports without filing the model in working form to disk.

SECTION 5

GENERATED REPORTS AND FILES

The Schema Manager software REPORT MENU has the options to generate the following six reports or files:

- Conceptual Schema Report
- PASCAL Include File
- Physical Subschema Report
- Runtime Subschema Files
- PDDI/GMAP Data Dictionary
- Cross Reference Report.

The content and format of these reports and files are described in this section.

5.1 CONCEPTUAL SCHEMA REPORT

The Conceptual Schema Report (CSR) shows all the definitions in the schema model, which include:

- Defined types
- Global attributes
- Inherited attribute groups (supertypes)
- Entities
- Classes
- Subschemas.

The report also contains indices to the definitions for inherited attributes, entities, classes, and subschemas. The format of the report entries is based on the EXPRESS information modeling language.

R20354/8

5.1.1 Defined Types

Each defined type, as illustrated in the example, is equivalent to one of the following data types:

- Primitive (integer, real, string, logical, pointer)

- Complex primitive (enumeration)

- Aggregation (array, list, set)

- Another defined type based on a primitive, complex primitive, or aggregation data type.

The keyword TYPE is followed by the defined type name. An equal sign serves as the delimiter between the name and type. A semicolon ends each definition.

```
--PAGE 1


(*------------------------------------------------------------------*)
(*                    DEFINED TYPE DEFINITIONS                      *)
(*------------------------------------------------------------------*)

TYPE DOUBLE_REAL      = REAL(15);

TYPE FULL_PARTIAL     = ENUMERATION OF
                        (FULL              ,
                          PARTIAL          );

TYPE NSE_CODES        = ENUMERATION OF
                        (REQUIRED          ,
                         OPTIONL           ,
                         UOS               ,
                         GLOBL             ,
                         EXCEPTION         ,
                         ALTERNATIVE       ,
                         INFORMATION       );

TYPE RB_SPLINE_CODE   = ENUMERATION OF
                        (UNDETERMINED      ,
                         LINEAR_ARC        ,
                         CIRCULAR_ARC      ,
                         ELLIPTICAL_ARC    ,
                         PARABOLIC_ARC     ,
                         HYPERBOLIC_ARC    );

TYPE SHORT_INTEGER    = INTEGER(4);
```

5.1.2 Global Attributes

Each global attribute, illustrated below, is defined by a data type. The keyword GLOBAL is followed by the global attribute name. A colon serves as the delimiter between the name and type. INTEGER and REAL data types are followed by the number of digits of precision enclosed in parenthesis; the STRING data type is followed by the string length in decimal digits, enclosed in parenthesis. A semi-colon ends each definition.

```
--PAGE 2


(*-----------------------------------------------------------------*)
(*                  GLOBAL ATTRIBUTE DEFINITIONS                   *)
(*-----------------------------------------------------------------*)

GLOBAL IDENT          : INTEGER(8);
```

5.1.3 Supertype

Each inherited attribute group (supertype), illustrated below, is defined by the list of attributes. The keyword SUPERTYPE is followed by the supertype name and a semicolon. The attributes are then defined. Each attribute name is followed by a colon and the attribute type. The supertype definition is concluded with the keyword END_SUPERTYPE and a semicolon.

```
--PAGE 3


(*-----------------------------------------------------------------*)
(*                     SUPERTYPE DEFINITION                        *)
(*-----------------------------------------------------------------*)

SUPERTYPE GROUPING        ;
      X             :   REAL(16);
      Y             :   REAL(16);
      Z             :   REAL(16);
      END_SUPERTYPE;
```

An inherited attribute group may also refer to another inherited attribute group, which is higher in the hierarchy. In this case, the supertype name is followed by the keywords SUBTYPE OF and the higher supertype name enclosed in parentheses, as illustrated below.

```
--PAGE 4


(*----------------------------------------------------------------*)
(*                      SUPERTYPE DEFINITION                      *)
(*----------------------------------------------------------------*)

SUPERTYPE COLLECTOR        SUBTYPE OF (GROUPING        );
    A                    :  STRING(10);
    B                    :  LOGICAL;
    END_SUPERTYPE;
```

5.1.4 Entities

Each entity is defined by the list of its attributes, as illustrated below. The keyword ENTITY is followed by the entity name, entity kind number enclosed in parentheses, and a semicolon. The entity attributes are then defined. Each attribute name is followed by a colon, the attribute data type, and a semicolon. The entity definition is concluded with the keyword END_ENTITY and a semicolon.

```
--PAGE 5


(*----------------------------------------------------------------*)
(*                       ENTITY DEFINITION                        *)
(*----------------------------------------------------------------*)

ENTITY AIRFLOW            (938) ;
    APPLICATION         :  NSE_CODES        ;
    PROP_REFS           :  LIST(0..255) OF POINTER TO
                           (PROPRTY         );
    INSP_CONTROLS       :  LIST(1..10) OF POINTER TO
                           (INSP_CONTROL    );
    AIRFLOW_KIND        :  FULL_PARTIAL     ;
    MAX_PRES_RATIO      :  DOUBLE_REAL      ;
    MIN_PRES_RATIO      :  DOUBLE_REAL      ;
    AIRFLOW_PARM        :  DOUBLE_REAL      ;
    BLOCKS              :  LIST(0..255) OF POINTER TO
                           (BLOCK           );
    END_ENTITY;
```

An entity may also refer to an inherited attribute group. In this case, the keywords SUBTYPE OF and the supertype name enclosed in parentheses follows the entity kind number, as illustrated below.

```
--PAGE 6


(*---------------------------------------------------------------------*)
(*                        ENTITY DEFINITION                            *)
(*---------------------------------------------------------------------*)

ENTITY RB_SPLINE2      (225) SUBTYPE OF (GROUPING        );
     CRV_IND       :  RB_SPLINE_CODE  ;
     DEGREE        :  SHORT_INTEGER   ;
     KNOTS         :  LIST(2..125) OF DOUBLE_REAL    ;
     NO_KNOTS      :  SHORT_INTEGER   ;
     PERIODIC      :  LOGICAL;
     WEIGHTS       :  LIST(2..125) OF DOUBLE_REAL    ;
     NO_WEIGHTS    :  SHORT_INTEGER   ;
     CTRL_PTS      :  LIST(2..125) OF POINTER TO
                      (SPLINE_POINT2   );
     END_ENTITY;
```

5.1.5 Classes

A class is defined by a list of entities and/or classes. The keyword CLASS is followed by the class name, class kind number enclosed in parentheses, and a semicolon. The class members are then listed. Each member name is followed by a semicolon, two dashes, and the member kind number and member type (entity or class). The class definition is concluded with the keyword END_CLASS and a semicolon. The class definition is illustrated in the example below.

```
--PAGE 7


(*---------------------------------------------------------------------*)
(*                        CLASS DEFINITION                             *)
(*---------------------------------------------------------------------*)

CLASS INSP_TECHNIQUE   (974);
     AIRFLOW          ;         --  938 ENTITY
     END_CLASS;
```

5.1.6 Subschemas

A subschema is defined by a list of entities and/or classes, as illustrated in the example below. The keyword SUBSCHEMA is followed by the subschema name and a semicolon. The subschema members are then listed. Each member name is followed by a semicolon, two dashes, and the member kind number and member type (entity or class). The subschema definition is concluded with the keyword END_SUBSCHEMA and a semicolon.

```
--PAGE 8


(*--------------------------------------------------------------*)
(*                      SUBSCHEMA DEFINITION                    *)
(*--------------------------------------------------------------*)

SUBSCHEMA GMAP           ;
        RB_SPLINE2       ;       -- 225 ENTITY
        AIRFLOW          ;       -- 938 ENTITY
        OPEN_CURVE2      ;       -- 222 CLASS
        INSP_TECHNIQUE   ;       -- 974 CLASS
        END_SUBSCHEMA;
```

5.1.7 Indices

All of the supertypes are listed in an index in alphabetical order. The name is followed by the report page number for the supertype definition. The Supertype Index is illustrated in the example below.

```
--PAGE 9


(*-----------------------------------------------------------*)
(*                     SUPERTYPE INDEX                       *)
(*-----------------------------------------------------------*)


COLLECTOR       ...........................................PAGE    3
GROUPING        ...........................................PAGE    4
```

All of the entities are listed in an index in alphabetical order. The name is followed by the kind number and the report page number for the entity definition. The Entity Index is illustrated in the following example.

```
--PAGE 10


(*-------------------------------------------------------------*)
(*                         ENTITY INDEX                        *)
(*-------------------------------------------------------------*)

AIRFLOW        (   938)...........................................PAGE    5
RB_SPLINE2     (   225)...........................................PAGE    6
```

All of the classes are listed in an index in alphabetical order. The name is followed by the kind number and the report page number for the class definition. The Class Index is illustrated below.

```
--PAGE 11


(*-------------------------------------------------------------*)
(*                         CLASS INDEX                         *)
(*-------------------------------------------------------------*)

INSP_TECHNIQUE  (   974)...........................................PAGE    7
OPEN_CURVE2     (   222)...........................................PAGE    8
```

All of the subschema definitions are listed in an index in alphabetical order. The name is followed by the report page number for the subschema definition. The Subschema Index is illustrated below.

```
--PAGE 12


(*-------------------------------------------------------------*)
(*                       SUBSCHEMA INDEX                       *)
(*-------------------------------------------------------------*)

GMAP             .................................................PAGE    8
```

5.2 PASCAL INCLUDE FILES

The PASCAL Include Files contain a compile-time binding representation of the entity definitions. Included in these files are the defined type declarations, the entity kind constants, the entity ADB and constituent definitions, the MAS entblock declaration, and the MAS keyblock constituent definition.

The first lines in the file indicate the subschema for which the PASCAL Include Files were generated, as shown in the example below.

```
(****************************************************************************)
(*   PASCAL INCLUDE FILES                                                 *)
(*     FOR THE APPLICATION SUBSCHEMA GMAP                                 *)
(****************************************************************************)
```

The contents of the file are outlined in the following example.

```
(****************************************************************************)
(*   THE PASCAL INCLUDE FILES CONTAIN                                     *)
(*     THE BASIC TYPES AND THEIR IMPLEMENTATION                           *)
(*     THE DEFINED TYPE DECLARATIONS                                      *)
(*     THE ENTITY KIND CONSTANTS                                          *)
(*     THE ENTITY ADB AND CONSTITUENT DEFINITIONS                        *)
(*     THE MAS ENTBLOCK DECLARATION                                       *)
(*     THE KEYBLOCK CONSTITUENT DEFINITION                                *)
(****************************************************************************)
```

The basic types and their IBM PASCAL implementation are described below.

```
(*****************************************************************)
(*    BASIC TYPES:                    IMPLEMENTED IN IBM PASCAL AS:  *)
(*                                                                   *)
(*    INTEGER (1 BYTE)                PACKED 0..255                  *)
(*    INTEGER (2 BYTES)               PACKED 0..65535               *)
(*    INTEGER (4 BYTES)               INTEGER                       *)
(*    REAL    (4 BYTES)               SHORTREAL                     *)
(*    REAL    (8 BYTES)               REAL                         *)
(*    STRING  (N BYTES)               PACKED ARRAY(.1..N.) OF CHAR *)
(*    BOOLEAN                         LOGICAL                      *)
(*    ENUMERATION                     ENUMERATED SCALAR            *)
(*****************************************************************)
```

The defined types are declared as described in the example below. T_CL_POSITION is automatically generated.

```
(*****************************************************************)
(*  DEFINED TYPE DECLARATIONS                                   *)
(*****************************************************************)

TYPE

    T_CL_POSITION      = INTEGER;

    T_DOUBLE_REAL      = REAL;

    T_FULL_PARTIAL     = ( FULL              ,
                           PARTIAL           );

    T_NSE_CODES        = ( REQUIRED          ,
                           OPTIONL           ,
                           UOS               ,
                           GLOBL             ,
                           EXCEPTION         ,
                           ALTERNATIVE       ,
                           INFORMATION       );

    T_RB_SPLINE_CODE   = ( UNDETERMINED      ,
                           LINEAR_ARC        ,
                           CIRCULAR_ARC      ,
                           ELLIPTICAL_ARC    ,
                           PARABOLIC_ARC     ,
                           HYPERBOLIC_ARC    );

    T_SHORT_INTEGER    = PACKED 0..65535;
```

The kind constants are established for each entity, as shown in the example below.

```
(*******************************************************************)
(*   ENTITY KIND CONSTANTS                                        *)
(*******************************************************************)


CONST

      K_ARRAY_ENTITY      =       1100 ;
      K_AIRFLOW           =        938 ;
      K_RB_SPLINE2        =        225 ;
```

The array entity is automatically generated. See the AIRFLOW entity for a description of the entity declarations.

```
(*********************************************************************)
(*   ENTITY DECLARATIONS                                            *)
(*********************************************************************)

(*###############################*)
(*   ARRAY_ENTITY                *)
(*###############################*)

TYPE

      P_ARRAY_ENTITY      = RECORD
        CL_ENTITIES       : ENTKEY;
      END;

      S_ARRAY_ENTITY      = RECORD
        CL_ENTITIES       : T_CL_POSITION;
      END;

CONST

      C_ARRAY_ENTITY      = S_ARRAY_ENTITY (1);
```

5.3 ENTITY ADB AND CONSTITUENT DEFINITIONS

Two type declarations and a structured constant are created for each entity, and shown in the example following these paragraphs. The first type declaration is the variant portion of the MAS ENTBLOCK which is unique to the entity. It is a record which contains the local and inherited attributes which are located in the ADB (nonpointer data types). The name of the record is formed from the name of the entity with a prefix 'E__'.

The second type declaration is the variant for the KEYBLOCK which is unique to the entity. It is a record which contains the local and inherited attributes which are located in the constituent list (pointer data type). The name of the record is formed from the name of the entity with a prefix 'P '.

```
(*################################*)
(*  AIRFLOW                       *)
(*################################*)

TYPE

        E_AIRFLOW           = RECORD
          MAX_PRES_RATIO    : T_DOUBLE_REAL      ;
          MIN_PRES_RATIO    : T_DOUBLE_REAL      ;
          AIRFLOW_PARM      : T_DOUBLE_REAL      ;
          APPLICATION       : T_NSE_CODES        ;
          AIRFLOW_KIND      : T_FULL_PARTIAL     ;
        END;

        P_AIRFLOW           = RECORD
          PROP_REFS         : ENTKEY;
          INSP_CONTROLS     : ENTKEY;
          BLOCKS            : ENTKEY;
        END;
```

A structured constant for constituent list positions (IBM specific) is defined. All of the entity's local and inherited constituent attributes (pointer types) are included. An example is presented below.

R20454/6

```
        S_AIRFLOW           ▪ RECORD
          PROP_REFS         : T_CL_POSITION;
          INSP_CONTROLS     : T_CL_POSITION;
          BLOCKS            : T_CL_POSITION;
        END;

  CONST

        C_AIRFLOW           ▪ S_AIRFLOW        (   1 ,
                                                   2 ;
                                                   3 );
```

The constant statement (IBM PASCAL peculiar) assigns values to the record fields defined above. (PROP_REFS is assigned the value of 1, INSP_CONTROLS is assigned the value of 2, and BLOCKS is assigned the value of 3.)

A user can reference a constituent using the MAS routine MALGTK in the following way:

MALGTK(AIRFLOW_KEY,   C_AIRFLOW.PROP_REFS,
PROP_REFS_KEY, XRC)

C_AIRFLOW.PROP_REFS represents the position of the PROP_REFS constituent in the constituent list for the AIRFLOW entity.

Below is another example of an entity declaration.

```
(*################################*)
(*  RB_SPLINE2                    *)
(*################################*)

TYPE

        E_RB_SPLINE2       = RECORD
          KNOTS            : ARRAY(.   1 ..   125 .) OF
                               T_DOUBLE_REAL      ;
          WEIGHTS          : ARRAY(.   1 ..   125 .) OF
                               T_DOUBLE_REAL      ;
          DEGREE           : T_SHORT_INTEGER   ;
          NO_KNOTS         : T_SHORT_INTEGER   ;
          NO_WEIGHTS       : T_SHORT_INTEGER   ;
          CRV_IND          : T_RB_SPLINE_CODE  ;
          PERIODIC         : BOOLEAN;
        END;

        P_RB_SPLINE2       = RECORD
          CTRL_PTS         : ENTKEY;
        END;

        S_RB_SPLINE2       = RECORD
          CTRL_PTS         : T_CL_POSITION;
        END;

CONST

        C_RB_SPLINE2       = S_RB_SPLINE2       (   1 );
```

The ENTBLOCK declaration contains two portions:

1. The fixed portion, which includes the automatically generated KIND, LENGTH, and SYSUSE fields and the user defined global fields (in the example below, IDENT).

2. The variant portion, which includes each entity record as previously declared (in the example below, the pseudo-entity ARRAY_ENTITY, and the entities AIRFLOW and RB_SPLINE2).

```
(********************************************************************)
(*  MAS ENTBLOCK DECLARATIONS                                      *)
(********************************************************************)

TYPE

    ENTBLOCK = RECORD
      KIND              : INTEGER;
      LENGTH            : INTEGER;
      SYSUSE            : INTEGER;
      IDENT             : INTEGER;
      CASE KIND : OF
        K_ARRAY_ENTITY     : ();
        K_AIRFLOW          : (AIRFLOW      : E_AIRFLOW      );
        K_RB_SPLINE2       : (RB_SPLINE2   : E_RB_SPLINE2   );
      END;
```

With the following variable definition:

VAR

        ADB:ENTBLOCK;

A user may *reference an attribute in the ADB using the MAS routine MAEGTK in the* following way:

MAEGTK(AIRFLOW_KEY, ADB, IRC);

ADB.IDENT represents the attribute which is global to all entities.

ADB.AIRFLOW.MAX_PRES_RATIO represents the attribute which is unique to the AIRFLOW entity.

The KEYBLOCK declaration, shown below, contains a variant portion for each entity's local and inherited constituent attributes (pointer types) as previously declared.

```
(****************************************************************)
(*   KEYBLOCK DECLARATIONS                                     *)
(****************************************************************)

TYPE

    KEYBLOCK = RECORD
      CASE INTEGER OF
        K_ARRAY_ENTITY       : (ARRAY_ENTITY     : P_ARRAY_ENTITY    );
        K_AIRFLOW            : (AIRFLOW          : P_AIRFLOW         );
        K_RB_SPLINE2         : (RB_SPLINE2       : P_RB_SPLINE2      );
    END;
```

5.4 PHYSICAL SUBSCHEMA REPORT (PSR)

The PSR displays all of the entity definitions in a particular physicalized subschema. The PSR contains information similar to that contained in the Data Dictionary. The following qualities are unique to the PSR:

- Provides indices organized by entity name and kind number
- Is presented in a more readable form than the Data Dictionary
- Excludes class information
- Excludes any minimum occurrence information.

The format of the PSR contains the elements listed below. These elements are represented with the appropriate reference number in the following examples.

1. Entity name and kind number.

2. ADB information:

    (a) Attribute name

    (b) Attribute type

    (c) Attribute conceptual schema order (position)

    (d) Attribute physical schema order (position)

    (e) Size of ADB attribute

    (f) ADB displacement

    (g) Total number and names of enumeration values (for enumeration attributes)

    (h) Total number of dimensions, the array low bounds, and the array high bounds (for array attributes)

3. Constituent list information

  (i) Attribute name

  (j) Attribute type

  (k) Attribute conceptual schema order

  (l) Constituent list position

  (m) Total number of constituent reference entity kinds

  (n) List of possible kind values

  (o) Total number of dimension, the array low bounds, and the array high bound (for arrays of pointers)

```
** PAGE   1**************************************************************

** ENTITY-NAME: RB_SPLINE2      (1)
          KIND: 225

** APPLICATION DATA BLOCK      (2)
.............................................................
ATTRIBUTE·              CS·      PS·                ADB·
NAME (a)   TYPE (b)   ORDER (c) ORDER (d) SIZE (e)  DISP (f)
.............................................................
KIND       INTEGER       1       1        4         0
LENGTH     INTEGER       2       2        4         4
SYSUSE     INTEGER       3       3        4         8
IDENT      INTEGER       4       4        4         12
CRV_IND    ENUMERATION   5       10       1         2022
                                  (g)         #-OF-VAL:  6,   VALUES : UNDETERMINED
                                                               LINEAR_ARC
                                                               CIRCULAR_ARC
                                                               ELLIPTICAL_ARC
                                                               PARABOLIC_ARC
                                                               HYPERBOLIC_ARC
DEGREE     INTEGER       6       7        2         2016
KNOTS      ARRAY OF
           REAL          7       5        8         16
                                  (h)         #-OF-DIM:  1,   LOW-BND:  1,   UPR-BND:125
NO_KNOTS   INTEGER       8                2         2018
PERIODIC   LOGICAL       9       11       1         2023
WEIGHTS    ARRAY OF
           REAL          10      6        8         1016
                                              #-OF-DIM:  1,   LOW-BND:  1,   UPR-BND:125
NO_WEIGHTS INTEGER       11      9        2         2020


** CONSTITUENT LIST
.............................................................
ATTRIBUTE·             CS·                     ELIG
NAME (i)   TYPE (j) ORDER (k) CL_POS (l) ELIG # (m) KIND (n)
.............................................................
CTRL_PTS   ARRAY OF
           POINTER     12       1          1      245
                                  (o)    #-OF-DIM:  1, LOW-BND:  1, UPR-BND:125 1
```

R30066/8

```
** PAGE   2 ** ***************************************************

** ENTITY-NAME: AIRFLOW
     KIND: 938


** APPLICATION DATA BLOCK
..................................................
ATTRIBUTE-          CS-    PS-          ADB-
NAME         TYPE   ORDER  ORDER  SIZE  DISP
..................................................
KIND         INTEGER    1      1     4     0
LENGTH       INTEGER    2      2     4     4
SYSUSE       INTEGER    3      3     4     8
IDENT        INTEGER    4      4     4    12
APPLICATION  ENUMERATION 5     8     1    40
                               #-OF-VAL:  7,   VALUES :REQUIRED
                                                       OPTIONL
                                                       UOS
                                                       GLOBL
                                                       EXCEPTION
                                                       ALTERNATIVE
                                                       INFORMATION

AIRFLOW_KIND  ENUMERATION  8     9     1    41
                               #-OF-VAL:  2,   VALUES : FULL
                                                        PARTIAL

MAX_PRES_RATIO  REAL     9      5     8    16
MIN_PRES_RATIO  REAL    10      6     8    24
AIRFLOW_PARM    REAL    11      7     8    32

** CONSTITUENT LIST
..................................................
ATTRIBUTE-          CS-                   ELIG
NAME         TYPE   ORDER  CL_POS ELIG #  KIND
..................................................
PROP_REFS    ARRAY OF
             POINTER     6      1      1    199
                               #-OF-DIM:  1,   LOW-BND:  1,   UPR-BND:255
INSP_CONTROLS ARRAY OF
             POINTER     7      2      1    931
                               #-OF-DIM:  1,   LOW-BND:  1,   UPR-BND: 10
BLOCKS       ARRAY OF
             POINTER    12      3      1    939
                               #-OF-DIM:  1,   LOW-BND:  1,   UPR-PND:255
```

The following example illustrates two indices provided by the PSR, one in order of entity kind and the other in alphabetical order. All of the entities are listed in order of entity kind and in alphabetical order. The name and kind number are followed by the page number for the entity definition. The PSR shows all of the entity physical definitions for a specified subschema.

```
┌─────────────────────────────────┐   ┌─────────────────────────────────┐
│      ** TABLE OF CONTENTS **     │   │      ** TABLE OF CONTENTS **     │
│      ( IN ORDER OF ENTITY KIND ) │   │      ( IN ORDER OF ENTITY NAME ) │
│                                  │   │                                  │
│  KIND    ENTITY-NAME      PAGE   │   │  ENTITY NAME     KIND      PAGE   │
│  ------------------------------- │   │  ------------------------------- │
│                                  │   │                                  │
│  ( 225) RB_SPLINE2    ------  1  │   │  AIRFLOW       ( 938) -----  2   │
│                                  │   │                                  │
│  ( 938) AIRFLOW       ------  2  │   │  RB_SPLINE2    ( 225) -----  1   │
│                                  │   │                                  │
└─────────────────────────────────┘   └─────────────────────────────────┘
```

5.5 RUNTIME SUBSCHEMA FILES

The Runtime Subschema is a binary form of the entity definitions for use by the Name/Value Interface. It may be thought of as a Data Dictionary which has been optimized for runtime performance.

Figure 5-1 describes the contents of the Runtime Subschema definition file.

5.6 PDDI/GMAP DATA DICTIONARY

The Data Dictionary is composed of two files:

- Index file
- Definition file.

Index file example:

```
1100,ARRAY_ENTITY          ,    2, 11,E
 225,RB_SPLINE2            ,  213, 19,E
 938,AIRFLOW              , 2560, 22,E
 222,OPEN_CURVE2           , 2867,  6,C
 974,INSP_TECHNIQUE        , 3288,  9,C
```

The format of the records in the Data Dictionary index file is:

Col. 1- 5 : Kind number (5 digits)
Col. 7-22 : Entity or class name (16 characters)
Col. 40-45 : Position of first definition record (6 digits)
Col. 47-49 : Number of records in definition (3 digits)
Col. 51   : /E' for entity or /C' for class

R3036A/8
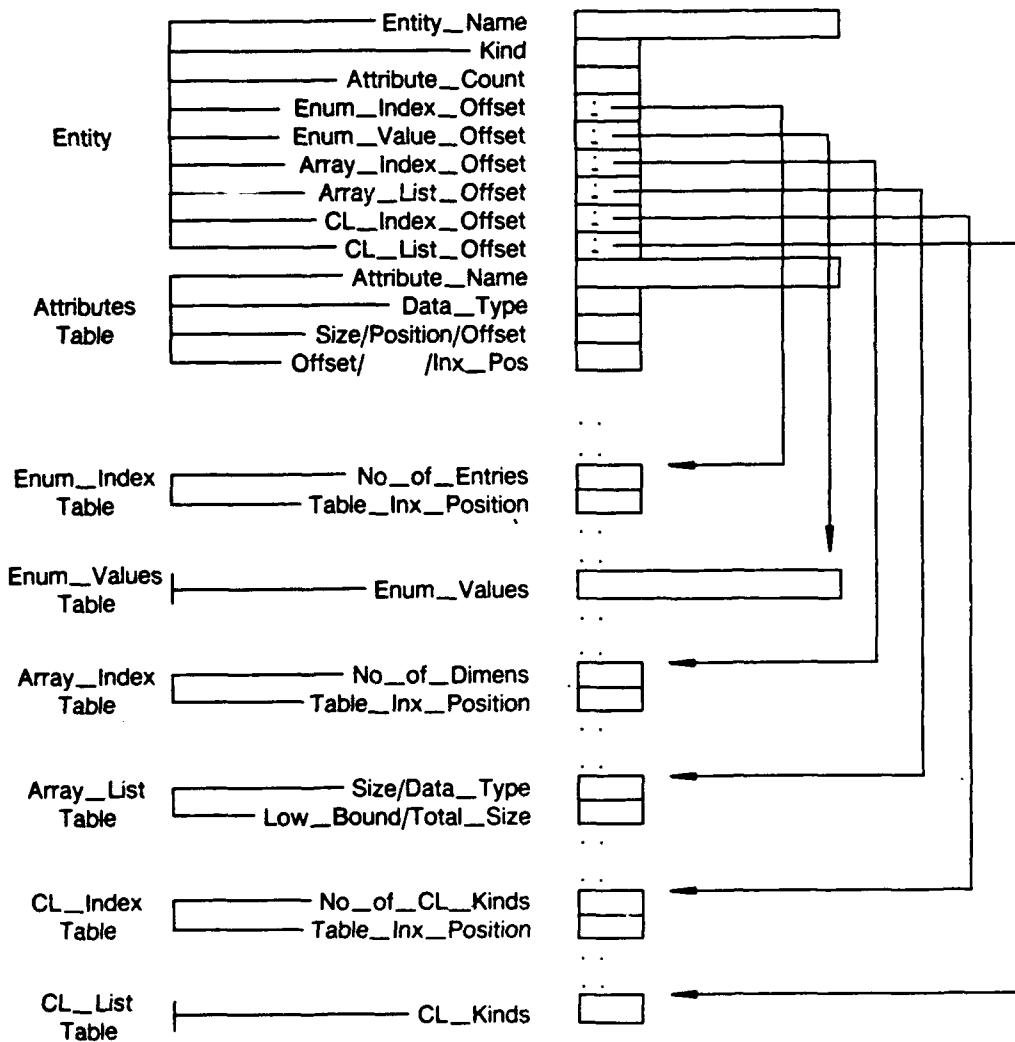
FDA 357847

Figure 5-1. Runtime Subschema Data Structure

5-19

R3085A/8

The Data Dictionary index file is in order by kind number. A definition file example follows.

```
########1########2########3########4########5########6########7########8
  ARRAY_ENTITY    , 1100, 5
  KIND         , 1, 1,  1, 0,1,  4,      0
  LENGTH       , 2, 2,  1, 0,1,  4,      4
  SYSUSE       , 3, 3,  1, 0,1,  4,      8
  IDENT        , 4, 4,  1, 0,1,  4,     12
  CL_ENTITIES     , 5, 0,  1, 1,7,  4,      1
X47,  101,  102,  106,  107,  108,  109,  118,  199,  210,  212,  245,  310
X47,  312,  321,  322,  330,  333,  343,  344,  345,  350,  360,  400,  401
X47,  406,  416,  420,  440,  450,  470,  510,  517,  529,  569,  630,  640
X47,  700,  701,  710,  717,  722,  744,  759,  900,  922,  930,  952
A  1,254
########1########2########3########4########5########6########7########8
  RB_SPLINE2     , 225,12
  KIND         , 1, 1,  1, 0,1,  4,      0
  LENGTH       , 2, 2,  1, 0,1,  4,      4
  SYSUSE       , 3, 3,  1, 0,1,  4,      8
  IDENT        , 4, 4,  1, 0,1,  4,     12
  CRV_IND      , 5,10,  1, 0,5,  1,   2022
X 6,UNDETERMINED    ,LINEAR_ARC       ,CIRCULAR_ARC     ,ELLIPTICAL_ARC
X 6,PARABOLIC_ARC   ,HYPERBOLIC_ARC
  DEGREE       , 6, 7,  1, 0,1,  2,   2016
  KNOTS        , 7, 5,  2, 1,2,  8,     16
A  1,125
  NO_KNOTS     , 8, 8,  1, 0,1,  2,   2018
  PERIODIC     , 9,11,  1, 0,4,  1,   2023
  WEIGHTS      ,10, 6,  2, 1,2,  8,   1016
A  1,125

  NO_WEIGHTS       ,11, 9,  1, 0,1,  2,   2020
  CTRL_PTS      ,12, 0,  2, 1,7,  4,      1
X 1,  245
A  1,125
```

```
########1########2########3########4########5########6########7########8
  AIRFLOW        ,  938,12
  KIND           ,  1, 1,   1, 0,1,   4,      0
  LENGTH         ,  2, 2,   1, 0,1,   4,      4
  SYSUSE         ,  3, 3,   1, 0,1,   4,      8
  IDENT          ,  4, 4,   1, 0,1,   4,     12
  APPLICATION    ,  5, 8,   1, 0,5,   1,     40
X 7,REQUIRED         ,OPTIONL          ,UOS                   ,GLOBL
X 7,EXCEPTION        ,ALTERNATIVE      ,INFORMATION
  PROP_REFS      ,  6, 0,   0, 1,7,   4,      1
X 1,  199
A 1,255
  INSP_CONTROLS  ,  7, 0,   1, 1,7,   4,      2
X 1,  931
A 1, 10
  AIRFLOW_KIND   ,  8, 9,   1, 0,5,   1,     41
X 2,FULL          ,PARTIAL
  MAX_PRES_RATIO ,  9, 5,   1, 0,2,   8,     16
  MIN_PRES_RATIO , 10, 6,   1, 0,2,   8,     24
  AIRFLOW_PARM   , 11, 7,   1, 0,2,   8,     32
  BLOCKS         , 12, 0,   0, 1,7,   4,      3
X 1,  939
A 1,255


########1########2########3########4########5########6########7########8
  OPEN_CURVE2    ,  222, 5
    227          ,  1, 0,   0, 0,0,   0,      0
    228          ,  2, 0,   0, 0,0,   0,      0
    226          ,  3, 0,   0, 0,0,   0,      0
    229          ,  4, 0,   0, 0,0,   0,      0
    225          ,  5, 0,   0, 0,0,   0,      0
########1########2########3########4########5########6########7########8
  INSP_TECHNIQUE , 974, 8
    938          ,  1, 0,   0, 0,0,   0,      0
    933          ,  2, 0,   0, 0,0,   0,      0
    942          ,  3, 0,   0, 0,0,   0,      0
    936          ,  4, 0,   0, 0,0,   0,      0
    937          ,  5, 0,   0, 0,0,   0,      0
    934          ,  6, 0,   0, 0,0,   0,      0
    941          ,  7, 0,   0, 0,0,   0,      0
    935          ,  8, 0,   0, 0,0,   0,      0
```

There are six record formats in the data dictionary definition file:

1. Entity or Class name record (first record of definition):

    Col. 1   : / '
    Col. 2-17 : Entity or class name (16 characters)
    Col. 19-23 : Kind number (5 digits)
    Col. 25-26 : Number of attributes or class members (2 digits)
                 (Note: Not necessarily the same as the number of records in
                        the definition.)

Example:

```
#########1#########2#########3#########4#########5#########6#########7########8
 AIRFLOW     ,  938,12
```

2. Attribute definition record (after first record of entity definition, first column is blank):

    Col. 1   :   / '
    Col. 2-17 :   Attribute name (16 characters)
    Col. 19-20 :   Conceptual schema order (2 digits)
    Col. 22-23 :   Physical schema order (2 digits)
    Col. 25-27 :   Minimum occurrences (3 digits)
                     (Note:0 for optional attributes, as specified for the list
                     and data types, 1 otherwise)
    Col. 29-30 :   Number of array dimensions (2 digits)
    Col. 32   :   Attribute data type (1 digit)
                     (Note: Integer = 1, Real = 2, String = 3, Logical = 4,
                     Enumeration = 5, and Constituent reference = 7)
    Col. 34-36 :   Size of ADB field (3 digits)
    Col. 38-43 :   ADB displacement or CL position (6 digits)

Example:

```
#########1#########2#########3#########4#########5#########6#########7########8
 APPLICATION      , 5, 8,  1, 0,5,  1,     40
```

R20356/8

3. Enumeration attribute continuation record (follows attribute definition record of enumeration data type (5), first column is /x'):

Col. 1   :   /X' (continuation flag)
Col. 2- 3 :   Number of values (2 digits)
Col. 6-20 :   Enumeration values (16 characters)
              Note: Enumeration values are repeated, as many as 4 per
              record: cols. 22-37, 39-54, 56-71. Succeeding records, if
              required, repeat this format)

Example:

```
########1########2########3########4########5########6########7########8
X 7,REQUIRED        ,OPTIONL        ,UOS            ,GLOBL
```

4. Constituent reference attribute continuation record (follows attribute defini-tion record of pointer data type (7), first column is /x'):

Col. 1 :   /X' (continuation flag)
Col. 2-3 :   Number of entity kinds (2 digits)
Col. 6-9 :   Entity Kind (5 digits)
             (Note: Entity kinds are repeated, as many as 12 per rec-
             ord: 11-15 ... 71-75. Succeeding records, if required, repeat
             this format)

Example:

```
########1########2########3########4########5########6########7########8
X 1,  199
```

5. Array bounds record (follows attribute definition record of array data type (10), first column is /A'):

Col. 1 :   /A' (array bounds flag)
Col. 2-4 :   Low bound of array dimension (3 digits)
Col. 6-8 :   High bound of array dimension (3 digits)
             Note: Pairs of low and high bound are repeated for mul-
             tidimensional arrays, as many as 10 dimensions per record:
             10-12, 14-16 ...74-76, 78-80. Succeeding records, if required,
             repeat this format.)

Example:

```
########1########2########3########4########5########6########7########8
A 1,255
```

6. Membership record (after first record of class definition, first column is blank):

Col. 1   :   / '
Col. 2-6 :   Kind number (5 characters)

Example:

```
#########1#########2#########3#########4#########5#########6#########7#########8
    227           , 1,  0,   0, 0,0,   0,      0
```

5.7 CROSS REFERENCE REPORT

The Cross Reference Report capabilities allow a user to interactively query a schema model. The query results are displayed on the screen and are also written to a dataset. The format of the entry in the dataset is the same as the format of the data displayed on the screen, as described in Section 4.5.5.

SECTION 6

BATCH INTERFACE

6.1 OVERVIEW AND KEYWORDS

The Batch interface allows a user to create a new schema or add to an existing one. It is particularly useful for the bulk load of entity definitions. The Batch interface accepts a file which is in EXPRESS format (i.e., the Conceptual Schema Report) and creates a schema model. To update or review a schema, the interactive Schema Manager must be used.

To add to an existing file, the keyword RETRIEVE must precede all declarations. RETRIEVE prompts the Batch interface to retrieve an existing file. Note: If the keyword RETRIEVE is not specified, then any existing model will be replaced.

Reports can also generated through the Batch interface. The keyword REPORT accesses this capability as illustrated in Table 6-1.

TABLE 6-1

ACCESSING REPORT FEATURES THROUGH BATCH INTERFACE

REPORT CSR	Generates a Conceptual Schema Report.
REPORT PIF GMAP	Generates a PASCAL Include File for the GMAP subschema.
REPORT RTS GMAP	Generates a Run-Time Subschema (Definition File and Index File) for the GMAP subschema.
REPORT DD GMAP	Generates a Data Dictionary (Definition File and Index File) for the GMAP subschema.
REPORT PSR GMAP	Generates a Physical Subschema Report for the GMAP subschema.

The schema is automatically filed when the end-of-file is encountered and no errors, warnings, or unresolved references exist.

Keywords available in the Batch Interface function of the Schema Manager include: array, class, entity, enumeration, global, integer, list, logical, of, option, pointer, real, schema/subschema, set, string, subtype, supertype, to, and type. Table 6-2 lists each keyword along with a description of each.

TABLE 6-2

BATCH INTERFACE KEYWORDS

KEYWORD	DESCRIPTION
ARRAY	A built-in type which defines the characteristics of an ordered, homogeneous collection which has a fixed size. **Example:** a : ARRAY(1:4) of real; b : ARRAY(1 to 3) of ARRAY(1 to 4) of real;
CLASS	Starts a class definition block. The block must be terminated by an END_CLASS statement. CLASS is an extension to EXPRESS. **Example:** CLASS ClassName (KindNumber); -- body of class end_class; The body of the class block may contain any combination of the following: entity blocks, entity identifiers, class blocks, or class identifiers.
END_CLASS END_ENTITY END_SCHEMA END_SUBSCHEMA END_SUPERTYPE	Blocks are terminated by explicit end keywords, i.e., an entity block is terminated by END_ENTITY.

ENTITY	Starts the declaration of an entity. Entity declarations follow the form:
	
	  ENTITY EntityName (KindNumber);
	    AttributeName1 : TypeDefinition;
	    AttributeName2 : TypeDefinition;
	  end_entity;
	
	Type Definition must be a defined type, or one of the built-in types such as INTEGER, STRING, ARRAY, etc.

ENUMERATION	A type which allows names to be given to an ordered collection of integer values. An enumeration may not be defined directly as an attribute of an entity. It must first be defined as a defined type. **Example:** type color = ENUMERATION OF (red, blue, green);
GLOBAL	Starts the declaration of a global attribute. GLOBAL is an extension to Express. **Example:** GLOBAL ident : integer;

TABLE 6-2

BATCH INTERFACE KEYWORDS (Continued)

KEYWORD	DESCRIPTION
INTEGER	A built-in type which defines the qualities of a whole decimal number. Precision is specified by the number of decimal digits (the default is 9 digits). **Example:** b : INTEGER(7);
LIST	A built-in type which defines the characteristics of an ordered, homogeneous collection which has a variable size. The size is specified by the minimum and maximum number of entries. **Example:** control : LIST(2 : 5) of real;
LOGICAL	A built-in type which describes the attribute to be a Boolean value (true or false). **Example:** c : logical;
OF	Used following the keywords ARRAY, ENUMERATION, LIST, and SET to improve readability.

OPTIONAL	Attribute modifier which stipulates that a value for an attribute may or may not be present. This does not mean that the attribute is optional, but that the value is optional.
	
	Example:
	
	  entity point (12345);
	    x,y : real;
	    z : OPTIONAL real;
	  end_entity;

POINTER	A reference to another entity, or to a class of entities, by either name or kind. POINTER is an EXPRESS external reference. **Example:** d : POINTER to (ClassName, EntityNumber, EntityName, ClassNumber);
REAL	A built-in type which describes the characteristics of a floating point decimal number. Precision is specified by the number of decimal digits in the mantissa (the default is 16 digits). **Example:** x : REAL(14);

R20444/9

TABLE 6-2

BATCH INTERFACE KEYWORDS (Continued)

KEYWORD	DESCRIPTION
REPORT	Generate one of the reports or files: CSR PSR PIF DD RTS
SCHEMA SUBSCHEMA	Starts a subschema definition block. The block must be terminated by an END_SCHEMA or END_SUBSCHEMA statement. SUBSCHEMA is an extension to EXPRESS. **Example:** SUBSCHEMA SubschemaName; -- body of subschema end_subschema; The body of the subschema block may contain any combination of the following: entity blocks, entity identifiers, class blocks, or class identifiers.
SET	A built-in type which defines the characteristics of an unordered, homogeneous collection which has a variable size. The size is specified by the minimum and maximum number of entries. **Example:** member : SET(1:5) of integer;
STRING	A built-in type which defines the characteristics of a string of characters. The string is fixed length for the number of characters specified (the default is one character). **Example:** name : STRING(24);
SUBTYPE	An optional part of an entity declaration which states that the entity is a subtype of the supertype listed. **Example:** supertype geometry; a : real; end_supertype; entity point SUBTYPE of (geometry); AttributeName : TypeDefinition; end_entity;

TABLE 6-2

BATCH INTERFACE KEYWORDS (Continued)

KEYWORD	DESCRIPTION
SUPERTYPE	Defines a group of attributes that can be inherited through a subtype reference. **Example:** SUPERTYPE geometry; a : real; end_supertype;
TO	Can be used in place of a colon in ARRAY, LIST, and SET declarations.
TYPE	An alias for a primitive, complex primitive, or aggregation data type. **Example:** TYPE string80 = string(80);

6.2 EXAMPLE BATCH INPUT FILE

An example batch input file is illustrated below:

```
SUBSCHEMA example; (* an example *)

   TYPE integer8 - INTEGER(8);

            real8  = REAL 8;

            string16 - STRING (16);

            log  = LOGICAL;

            array16 - ARRAY (1 to 16) OF POINTER(entity_one);

            enum = ENUMERATION OF (red, white, blue);

GLOBAL ident : INTEGER;

SUPERTYPE collector;
  supertype_field : INTEGER;
END_SUPERTYPE;

ENTITY entity_one (11111);
  field_one   : log;
  field_two   : POINTER(entity_two);
  field_three : enum;
END_ENTITY;

ENTITY entity_two (22222) SUBTYPE OF (collector);
  field_one   : array16;
  field_two   : integer8;
  field_three : real8;
  field_four : string16;
END_ENTITY;

CLASS class_one 12345;
  22222;
  entity_one;
END_CLASS;

END_SCHEMA;

REPORT CSR          -- generate the Conceptual Schema Report

REPORT PIF example-- generate the PASCAL Include File for the example
                     subschema
```

R2uu5a/9

SECTION 7

MODEL QUERY UTILITY (MQU)

MQU is a tool for the application developer to verify the working form model instances by using Entity Definitions from the Data Dictionary. The commands available from the Model Query Main Menu allow a user to retrieve a model, to verify a working form model in batch or by interaction, and to exit the Model Query. The MODEL QUERY MAIN MENU is illustrated in Figure 7-1.

Note: A user must retrieve (Command 3) a model interface before verifying a working form model in Batch or Interactive.

```
--------------------------MODEL QUERY MAIN MENU-----------------------

COMMAND ---> (a)


              1 - BATCH

              2 - INTERACTIVE

              3 - RETRIEVE

              4 - EXIT
```

Figure 7-1. Model Query Main Menu

Menu Description:

(a) Select option 1 to display the MODEL QUERY BATCH MENU.
Select option 2 to display the MODEL QUERY INTERACTIVE MENU.
Select option 3 to display the DATASET MENUs to restore a model from a
file to working form and the data dictionary.
Select option 4 to exit from the Model Query Utility.

7.1 RETRIEVE

The screen illustrated in Figure 7-2 is displayed when command 3, Retrieve, is selected from the MODEL QUERY MAIN MENU. Enter the name of the dataset containing the part model.

```
----------------------SELECT DATASET NAME FOR MODEL QUERY----------------

COMMAND ===>


        +----------------------------------------------------------+
        |                                                          |
        |         ENTER A DATASET NAME (WITHOUT QUOTES)            |
        |                                                          |
        |     ===>  (a)                                            |
        |                                                          |
        |                                                          |
        +----------------------------------------------------------+
```

Figure 7-2. Select Dataset Name for Model Query

Menu Description:

(a) Enter a dataset name for the model to be retrieved.
The DATA DICTIONARY DATA FILE menu will be displayed.

7-2

The **Data Dictionary Data File** screen, a submenu of the Retrieve Function, is illustrated in Figure 7-3. Enter the name of the Data Dictionary definition file.

```
-----------------------DATA DICTIONARY DATA FILE--------------------

COMMAND ===>


    +-------------------------------------------------------------+
    |                                                             |
    |          ENTER A DATASET NAME (WITHOUT QUOTES)              |
    |                                                             |
    |     ===> (a)                                                |
    |                                                             |
    +-------------------------------------------------------------+
```

Figure 7-3.   Data Dictionary Data File

Menu Description:

(a)   Enter a dataset name for the data dictionary data file.
The DATA DICTIONARY INDEX FILE menu will be displayed.

R20334/10

The **Data Dictionary Index File** screen, a submenu of the Retrieve Function, is illustrated in Figure 7-4. Enter the name of the Data Dictionary index file.

```
------------------------DATA DICTIONARY INDEX FILE---------------------

COMMAND ===>


     --------------------------------------------------------------
     |                                                            |
     |          ENTER A DATASET NAME (WITHOUT QUOTES)             |
     |                                                            |
     |     ===> (a)                                               |
     |                                                            |
     --------------------------------------------------------------
```

Figure 7-4. Data Dictionary Index File

Menu Description:

(a) Enter a dataset name for the data dictionary index file.
     The MODEL QUERY MAIN MENU will be displayed.

7.2 INTERACTIVE

The screen illustrated in Figure 7-5 is displayed when command 2, **Interactive,** is selected from the MODEL QUERY MAIN MENU. The menu contains a list of the entity kinds in the working form. A specific entity kind may be selected.

```
-----------------------MODEL QUERY INTERACTIVE MENU-----------------

COMMAND ---> (a)



        COMMANDS:
            1    : RETURN TO MAIN MENU
            2    : EXIT TO END THIS SESSION

        OPTION:
          SELECT AN ENTITY BY ENTERING ANY CHARACTER NEXT TO
          THE DESIRED MEMBER (AND BENEATH THE ASTERISK).

        ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).




                                                    NO. OF
    *  KIND NO.     ENTITY NAME                      INSTANCES
   (b)  (c)             (d)                          (e)
```

Figure 7-5. Model Query Interactive Menu

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to exit from the MQU.

(b) Select an entity kind by entering any character next to the entry on the list of entity kinds contained in the working form part model.
Either the INTERACTIVE ENTITY MENU will be displayed or the NO ENTITY FOUND menu will be displayed.

(c) The entity KIND number is displayed.

(d) The entity NAME is displayed.

(e) The total number of instances is displayed.

The **Interactive Entity Menu**, a submenu of the Interactive Function, is illustrated in Figure 7-6. This menu is displayed when an entity kind is selected. A specific instance of the entity kind may be selected.

```
------------------------INTERACTIVE ENTITY MENU----------------------

COMMAND ===> (a)


        COMMANDS:
              1     : RETURN TO PREVIOUS MENU
              2     : EXIT TO END THIS SESSION


    SELECTED  KIND  ===> (b)

              NAME  ===> (c)

    INSTANCE NUMBER ===> (d)
```

Figure 7-6.   Interactive Entity Menu

Menu Description:

(a)   Select option 1 to redisplay the previous menu.
      Select option 2 to exit from the MQU.

(b)   The selected entity kind number is displayed.

(c)   The selected entity name is displayed.

(d)   Enter an instance number ranging from 1 to the number of instances in the selected entity.
      The DISPLAY ATTRIBUTES menu will be displayed.

The **Display Attributes,** a submenu of the Interactive Function, is illustrated in Figure 7-7. The attributes of a specific entity instance are displayed.

```
------------------------DISPLAY ATTRIBUTES--------------------------

COMMAND ---> (a)



            COMMANDS:
                1    : RETURN TO PREVIOUS MENU
                2    : SELECT CONSTITUENTS
                3    : SELECT USERS
                4    : EXIT TO END THIS SESSION


    (b) a list of attributes showing the name of each attribute and its value
```

Figure 7-7.   Display Attributes

Menu Description:

(a)   Select option 1 to redisplay the previous menu.
      Select option 2 to display the SELECT CONSTITUENT MENU or the NO CONSTITUENTS menu.
      Select option 3 to display the SELECT USER MENU or the NO USERS menu.
      Select option 4 to exit from the MQU.

(b)   The attribute name and its value are displayed.

R30484/10

The **Select Constituent** screen, a submenu of the Interactive Function, is illustrated in Figure 7-8. The list of constituents for the specific entity instance is displayed. A specific constituent entity may be selected.

```
 ------------------------------SELECT CONSTITUENT-----------------------------

  COMMAND ===> (a)



          COMMANDS:
              1     : RETURN TO PREVIOUS MENU
              2     : EXIT TO END THIS SESSION

          OPTION:
            SELECT A CONSTITUENT BY ENTERING ANY CHARACTER NEXT
            TO THE DESIRED MEMBER (AND BENEATH THE ASTERISK).

          ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).



                                                                    OCC.
  ATTRIBUTE NAME                  * KIND  ENTITY NAME                NO.
      (b)                        (c)(d)    (e)                       (f)
```

Figure 7-8. Select Constituent

Menu Description:

(a) Select option 1 to redisplay the previous menu.
    Select option 2 to exit from the MQU.

(b) The attribute name is displayed

(c) Select a constituent entity by entering any character next to a member.

(d) The constituent kind number is displayed.

(e) The constituent entity name is displayed.

(f) The occurrence number of instance is displayed.

The **No Constituents** screen, a submenu of the Interactive Function, is illustrated in Figure 7-9. This menu is displayed when command 2, SELECT CONSTITUENTS, is selected from the DISPLAY ATTRIBUTES submenu (Figure 7-7) and there is no constituents for that entity.

```
-----------------------------NO CONSTITUENTS-----------------------------

COMMAND ===> (a)



              COMMANDS:
                     1     : RETURN TO PREVIOUS MENU
                     2     : EXIT TO END THIS SESSION



                       NO CONSTITUENTS
```

Figure 7-9. No Constituents

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to exit from the MQU.

R20254/10

CI UM560240011U
February 1989

The **Select User** screen, a submenu of the Interactive Function, is illustrated in Figure 7-10. The list of users of the specific entity instance are displayed. A specific user entity may be selected.
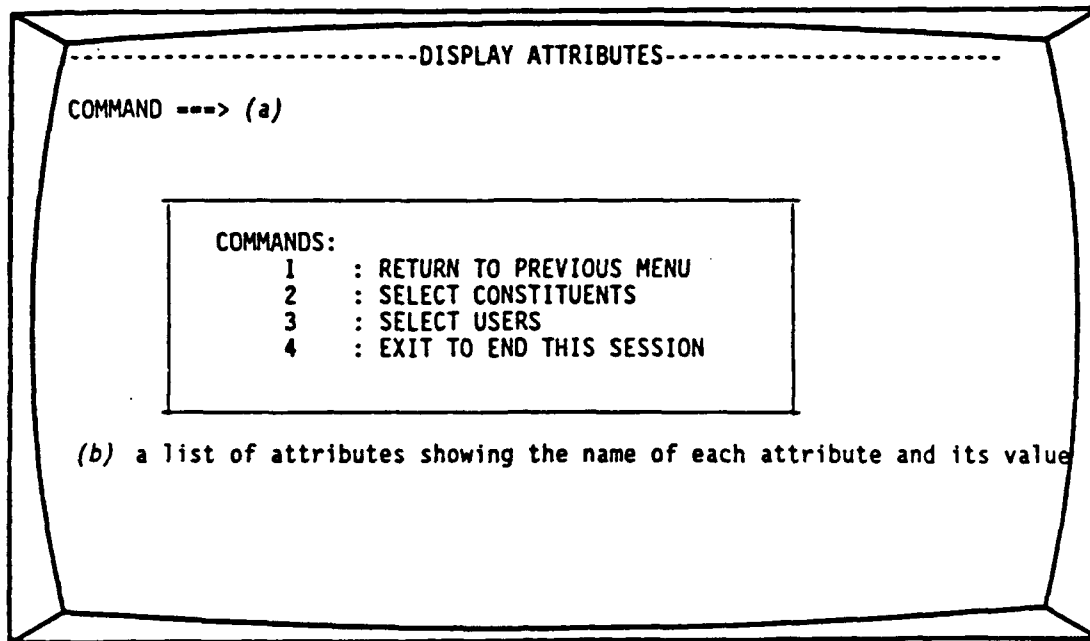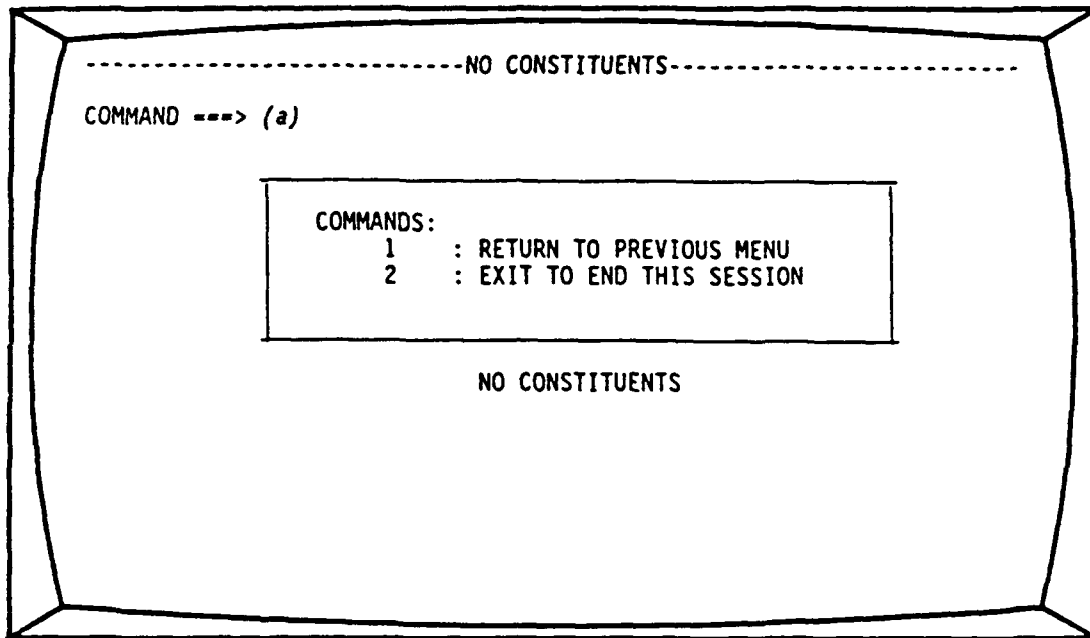
```
 ----------------------------SELECT USER----------------------------

COMMAND ===> (a)


    ┌──────────────────────────────────────────────────────────┐
    │  COMMANDS:                                                │
    │       1     : RETURN TO PREVIOUS MENU                     │
    │       2     : EXIT TO END THIS SESSION                    │
    │                                                           │
    │  OPTION:                                                  │
    │    SELECT AN USER BY ENTERING ANY CHARACTER NEXT TO THE   │
    │    DESIRED MEMBER (AND BENEATH THE ASTERISK).             │
    │                                                           │
    │  ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).     │
    │                                                           │
    └──────────────────────────────────────────────────────────┘


    *    KIND NO.     ENTITY NAME                    OCC. NO.
   (b)   (c)          (d)                            (e)
```

Figure 7-10. Select User

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to exit from the MQU.

(b) Select an user by entering any character next to a member.

(c) The user kind number is displayed.

(d) The user entity name is displayed.

(e) The occurrence number of instance is displayed.

7-10

The **No Users** screen, a submenu of the Interactive Function, is illustrated in Figure 7-11. This menu is displayed when command 3, SELECT USERS, is selected from the DISPLAY ATTRIBUTES submenu (Figure 7-7) and there is no user for that entity.

```
-------------------------------NO USERS----------------------------------

COMMAND ===> (a)



            COMMANDS:
                 1     : RETURN TO PREVIOUS MENU
                 2     : EXIT TO END THIS SESSION



                          NO USERS
```

Figure 7-11. No Users

Menu Description:

(a) Select option 1 to redisplay the previous menu.
      Select option 2 to exit from the MQU.

R30544/10

The **No Entity Found** screen, a submenu of the Interactive Function, is illustrated in Figure 7-12. This menu is displayed if there is an entity kind in the working form part model for which there is no definition in the Data Dictionary.

```
-------------------------------NO ENTITY FOUND------------------------------

COMMAND ===> (a)



                COMMANDS:
                    1     : RETURN TO PREVIOUS MENU
                    2     : EXIT TO END THIS SESSION


        DID NOT FIND ENTITY DEFINITIONS IN DATA DICTIONARY
```

Figure 7-12. No Entity Found

Menu Description:

(a) Select option 1 to redisplay the previous menu.
     Select option 2 to exit from the MQU.

7.3 BATCH

The screen illustrated in Figure 7-13 is displayed when command 1, **Batch**, is selected from the MODEL QUERY MAIN MENU. Enter the name of the dataset to contain the output from the batch query.

```
-------------------DATASET NAME FOR BATCH OUTPUT-------------------

COMMAND ===>


    ┌─────────────────────────────────────────────────────────────┐
    │              ENTER A DATASET NAME (WITHOUT QUOTES)            │
    │                                                               │
    │      ---> (a)                                                 │
    │                                                               │
    └─────────────────────────────────────────────────────────────┘
```

Figure 7-13. Dataset Name for Batch Output

Menu Description:

(a) Enter a dataset name for the batch output.
     The MODEL QUERY BATCH MENU will be displayed.

The **Model Query Batch Menu,** a submenu of the Batch Function, is illustrated in Figure 7-14. The menu contains a list of the entity kinds in the working form. All entity kinds may be reported, or a specific entity kind selected.

```
----------------------MODEL QUERY BATCH MENU----------------------
COMMAND ===> (a)

    ┌──────────────────────────────────────────────────────────┐
    │ COMMANDS:                                                 │
    │     1     : RETURN TO MAIN MENU                           │
    │     2     : PRINT ALL ENTITIES IN THE MODEL              │
    │     3     : EXIT TO END THIS SESSION                      │
    │                                                           │
    │ OPTION:                                                   │
    │   SELECT AN ENTITY TO PRINT BY ENTERING ANY CHARACTER NEXT│
    │   TO THE DESIRED MEMBER (AND BENEATH THE ASTERISK).       │
    │                                                           │
    │ ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).      │
    │                                                           │
    └──────────────────────────────────────────────────────────┘

                                                    NO. OF
    *    KIND NO.    ENTITY NAME                    INSTANCES
  (b)    (c)        (d)                             (e)
```

Figure 7-14. Model Query Batch Menu

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to print all entities in the model.
Select option 3 to exit from the MQU.

(b) Select entity kind by entering any character next to the entry on the list of entity kinds contained in the working form part model.
Either the BATCH ENTITY MENU will be displayed or the NO ENTITY FOUND menu will be displayed.

(c) The entity KIND number is displayed.

(d) The entity NAME is displayed.

(e) The total number of instances is displayed.

7-14

The **Batch Entity Menu,** a submenu of the Batch Function, is illustrated in Figure 7-15.
When a specific entity kind is selected, all instances may be reported, or a specific instance may
be reported.

```
------------------------------BATCH ENTITY MENU------------------------------

COMMAND ===> (a)


    ┌──────────────────────────────────────────────────────────────┐
    │  COMMANDS:                                                     │
    │      1    : RETURN TO PREVIOUS MENU                            │
    │      2    : EXIT TO END THIS SESSION                           │
    │                                                                │
    │  OPTION:                                                       │
    │    PRESS RETURN TO PRINT ALL INSTANCES OF THIS ENTITY          │
    │    OR SELECT AN INDIVIDUAL INSTANCE TO PRINT.                  │
    │                                                                │
    │  ENTER EITHER A COMMAND (ABOVE) OR AN OPTION (BELOW).          │
    │                                                                │
    └──────────────────────────────────────────────────────────────┘

    SELECTED  KIND   ===> (b)
               NAME  ===> (c)
    INSTANCE NUMBER ===> (d)
```

Figure 7-15. Batch Entity Menu

Menu Description:

(a) Select option 1 to redisplay the previous menu.
Select option 2 to exit from the MQU.

(b) The selected entity kind number is displayed.

(c) The selected entity name is displayed.

(d) Press ENTER to print all instances of defined entity or select an individual
instance of defined entity to print.

The **No Entity Found** screen, a submenu of the Batch Function, is illustrated in Figure 7-16. This menu is displayed if there is an entity kind in the working form part model for which there is no definition in the Data Dictionary (since the report is based on a Data Dictionary definition, no report can be produced).

```
--------------------------NO ENTITY FOUND--------------------------

COMMAND ===> (a)



        ┌─────────────────────────────────────────────────┐
        │ COMMANDS:                                        │
        │      1     : RETURN TO PREVIOUS MENU             │
        │      2     : EXIT TO END THIS SESSION            │
        │                                                  │
        └─────────────────────────────────────────────────┘

        DID NOT FIND ENTITY DEFINITIONS IN DATA DICTIONARY
```

Figure 7-16.   No Entity Found

Menu Description:

(a)   Select option 1 to redisplay the previous menu.
      Select option 2 to exit from the MQU.

APPENDIX A

GETDD-GMAP/PDDI DATA DICTIONARY INTERFACE

The GMAP/PDDI Data Dictionary contains entity and class definitions. It is composed of two files:

- Index File
- Definition File.

The GETDD interface routine uses the index file to locate a specific entity or class within the definition file, and then returns a copy of the definition. The formats for the definition records are explained in Section 5.6.

Format

GETDD (Kind, Avail, Order, User Array, Actual, Return Code)

where:

Parameter	Data Type	Description
Kind	Integer	The KIND number of the definition to be retrieved.
	Avail	Integer
Order	Char	The order for the attributes: /C' = conceptual schema order /P' = physical schema order other = definition file order
User_Array	Array of 80 character records	The calling program array to the definition records.
Actual	Integer	The actual number of records in the definition.
Return_Code	Integer	The possible return code values are: 0 = Success: the definition was copied into the calling program array. 1 = Failure: the specified KIND is not contained in the data dictionary; the calling program array was not changed. −1 = Warning: the actual number of records in the definition is greater than the number of records available in the calling program; the array contains a partial definition.

A-1

APPENDIX B

METAMODEL WORKING FORM
(MAS Working Form of the Schema Model)

The following information about the metamodel working form is for the use of system programmers who maintain or enhance the Schema Manager Software. It is not required for users of the Schema Manager Software.

The main purpose of the Schema Manager Software is to generate a physical implementation of entity definitions suitable for the working form of the MAS. The metamodel is the framework within which entities are defined. The elements of the metamodel are listed below:

- Entity
- Attribute
- Supertype
- Global attribute
- Integer data type
- Real data type
- String data type
- Logical data type
- Pointer data type

- Defined data type
- Array data type
- List data type
- Set data type
- Structure data type
- Enumeration data type
- Enumeration item
- Class
- Subschema
- Schema

The Schema Manager creates a *working form* model of the entity definitions, that is a metamodel. The metamodel working form is defined on the following pages.

R20S5A/12

```
                                                          Page 1 of 5
*)
ENTITY ENTITY_MODEL ( 1018);
    NAME             : STRING (16);
    KIND             : INTEGER;
    COMMENT          : STRING (150);
    PHYSICAL         : LOGICAL;
    CONSTITUENT      : ARRAY ( 1:# ) OF
                       POINTER TO ( SUPERTYPE_MODEL, -- maximum one supertype
                                    ATTRIBUTE_MODEL );
    END_ENTITY;

ENTITY SUPERTYPE_MODEL ( 1016);
    NAME             : STRING (16);
    CONSTITUENT      : ARRAY ( 1:# ) OF
                       POINTER TO ( SUPERTYPE_MODEL, -- maximum one supertype
                                    ATTRIBUTE_MODEL );
    END_ENTITY;

ENTITY ATTRIBUTE_MODEL ( 1019);
    NAME             : STRING (16);
    REQUIREDNESS     : T_REQD;
    COMMENT          : STRING (50);
    POSITION_OVERIDE : INTEGER;
    ADB_OFFSET_OR_CL_POSITION : INTEGER;
    ADB_SIZE_OR_ENUMERATION_INDEX : INTEGER;
    CONSTITUENT      : POINTER TO ( INTEGER_MODEL,
                                    REAL_MODEL,
                                    STRING_MODEL,
                                    LOGICAL_MODEL,
                                    POINTER_MODEL,
                                    ARRAY_MODEL,
                                    LIST_MODEL,        -- not implemented
                                    SET_MODEL,         -- not implemented
                                    STRUCTURE_MODEL,   -- not implemented
                                    DEFINED_TYPE_MODEL );
    END_ENTITY;

ENTITY GLOBAL_ATTRIBUTE_MODEL ( 1022);
    CONSTITUENT      : POINTER TO ( ATTRIBUTE_MODEL );
    END_ENTITY;

(*
```

```
*)

ENTITY INTEGER_MODEL ( 1001);
    PRECISION        : INTEGER;
    END_ENTITY;

ENTITY REAL_MODEL ( 1002);
    PRECISION        : INTEGER;
    END_ENTITY;

ENTITY STRING_MODEL ( 1003);
    LENGTH           : INTEGER;
    END_ENTITY;

ENTITY LOGICAL_MODEL ( 1004);
    DUMMY_ATTRIBUTE  : LOGICAL;
    END_ENTITY;

ENTITY POINTER_MODEL ( 1008);
    CONSTITUENT      : ARRAY ( 1:# ) OF
                       POINTER TO ( ENTITY_MODEL,
                                    CLASS_MODEL );
    END_ENTITY;

ENTITY DEFINED_TYPE_MODEL ( 1020);
    NAME             : STRING (16);
    REFERENCE        : POINTER TO ( INTEGER_MODEL,
                                    REAL_MODEL,
                                    STRING_MODEL,
                                    LOGICAL_MODEL,
                                    POINTER_MODEL,
                                    ENUMERATION_MODEL,
                                    ARRAY_MODEL,
                                    LIST_MODEL,        -- not implemented
                                    SET_MODEL,         -- not implemented
                                    STRUCTURE_MODEL,   -- not implemented
                                    DEFINED_TYPE_MODEL );
    END_ENTITY;

ENTITY ENUMERATION_MODEL ( 1005);
    CONSTITUENT      : ARRAY ( 1:# ) OF
                       POINTER TO ( ENUMERATION_ITEM_MODEL );
    END_ENTITY;

ENTITY ENUMERATION_ITEM_MODEL ( 1006);
    NAME             : STRING (16);
    END_ENTITY;
(*
```

R20444/12

```
*)

ENTITY ARRAY_MODEL ( 1010);
   LO_BOUND          : INTEGER;
   HI_BOUND          : INTEGER;
   MIN_OCCUR         : INTEGER;        -- used for simulation of list and set
   ARRAY_TYPE        : T_ARRAY_TYPE;
   CONSTITUENT       : POINTER TO ( INTEGER_MODEL,
                                    REAL_MODEL,
                                    STRING_MODEL,
                                    LOGICAL_MODEL,
                                    POINTER_MODEL,
                                    ARRAY_MODEL,
                                    DEFINED_TYPE_MODEL ); -- except structure,
                                                          -- list, or set
   END_ENTITY;

ENTITY LIST_MODEL ( 1999);                              -- not implemented
   MINIMUM           : INTEGER;
   MAXIMUM           : INTEGER;
   CONSTITUENT       : POINTER TO ( INTEGER_MODEL,
                                    REAL_MODEL,
                                    STRING_MODEL,
                                    LOGICAL_MODEL,
                                    POINTER_MODEL,
                                    LIST_MODEL,
                                    DEFINED_TYPE_MODEL );-- except structure,
                                                         -- array, or set
   END_ENTITY;

ENTITY SET_MODEL ( 1998);                               -- not implemented
   MINIMUM           : INTEGER;
   MAXIMUM           : INTEGER;
   CONSTITUENT       : POINTER TO ( INTEGER_MODEL,
                                    REAL_MODEL,
                                    STRING_MODEL,
                                    LOGICAL_MODEL,
                                    POINTER_MODEL,
                                    SET_MODEL,
                                    DEFINED_TYPE_MODEL ); -- except structure,
                                                          -- array, or list
   END_ENTITY;

ENTITY STRUCTURE_MODEL ( 1009);                         -- not implemented
   CONSTITUENT       : ARRAY ( 1:# ) OF
                       POINTER TO ( ATTRIBUTE_MODEL );
   END_ENTITY;
(*
```

```
*)

ENTITY CLASS_MODEL ( 1017);
   NAME               : STRING (16);
   KIND               : INTEGER;
   COMMENT            : STRING (150);
   CONSTITUENT        : ARRAY ( 1:# ) OF
                        POINTER TO ( ENTITY_MODEL,
                                     CLASS_MODEL );

   END_ENTITY;

ENTITY SUBSCHEMA_MODEL ( 1021);
   NAME               : STRING (16);
   COMMENT            : STRING (150);
   ADB_SIZE           : INTEGER;
   PHYSICAL           : LOGICAL;
   CONSTITUENT        : ARRAY ( 1:# ) OF
                        POINTER TO ( ENTITY_MODEL,
                                     CLASS_MODEL );

   END_ENTITY;

ENTITY SCHEMA_MODEL ( 1089);                             -- not implemented
   NAME               : STRING (16);
   SCHEMA_VERSION     : INTEGER;
   DATE               : STRING (9);
   STATUS             : T_STATUS;
   CONSTITUENT        : ARRAY ( 1:# ) OF
                        POINTER TO ( ENTITY_MODEL,
                                     CLASS_MODEL,
                                     GLOBAL_ATTRIBUTE_MODEL,
                                     SUPERTYPE_MODEL,
                                     DEFINED_TYPE_MODEL,
                                     SUBSCHEMA_MODEL );

   END_ENTITY;

(*
```

```
*)

TYPE
   T_ARRAY_TYPE      : ENUMERATION OF ( nil_0_DT,
                                        nil_1_DT,
                                        nil_2_DT,
                                        nil_3_DT,
                                        ARRAY_DT,
                                        LIST_DT,
                                        SET_DT );
   T_REQD            : ENUMERATION OF ( REQ, OPT );

   T_STATUS          : ENUMERATION OF ( FROZEN, UNFRZN );

   T_TYP             : ENUMERATION OF ( INTEGER_DT,
                                        REAL_DT,
                                        STRING_DT,
                                        LOGICAL_DT,
                                        ARRAY_DT,
                                        LIST_DT,
                                        SET_DT,
                                        ENUMERATION_DT,
                                        POINTER_DT,
                                        STRUCTURE_DT,
                                        DEFINED_TYPE_DT,
                                        CLASS_DT,
                                        SUPERTYPE_DT,
                                        ENTITY_DT,
                                        ATTRIBUTE_DT,
                                        GLOBAL_ATTRIB_DT,
                                        SUBSCHEMA_DT,
                                        SCHEMA_DT );

GLOBAL
   VERSION           : INTEGER;
   SYS_IDENT         : INTEGER;
   IDENT             : INTEGER;
   TYP               : T_TYP;
```