

DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

VECTOR DATA STRUCTURE CONVERSION
AT THE EROS DATA CENTER

Jan W. van Roessel, TGS Technology, Inc.
Stuart W. Doescher, U.S. Geological Survey

Open-File Report 87-309

Sioux Falls, South Dakota

1986

VECTOR DATA STRUCTURE CONVERSION AT THE EROS DATA CENTER

by
Jan W. van Roessel ¹
Stuart W. Doescher ²

Abstract

With the increasing prevalence of GIS systems and the processing of spatial data, conversion of data from one system to another has become a more serious problem. This report describes the approach taken to arrive at a solution at the EROS Data Center.

The report consists of a main section and a number of appendices. The methodology is described in the main section, while the appendices have system specific descriptions. The overall approach is based on a central conversion hub consisting of a relational database manager and associated tools, with a standard data structure for the transfer of spatial data. This approach is the best compromise between the two goals of reducing the overall interfacing effort and producing efficient system interfaces, while the tools can be used to arrive at a progression of interface sophistication ranging from toolbench to smooth flow.

The appendices provide detailed information on a number of spatial data handling systems and data structures and existing interfaces as well as interfaces developed with the described methodology.

Publication approved by the Director, U.S. Geological Survey.

DISCLAIMER

Any use of trade names is for descriptive purposes only and does not imply endorsement by the U.S. Geological Survey.

¹ TGS Technology, Inc. Work performed under U.S. Geological Survey contract 14-08-0001-22521.

² U.S. Geological Survey

Contents

1	Acknowledgments	1
2	Introduction	1
2.1	Definitions	1
2.2	Objectives	2
2.3	Workplan	3
2.4	Scope	3
3	Current Status	4
3.1	Systems	4
3.1.1	Input	5
3.1.2	Processing	5
3.1.3	Output	6
3.2	Data Structures	6
3.3	Interfaces	8
3.4	Data Flows	11
4	Development of Methodology	11
4.1	Interfacing models	12
4.2	The Relational Approach	12
4.3	Relational Data Structure	13
4.4	Restructuring Tools	15
4.5	Relational Operators	18
4.6	Topological Tools	18
4.7	Conversion Experience to Date	18
4.8	Concluding Remarks	19
5	References	19
A	System Descriptions	21
A.1	ADS	21

A.2	AGIS/GRAM	23
A.3	AHDS	25
A.4	ALMRS	25
A.5	AMS	26
A.6	ARC/INFO	26
A.7	COS	29
A.8	CRAIGES	29
A.9	DCASS	30
A.10	DLG100K	30
A.11	GES	31
A.12	GIMMS	31
A.13	GIRAS	32
A.14	GPPU	32
A.15	ICARAS	32
A.16	INTERGRAPH	33
A.17	LAS	33
A.18	MOSS	34
A.19	SDL/SGL	37
A.20	RIPS	37
A.21	TIGER	37
A.22	UCLGES	38
A.23	VDI(RIM)	40
B	Data Structures	42
B.1	BNF Notation	42
B.1.1	BNF Defined in BNF	43
B.2	Tree Structures	43
B.3	Characteristics Description	44
B.3.1	Vector Data Structure Characteristics	44

C	Data Structure Descriptions	50
C.1	AGIS/GRAM	50
C.1.1	AGIS/GRAM BNF	50
C.2	AMS	51
C.2.1	AMS BNF	51
C.2.2	AMS Tree Structure	53
C.3	ARC/INFO	55
C.3.1	ARC/INFO BNF	55
C.4	DLG	57
C.4.1	DLG BNF	57
C.4.2	DLG Tree Structure	60
C.4.3	DLG Characteristics	63
C.5	FGEF	65
C.5.1	FGEF BNF	65
C.6	GES	72
C.6.1	GES BNF	72
C.6.2	GES Tree Structure	75
C.6.3	GES Characteristics	78
C.7	IDIMS	80
C.7.1	IDIMS ICARAS BNF	80
C.7.2	IDIMS ICARAS Tree Structure	81
C.8	ISO 8211	81
C.8.1	ISO 8211 BNF	81
C.9	MOSS	84
C.9.1	MOSS ADD FORMAT	84
C.9.2	MOSS IMPORT/EXPORT BNF	85
C.9.3	MOSS IMPORT/EXPORT Tree Structure	86
C.10	VDI(RIM)	86
D	Input Interfaces	88
D.1	ADS from MAP	88

D.2	AGIS/GRAM from MAP	89
D.3	AGIS/GRAM from VDI(RIM)	89
D.4	AHDS from MAP	89
D.5	AMS from MAP	90
D.6	ARC/INFO from ARC/INFO IMPORT/EXPORT	93
D.7	ARC/INFO from DIME	93
D.8	ARC/INFO from DLG	93
D.9	ARC/INFO from DLG OPTIONAL	93
D.10	ARC/INFO from GIMMS	93
D.11	ARC/INFO from MAP	93
D.12	ARC/INFO from PIOS	94
D.13	ARC/INFO from SVF	94
D.14	ARC/INFO from VDI(RIM)	94
D.15	CONTRACTOR FMT from MAP	94
D.16	COS from MOSS	94
D.17	CRAIGES from MAP	95
D.18	GES from DLG and from DLG OPTIONAL	95
D.19	GES from MAP	96
D.20	GIMMS from ARC/INFO	96
D.21	GIRAS from CONTRACTOR FMT	96
D.22	GIRAS from MAP	96
D.23	INTERGRAPH from DLG	96
D.24	INTERGRAPH from MAP	97
D.25	INTERGRAPH from SIF	98
D.26	MOSS from AHDS	98
D.27	MOSS from AMS	98
D.28	MOSS from DIME	98
D.29	MOSS from DLG OPTIONAL	98
D.30	MOSS from MAPS	98
D.31	MOSS from MOSS IMPORT/EXPORT	99

D.32	MOSS from VDI(RIM)	99
D.33	TIGER from DLG	99
D.34	UCLGES from UCLGES UPDATE	99
D.35	VDI(RIM) from ALMRS	100
D.36	VDI(RIM) from AMS	100
D.37	VDI(RIM) from ARC/INFO	100
D.38	VDI(RIM) from DLG	101
D.39	VDI(RIM) from DLG OPTIONAL	101
D.40	VDI(RIM) from FGEF	101
D.41	VDI(RIM) from GES	101
D.42	VDI(RIM) from INTERGRAPH	102
D.43	VDI(RIM) from LAS	102
D.44	VDI(RIM) from LUNR	102
E	Output Interfaces	103
E.1	ADS to DLG OPTIONAL	103
E.2	ADS to MAP	103
E.3	ADS to MOSS IMPORT/EXPORT	103
E.4	AGIS/GRAM to MAP	103
E.5	AHDS to MOSS	104
E.6	AMS to MAP	104
E.7	ARC/INFO to ARC/INFO IMPORT/EXPORT	104
E.8	ARC/INFO to DLG OPTIONAL	104
E.9	ARC/INFO to GIMMS	104
E.10	ARC/INFO to MAP	104
E.11	ARC/INFO to PIOS	105
E.12	ARC/INFO to SVF	105
E.13	ARC/INFO to VDI(RIM)	105
E.14	CRAIGES to UCLGES UPDATE	105
E.15	DLG to DLG OPTIONAL	106
E.16	EXTELE to MAP	106

E.17	GES to DLG OPTIONAL	107
E.18	GES to MAP	107
E.19	GES to VDI(RIM)	107
E.20	GIMMS to ARC/INFO	108
E.21	GIRAS to DLG	108
E.22	INTERGRAPH to DLG	108
E.23	INTERGRAPH to EXTELE	109
E.24	INTERGRAPH to VDI(RIM)	109
E.25	MOSS to COS	110
E.26	MOSS to MAP	110
E.27	MOSS to MAPS	110
E.28	MOSS to MOSS IMPORT/EXPORT	110
E.29	TIGER to DLG	111
E.30	UCLGES to DLG	111
E.31	VDI(RIM) to AGIS/GRAM	111
E.32	VDI(RIM) to ARC/INFO	112
E.33	VDI(RIM) to DLG OPTIONAL	112
E.34	VDI(RIM) to FGEF	112
E.35	VDI(RIM) to Intergraph	113
E.36	VDI(RIM) to LAS	113
E.37	VDI(RIM) to MAP	113
E.38	VDI(RIM) to MOSS	113
F	Data Flows	115
F.1	Data Flows by Input Data Structure	115
F.2	Data Flows by Output Data Structure	127

1 Acknowledgments

Gene Fosnight, Victor Conocchioli, Norman Bliss, Dean Tyler, Mike Wehde and others made valuable contributions to this report. Fosnight in particular contributed to the main section, which is based on the paper "A Relational Approach to Vector Data Structure Conversion," of which he was a co-author with the senior author of this report. Both Fosnight and Conocchioli have contributed substantially to the development of the vector data interfacing approach described. Stanton Moll and Dean Tyler have been instrumental in the testing and assembly of the interfacing tools into "smooth" flow interfaces.

2 Introduction

Scientists and managers in the U.S. Geological Survey (USGS) and other Federal agencies are making increasing use of geographic information systems (GIS's) to assist complex decisionmaking processes. This use has created a growing need to access geographic and cartographic data from a variety of sources. However, building data bases for a specific GIS is difficult because the data that are needed have been acquired and compiled for different purposes at different resolutions and projections and have been stored in different data structures and formats. As stated in Nyerges, (1984, p77): "The concerns for common data formats and geocoding conversions cut across all disciplines from computer-assisted cartography to remote sensing. At this stage it is intractable, clumsy, and inefficient for diverse users to use a given data set. It is also difficult to assemble a data set from a variety of sources for general use. The time has come to consider coordinated data systems which retrieve data from the various archives and combine and deliver them to the users in a common format..."

This report describes a solution to the problem of transferring data among the different systems and subsystems at the EROS Data Center (EDC) that support vector data. It is based upon experiences gained with EDC's various GIS's using data from a variety of sources, on cooperative projects between USGS and other Federal agencies. The report describes a methodology that has been developed to handle the problem of interfacing, provides a description of the various GIS and support systems used to prepare and handle vector data, and describes the current status of the interfaces among these systems.

2.1 Definitions

For the purposes of this report the following definitions hold:

- Support system - The spatial data handling software and interfaces available at the EROS Data Center.
- Vector support system - The vector data handling aspect of the support system.
- System - Basic building block of the support system. A major software package, usually associated with a vendor or certain hardware.
- Subsystem - A set of distinct capabilities with a common orientation within a system.
- Module - An executable system unit, mostly with a single functional capability.

- **Communication** - The process by which data are transferred from one system to another so that the receiving system can correctly process the data according to the sending system's original specifications without misinterpreting the intent of the data.
- **Vector format** - The word vector refers to a type of spatial data arrangement whereby the data are encoded in coordinates (or addresses) relative to the vertices of a graph. Point data can be considered a degenerate case of this arrangement. The word format refers to the way in which this type of data is arranged on a computer accessible medium. Format may have both the connotation of a certain type of data structure or it may refer to the sequential arrangement of the data on a medium. In the former case the word format may be somewhat misleading.
- **Data structure** - The manner in which basic data elements are organized into units, the units are organized into other units, and so on, and the relationships between these elements and units.

Non-spatial data structures have a physical and a logical representation. Spatial data structures have an added dimension, the spatial representation, which can be partitioned according to location and topology. Topology refers to the non-metric logical-spatial relationships among the various features in the data structure (Fegeas, 1982), and it can therefore be viewed as the intersection of the logical and spatial representation of the data structure. Attribute data, or the links to attribute data, are also integral to handling spatial data because attributes are related to the topology. Conversion of one spatial data structure to another must therefore take into consideration the following properties of the data structures: (a) physical, (b) logical, (c) logical-spatial (topological), (d) locational, and (e) attributal.

- **Interface** - A mechanism by which one data structure can be converted directly to another for the purpose of communication between systems or subsystems.
- **Data flow** - A communication involving more than one interface.
- **Interfacing system** - A system designed as a buffer system between two other systems for the purpose of allowing indirect communication. This implies that the communications with the buffer system are designed specifically to accomplish the indirect communication.

2.2 Objectives

Using the above definitions, the overall goals of the investigation were: (a) to describe the current status of the vector support system, and (b) to plan future development of this system. These were accomplished through the following more specific objectives:

1. To assess the current state of the vector support system by describing and categorizing its systems, interfaces and data structures (internal, as well as external).
2. To identify and categorize future capability and data product requirements for the vector support system.
3. To suggest new systems and interfaces to meet these requirements.
4. To conduct research to find a systematic approach for the application of a variety of vector data structures in the form of a generic solution to the interfacing problem.

Although objective 2 would also include analysis and input and output issues, in the ensuing investigation the focus was on data transfer and interfacing issues. From this perspective the investigation considered elements of the vector support system that are at the EROS Data Center, and also considered components that are important to NMD and other Department of the Interior (DOI) agencies.

2.3 Workplan

The workplan for the investigation included the following elements:

1. Develop a base guideline for organizing the work and the results.
2. Identify and briefly describe the systems of the vector support system, and external systems peripheral to the vector support system.
3. Identify and concisely describe each of the vector data structures and each relevant peripheral external vector data structure.
4. Identify and describe existing interfaces.
5. Investigate and describe existing vector data structure interfacing methods and programs (outside the EROS Data Center) relevant to the support system.
6. Assess the feasibility of a generic approach to the solution of the interfacing problem and conduct trial experiments, if necessary.
7. Identify requirements for future products and capabilities.
8. Suggest new systems to be acquired and interfaces to be developed.
9. Evaluate possible approaches and make recommendations based on advantages, disadvantages, and effort required.
10. Complete final report.

2.4 Scope

The steps in the workplan can be assigned to two categories (1) description of current status, and (2) development of methodology. The main body of this report is subdivided according to these two categories, with primary emphasis on a description of the recommended methodology. In developing this methodology, some new interfaces were created, thereby changing the status from how it existed at the beginning of the study. Descriptions of systems, data structures and interfaces in existence at the conclusion of the current phase of the vector data interface development at EROS Data Center are provided in the appendices to the report.

With the large number of possible systems and interfaces that could be considered, some systems or interfaces may have been excluded. Emphasis was placed on those systems that are used by, or are of interest to the Department of the Interior; only those systems and interfaces of direct interest to the Data Center have been fully described.

One has to take into consideration also the ongoing nature of the interfacing work at the EROS Data Center. Many aspects remain to be realized and are awaiting practical testing.

Given these conditions, this report is far from complete, and probably never will be complete. Conditions change too fast to obtain a complete picture at a given time. However, this report is meant to be a useful starting point for an introduction to the world of data structures, systems interfaces, and data flows, and to the EROS Data Center's approach to this complex system. A second function of this report, with continued updating, is to serve as a framework for organizing information and knowledge gained in the course of the implementation of interfaces and data flows.

3 Current Status

To gain some insight into the data structure conversion and interfacing problem, it was necessary to take stock of existing components of the vector support system. The components judged most important were individual systems, data structures, interfaces, and data flows, each of which is described in detail in this report.

In the following sections the rationale for the selection of the components is presented, and the individual components are listed; detailed descriptions are found in the appendices to the report.

3.1 Systems

The systems that were judged relevant for inclusion in the study are described in the following sections. These descriptions reflect the current status of the support system.

As with most GIS's the software systems for the vector support system can be grouped into three distinct classes: (1) operating-system software, (2) graphics, database, and statistical-support software, and (3) applications-oriented software. These three classes can be thought of as concentric rings, with the third class as the outer ring. Although the two inner rings are the foundation for all other activities, they are beyond the scope of this report, and will be mentioned only where appropriate.

The third ring can be divided into three segments: (1) input, (2) processing, and (3) output. These categories will be used to classify the systems in this report. Some systems undoubtedly will have components that qualify the system in more than one category, in which case its name will be listed in each appropriate grouping.

Not all systems are of equal interest to the EROS Data Center, so a priority code is presented. All systems of first priority are described in full detail, while others may be mentioned briefly, or not at all. Systems may have different priorities in different categories.

The term "system" is rather flexible: a system can be a standalone system with its own user interface; it may be a subroutine library that may be used by other systems, or it may be a collection of programs that interface to the same database management system.

3.1.1 Input

The systems listed in table 1 can be used to capture digital data from hardcopy maps.

Table 1: Input Systems

SYSTEM	PRIORITY
AGIS/GRAM	1
AMS	1
ARC/INFO	1
CRAIGES	1
GES	1
INTERGRAPH	1
ADS	2
AHDS	3
GIRAS	3

3.1.2 Processing

The systems listed in table 2 can be used to process and significantly change data structures and contents of files in order to perform user invoked information, analysis, or restructuring functions.

Table 2: Processing Systems

SYSTEM	PRIORITY
AGIS/GRAM	1
ARC/INFO	1
INTERGRAPH	1
LAS	1
MOSS	1
UCLGES	1
VDI(RIM)	1
GIRAS	2
RIPS	2
SDL/SGL	2
SURFACE II	2
AALRS	3
ALMRS	3
DCASS	3
GPPU	3
PIOS	3
TIGER	3

3.1.3 Output

The systems listed in table 3 can be used to produce hardcopy map output. However, these systems are more than general graphics systems and are generally oriented to providing GIS type hardcopy graphics, such as maps and surface displays.

Table 3: Input Systems

SYSTEM	PRIORITY
AGIS/GRAM	1
ARC/INFO	1
COS	1
CRAIGES	1
GES	1
GIMMS	1
INTERGRAPH	1
LAS	1
MOSS	1
SDL/SGL	1
SURFACE II	1
VDI(RIM)	1

3.2 Data Structures

Table 4 is a list of data structures associated with the input, processing, and output systems. Some data structures carry the same name as the system to which they pertain, in which case it represents the internal data structure for that system.

Priorities assigned to the data structure are based on current or past uses of the data structure and its importance to organizing interfaces and data flows at the Data Center.

The data structure MAP refers to a hardcopy map.

Table 4: Data Structures

NAME	PRIORITY
AGIS/GRAM	1
AMS	1
ARC IMP/EXP	1
ARC/INFO	1
DLG	1
DLG OPTIONAL	1
FGEF	1
GES	1
GIMMS	1
IDIMS	1
INTERGRAPH	1
LAS	1
LUNR	1
MAP	1
MOSS	1
MOSS IMP/EXP	1
VDI(RIM)	1
ADS	2
COS	2
EXTELE	2
GIRAS	2
MAPS	2
AHDS	3
CONTRACTOR FMT	3
DIME	3
PIOS	3
SIF	3
SVF	3
UCLGES UPDATE	3

3.3 Interfaces

An interface is a mechanism by which one data structure can be directly converted into another data structure for the purpose of communication among systems or subsystems. Interfaces therefore relate pairs of data structures, so with N data structures there are $N(N-1)$ possible interfaces.

Each interface can be assigned a classification based upon its status, and a cost can be assigned to it in order to determine the most cost-effective data flows (see section 3.4):

Table 5: Status Codes

CODE	DESCRIPTION	COST
S	smooth flow	1
T	tool bench	5
D	digitizing	100
M	hardcopy map	1
O	not in-house	10
P	planned	20
U	unknown	30
N	non-existing	9999

The following tables show systems, possible inputs and possible outputs for all but the non-existing category. In addition, the status and priority of each interface is indicated.

Table 6: Input Interfaces

SYSTEM	FROM	STATUS	PRIORITY
AGIS/GRAM	MAP	D	1
AGIS/GRAM	VDI(RIM)	T	1
ARC/INFO	ARC IMP/EXP	S	1
ARC/INFO	DLG	S	1
ARC/INFO	DLG OPTIONAL	S	1
ARC/INFO	MAP	D	1
ARC/INFO	VDI(RIM)	S	1
GES	MAP	D	1
INTERGRAPH	MAP	D	1
MOSS	MOSS IMP/EXP	S	1
MOSS	VDI(RIM)	S	1
VDI(RIM)	AMS	T	1
VDI(RIM)	ARC/INFO	S	1
VDI(RIM)	DLG	S	1
VDI(RIM)	DLG OPTIONAL	P	1
VDI(RIM)	FGEF	P	1
VDI(RIM)	GES	S	1
VDI(RIM)	INTERGRAPH	T	1
VDI(RIM)	LAS	S	1
ADS	MAP	D	2
AMS	MAP	D	2
ARC/INFO	DIME	S	2
COS	MOSS	S	2
CRAIGES	MAP	D	2
GES	DLG	S	2
GES	DLG OPTIONAL	S	2
INTERGRAPH	DLG	S	2
MOSS	AMS	S	2
MOSS	DLG OPTIONAL	S	2
MOSS	MAPS	S	2
UCLGES	UCLGES UPDATE	S	2
VDI(RIM)	ALMRS	P	2
VDI(RIM)	LUNR	S	2
AHDS	MAP	D	3
ARC/INFO	GIMMS	S	3
ARC/INFO	PIOS	S	3
ARC/INFO	SVF	S	3
CONTRACTOR FMT	MAP	D	3
DCASS	DLG	O	3
GIMMS	ARC/INFO	S	3
GIRAS	CONTRACTOR FMT	O	3
GIRAS	MAP	D	3
INTERGRAPH	SIF	S	3
MOSS	DIME	P	3
TIGER	DLG	O	3

Table 7: Output Interfaces

SYSTEM	TO	STATUS	PRIORITY
AGIS/GRAM	MAP	M	1
ARC/INFO	ARC IMP/EXP	S	1
ARC/INFO	DLG OPTIONAL	S	1
ARC/INFO	MAP	M	1
ARC/INFO	VDI(RIM)	S	1
GES	DLG OPTIONAL	O	1
GES	MAP	M	1
GES	VDI(RIM)	S	1
INTERGRAPH	VDI(RIM)	T	1
MOSS	COS	S	1
MOSS	MOSS IMP/EXP	S	1
UCLGES	DLG	S	1
VDI(RIM)	AGIS/GRAM	S	1
VDI(RIM)	ARC/INFO	S	1
VDI(RIM)	FGEF	P	1
VDI(RIM)	INTERGRAPH	T	1
VDI(RIM)	LAS	S	1
VDI(RIM)	MAP	M	1
VDI(RIM)	MOSS	S	1
ADS	DLG OPTIONAL	O	2
ADS	MOSS IMP/EXP	O	2
AMS	DFAD	O	2
AMS	DLG	O	2
CRAIGES	UCLGES UPDATE	S	2
DLG	DLG OPTIONAL	O	2
EXTELE	MAP	M	2
INTERGRAPH	DLG	S	2
INTERGRAPH	EXTELE	S	2
MOSS	MAP	M	2
MOSS	MAPS	S	2
VDI(RIM)	DLG OPTIONAL	P	2
ADS	MAP	M	3
AHDS	MOSS	U	3
AMS	MAP	M	3
ARC/INFO	GIMMS	S	3
ARC/INFO	PIOS	S	3
ARC/INFO	SVF	S	3
DCASS	MAP	M	3
GIMMS	ARC/INFO	S	3
GIRAS	DLG	O	3
TIGER	DLG	O	3

3.4 Data Flows

Data flows are communications involving more than one interface. When it is impossible to convert directly from one data structure to another, it is possible that the desired goal may be reached by passing through one or more intermediate data structures.

Each data structure may be thought of as a node in a directed graph in which each edge has an associated cost. This cost can be infinite when an interface does not exist. Finding the optimum manner to change an input data structure into an output data structure equates to following the least-cost path connecting the input and output node. To find all possible paths or data flows, one can locate all possible least-cost paths in the graph.

The data flows shown in tables F1 and F2 were derived by this method. Least-cost paths were found using the algorithm developed by Dijkstra (1959). A cost matrix was built by assigning costs to the interfaces listed in tables 6 and 7, depending on the type of interfaces. A fairly high cost was assigned to a digitizing interface, in comparison to digital interfaces (even though in practice the reverse may be true), in order to exclude paths with an intermediate digitizing step. All paths with a cost higher than the basic digitizing cost were excluded, so that only input digitizing steps are shown. In addition, higher costs were assigned to interfaces that are not in-house, to prevent least-cost path selections with outside processing requirements. All selected costs are purely hypothetical, and are based solely on relative judgement of the efficiency of the different types of interfaces. If the exploratory nature of this type of exercise would change in the future, one might apply more accurate measures of efficiency to assign costs.

The reader should keep in mind that the data flows listed in tables F1 and F2 are potential data flows, many of which have never been tried. However they should be helpful for charting new paths that potentially could be useful. By keeping track of those paths that have been tried and by assigning appropriate costs, a more realistic road map for data conversion at the EROS Data Center can be developed.

4 Development of Methodology

With a growing number of geographic information systems and digital spatial data sets in use, an ability to adapt data from one system to another has become much sought after. Yet such adaptations are not easily accomplished because of the special characteristics of spatial data. Non-spatial digital data have two main aspects with respect to their organization: physical and logical (Martin, 1977). Spatial data can be thought of as having two additional traits, spatial-logical (topological) and locational (coordinate system, projection, measurement units).

Adaptation is, therefore, not a simple reformatting problem, as with other data sources, but rather is a far more complex proposition with spatial implications that cannot be readily deduced by an initial inspection of the data. It may even be impossible to convert certain types of data structures, without a major processing effort. However, data structures of certain types can be converted from one system to another with a reasonable amount of effort.

One of the first steps in developing a conversion methodology is to obtain a consistent description of the vector data structures involved, and to characterize them, in order to anticipate possible conversion problems. For this purpose a Backus Naur Form (BNF) type of notation was developed

to describe the data organization concisely and consistently. The use of BNF for describing the syntax of geographic data was earlier suggested by Cox, Aldred, and Rhind (1980).

A second step in the description process has been to cull the most important characteristics of each data structure. The five major spatial data aspects – physical, logical, logical-spatial (topological), locational, and attributal– were further reduced to constituent components, and this process was repeated until a satisfactory level of detail was reached. The overall breakdown was then used as a checklist against each BNF description, as well as other documentation, to obtain a set of hierarchical characteristics of the data structure under consideration.

4.1 Interfacing models

One approach to the interfacing of N data structures is to match each one directly with every other one, thereby creating $N(N-1)$ interfaces. Alternatively, one may implement a unidirectional “ring” concept whereby each system interfaces to one and only one other system. This yields the minimum number of N interfaces, but its drawback is that data may have to pass through $N-1$ other systems before a target system is reached, with proportional processing time and potential for errors. A third approach uses an intermediate data structure to which each original structure must be converted before it is transformed to the target configuration. This method requires $2N$ interfaces and is the one most commonly adopted; it is also the approach used here. Quite often this method is associated with an intermediate tape format. However our concern was to define an intermediate data structure maintained in a database system, which can in turn be readily transferred between systems.

The Workgroup on Data Organization of the National Committee for Digital Cartographic Data Standards (Nyerges, 1984) has proposed a model for digital cartographic data conversion to an intermediate structure as follows:

$$S \xrightarrow{T1} I1 \xrightarrow{T2} I2 \quad (1)$$

where S represents the source data structure, $T1$ is a transformation producing an “intermediate” intermediate structure $I1$, and $T2$ is a transformation producing the desired standard interchange form. In this representation $I1$ is necessary because of a differences in hardware and software related to the data structures S and $I2$. The Workgroup mentions the importance of the data schema for this model. The schema defines the logical aspect of the data structure and forms a link with the physical representation. The above model is only for the conversion from source to intermediate although the conversion from intermediate to target is similar.

A sobering note is the complete absence of any kind of explicit logical or topological schema in any of the existing data structures considered. However, for our approach, the model proves to be useful for considering the different transformation stages in the data structure conversion.

4.2 The Relational Approach

Other investigators have applied the relational model or relational concepts to geographic data handling. At least one prototype geographic information system (GEOQUEL) has been constructed using the relational database management system INGRES (Go, Stonebraker, and Williams, 1975). Shapiro and Haralick (1980) have proposed spatial data structures based on relations in relation trees. Garret and Foley (1982) have reported an attempt to build an experimental graphics system using relations, where events trigger continuously evaluated qualified updates, through which the

consistency between relations is always maintained. A graphics application (IMPS) based on a relational data base (IMS) was developed by IBM in the United Kingdom (IBM, 1979).

We have selected the relational data model for the intermediate data structures for a number of reasons, one being the elegance and simplicity of the data representation resulting from the use of simple tabular files and a complete absence of pointers.

A second consideration in using the relational approach is the availability of the relational algebra and its unique relational operators. To minimize the effort associated with creating a number of transformation functions for different input models, it follows logically that one should attempt to develop higher level tools that can be reused for each conversion. The aim of our study has been to investigate whether relational operators fall in this category.

A third reason for considering the relational model is the availability of a number of different software systems, including many operating in the microcomputer environment. In the future, it will be possible to consider the implementation of the relational model in firmware/hardware implementations such as a database machines. Research for this paper was performed using the Relational Information Management (RIM) system.

Normalization is one of the important considerations in relational database theory. Five normal forms are recognized (Kent, 1983). The first normal form is the most crucial and states that "all occurrences of a record type must contain the same number of fields." This is not an option, but a matter of definition: relational database theory does not deal with variable-length records or vector or matrix type attributes. Our proposed intermediate data structure is strictly of first normal form, which allows us to use the full range of relational operators.

Given the simplicity of flat files, the database schema also becomes very simple and consists of a description of the individual attributes and the composition of the relations in terms of these attributes. Although much can be learned about the contents of a database by inspection of the schemas, the schema is still inadequate for describing the spatial organization of the data.

4.3 Relational Data Structure

The spatial data structure selected is the arc-node type. A full complement of spatial elements is represented: regions, polygons, arcs, nodes, and vertex points (line and point type elements are not further considered in this paper). We define a polygon as a closed figure bounded by connected straight line segments, a region as consisting of an outer polygon and zero or more island polygons, and nodes as arc termination points.

The mutual referencing of the elements is resolved as follows (the relational theory excludes direct pointers, so all referencing is in terms of element identifiers). Given the hierarchy obtained by defining more complex elements in terms of their component parts—regions, polygons, arcs, nodes and vertex points—the most austere organization is obtained by having each element reference its subordinated elements. For example, by defining the element identifier attributes *regnum* (region number), *polnum* (polygon number), *arcnum* (arc number), *nodenum* (node number), *strtnode* (start node), *endnode* (end node), and *pointnum* (point number), as well as the coordinate attributes *x*, *y*, we can propose to maintain the following relations:

regpol : *regnum, polnum*
polarc : *polnum, arcnum*

$$\begin{aligned}
\text{arcnode} &: \text{arcnum}, \text{strtnode}, \text{endnode} \\
\text{arcpoint} &: \text{arcnum}, \text{pointnum} \\
\text{nodepoint} &: \text{nodenum}, \text{pointnum} \\
\text{pointxy} &: \text{pointnum}, x, y
\end{aligned}
\tag{2}$$

where strtnode and endnode refer to nodenum in the nodepoint relation.

The above scheme has downward referencing only. Reverse references can be established with relational operators. For instance, to compute an "arcpol" relation containing polygon numbers by arc, one can isolate the arc column from the polarc relation using the project operator, remove duplicate arc numbers (if not included in the project), and then join this temporary relation with the polarc relation using arcnum as the join item, while maintaining the derived arc ordering.

Spatial references, such as which regions are to the left and which are to the right of the arc, cannot be so easily established in this manner. Considering the frequency with which this type of information is used, one is led to consider storing some higher level information at the lower levels. In particular, an archdr relation can be created by renaming arcnode as archdr, and by adding two additional attributes, lftreg and rgtreg, as follows:

$$\text{archdr} : \text{arcnum}, \text{strtnode}, \text{endnode}, \text{lftreg}, \text{rgtreg} \tag{3}$$

Also, for network tracking, a nodearc relation is desirable in which arcs emanating from each node are stored:

$$\text{nodearc} : \text{nodenum}, \text{arcnum} \tag{4}$$

The above considerations, and the removal of the pointxy relation, lead to our proposed intermediate data structure I2:

$$\begin{aligned}
\text{regpol} &: \text{regnum}, \text{polnum} \\
\text{polarc} &: \text{polnum}, \text{arcnum} \\
\text{archdr} &: \text{arcnum}, \text{strtnode}, \text{endnode}, \text{lftreg}, \text{rgtreg} \\
\text{arcxy} &: \text{arcnum}, x, y \\
\text{nodearc} &: \text{nodenum}, \text{arcnum} \\
\text{nodezy} &: \text{nodenum}, x, y
\end{aligned}
\tag{5}$$

A decision must also be made on how to cope with complex islands. The union of the polygons of a complex island constitutes the hole in the outer polygon. Not all arcs of the island polygon are, therefore, island arcs. We have solved this problem by defining the outer boundary of the complex island to be a separate polygon. This gives the advantage of being able to join the regpol relation with polarc and arcxy relations to directly obtain a regxy relation in which each region is explicitly represented in sequence, complete with all its islands, including complex ones. In this way, the topology is defined in a simpler, clearer organization. Thus, our approach is to organize the topological information implicitly by position, rather than explicitly.

Position is critical within most relations. The vertex point coordinates in the arcxy relation are obviously stored in positional sequence. Less obviously, the arcs in the nodearc relation are stored in a clockwise rotation around the node. The arcs in the polarc relation are stored in clockwise order

for outer polygons and counter-clockwise for island polygons. Finally, in *regpol*, the first polygon for each region is the outside polygon.

When constructing polygon or region boundary lists, one must keep in mind that arcs must be traversed in different directions for the regions to the left and to the right of the arc. The traversal direction is stored implicitly in the *archdr* relation. If the polygon currently being traversed corresponds to the right region in the *archdr* relation, the arc is traversed in the clockwise direction. The converse is true when it matches the left region.

Other non-spatial relations are defined in addition to the above core of spatial relations. Those for attribute data are most important. They are visualized as a nested set, each consisting of one primary and a number of secondary relations, namely:

$$attprime : elnum, att1, att2, att3...etc. \quad (6)$$

This relation links the primary attributes to the spatial elements through the common element identifier. In addition, each attribute in the primary relation can be a key for an entry in secondary relation in which that attribute is further detailed. For instance, one may have:

$$attsec : att1, satt1, satt2, satt3...etc. \quad (7)$$

This allows new attribute relations to be created in which the attribute detail can be varied at will, using join operators.

The overall interfacing effort is reduced not only by the total number of required interfaces but also by the amount of time required to construct each interface. This time is reduced further when suitable higher level tools are available.

According to the Workgroup model, these tools can be assigned to T1 and T2 categories. The database system used here is perhaps an exception, as it can be assigned to both categories because it allows the data structures I1 and I2 to exist.

The scope of T1 is the following. The scheme being tested makes use of the fact that the input data structures can be cast in the form of files with fixed- or variable-length records. These files can be translated on a one-to-one basis to unnormalized (with respect to the first degree) records in the RIM system, storing one record per relation row. One would not expect this capability in an RDBMS, but RIM permits records with variable-length attributes, although relational operations cannot be performed on variable-length items. With the transfer, a schema is developed for the unnormalized data structure in the relational database, and I2 represents the normalized intermediate structure presented earlier.

In the T2 category, a breakdown can be made considering tools that allow data to be restructured from T1 to T2, and those to be used to repair errors, perform checks, and make I2 more complete.

4.4 Restructuring Tools

The transformation from the unnormalized form to a normalized representation is accomplished by a specially developed program (RIMNET) that allows one to track data items along predefined access

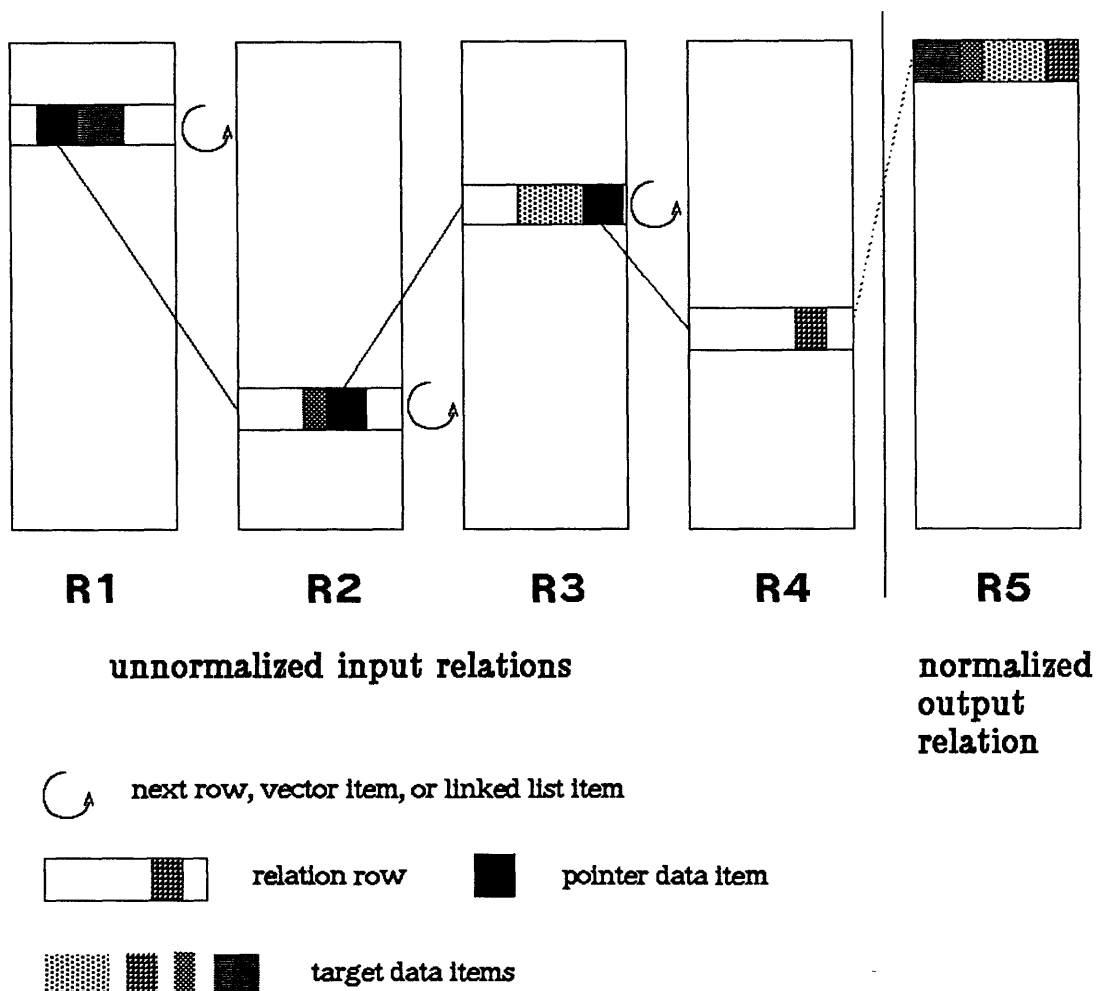


Figure 1: Schematic of RIMNET operation.

paths in the unnormalized representation. This program is one of the main conversion tools. Using relational operators, the normalized forms are then further reduced to the essential relations in the intermediate relational format (I2).

RIMNET has been designed to serve as a normalization tool, and also to cope with such data organizations as linked lists that cannot easily be unscrambled with relational operators. The principal idea behind the program is demonstrated in figure 1. The desired tracking sequence can be defined by a concise syntax in which the relations, groups, and types of increments are specified.

The basic element in the syntax is the group specification. A group consists of a pointer data item for accessing the next relation row, and one or more target data items, to be transferred to the normalized relation. The data items' names are enclosed in brackets of different types, indicating the types of increments by which the relation rows are to be accessed. In a group the pointer data item name is separated from the target data item names by a colon.

Either pointer or target item names can be omitted. The entire group specification can be enclosed in brackets with following meaning: {} row increment; [] linked list; <> vector increment. Brackets can be nested. The following are examples of valid group and increment specifications:

$$\begin{aligned} &\{POINTER : TARGET\} \\ &[POINTER :] \\ &<: TARGET > \\ &\{<: TARGET >\} \end{aligned} \tag{8}$$

For a relation, group specifications can be concatenated. To indicate the relevant relation, groups are preceded by the relation name, as follows:

$$REL1 = \{POINTER : TARGET\} \tag{9}$$

When more than one relation is involved, specifications are strung together, as in the following example:

$$\begin{aligned} REL1 &= \{POINTER : TARGET\} \\ REL2 &= <: VECTOR > \end{aligned} \tag{10}$$

Here, POINTER points to a row of REL2. The output relation will contain two data items per row: namely TARGET, and an element of VECTOR.

An example of the programmable tree-traversing that is possible with RIMNET is found in a simple situation in which the rows of the unnormalized relation REL contain a vector attribute, VECTOR, and this relation must be normalized by assigning one vector element to each row of the output relation. The normalizing operation is then specified as follows:

$$REL = \{<: VECTOR >\} \tag{11}$$

where: VECTOR is the right side of the group specification, VECTOR is the target item, and an explicit pointer is not needed because the <> brackets imply a next vector element pointer, while the brackets imply a next row pointer. REL is the input relation for which VECTOR is defined as an attribute in the schema.

Execution of the syntax starts at the first row and causes each vector element to be transferred, proceeding then to the second vector of the second row, and so on.

Slightly more complex is an operation in which a variable number of elements must be retrieved, with the total number of elements to be transferred indicated by another data item TOTEL stored in the rows of the input relation. The syntax is then:

$$REL = \{TOTEL\} <: VECTOR(1 - TOTEL) > . \tag{12}$$

where 1-TOTEL represents a subscript of VECTOR, ranging from 1 through TOTEL. If, in addition, the elements may be further spread over more than one record, and the continuation record is indicated by an item NEXTREC, the syntax becomes:

$$REL = \{ : TOTEL \} [NEXTREC :] <: VECTOR(1 - TOTEL) > . \tag{13}$$

The square brackets indicate a linked list for which NEXTREC is the pointer item.

4.5 Relational Operators

Once the initial relations (I1) are normalized, relational operators can then be applied to the extent possible to convert them to the desired intermediate structure (I2). Thus, we are testing the concept that these operators can be an important set of higher level tools.

So far, the following RIM operators have been used frequently: JOIN, PROJECT with WHERE clause, INTERSECT, CHANGE with WHERE clause and DELETE DUPLICATES. Especially the INTERSECT command was found to be a powerful operator, with advantages over a JOIN in many instances.

4.6 Topological Tools

One of the objectives of the intermediate data structure is to serve as a way-station where improvements and quality checks can be made. Therefore, we have adopted the philosophy of weak input. This means that data deficient in topology can be accepted because the necessary fixes and enhancements will be made with our tools. One of the significant advantages of a relational information management system such as RIM is that the data structure can be inspected, partially copied, changed, and otherwise manipulated in any manner, all interactively. One therefore enjoys total access to the data.

In another sense "weak input" means that only certain basic relations need to be extracted from I1; others can be built from these initial relations. The fully defined topology of I2 can actually be created given a weak input of two tables. One table must contain the arcnumbers and vertex point coordinates (arcxy), while the other must contain the adjacent regions for each arc (regardless of order – a weak archdr).

If a very weak input data source that consists merely of a set of connected arcs or even arbitrary spaghetti is encountered, it should be possible to use a generalized vector processor which will be able to reduce the vector data to basic line segments and recreate the desired topology. However, to properly handle this case, attribute information must be available or the attribute information must be provided manually.

4.7 Conversion Experience to Date

The first system for which a data structure has been converted to the relational format is the Geographic Entry System (GES) of Electromagnetic Systems Laboratory's (ESL) Interactive Digital Image Manipulation System (IDIMS). From the relational format GES, data have been transferred to the ARC/INFO system of the Environmental Systems Research Institute (ESRI), to the Map Overlay and Statistical System (MOSS) of the Bureau of Land Management (BLM) and the U.S. Fish and Wildlife Service, and to the AGIS/GRAM system of Interactive Systems Corporation (ISC).

A second system for which data structures were converted is the Analytical Mapping System (AMS) used by the U.S. Fish and Wildlife Service. This system is one of the vector input systems to MOSS.

Subsequent systems for which data were transferred to the relational format were ICARAS (a raster-to-vector conversion system developed by D. Nichols of the Jet Propulsion Laboratory), as well as data from an Intergraph digitizing system (EDCIDS) being developed at the EROS Data Center,

and the Unified Cartographic Line Graph Encoding System (UCLGES) of the U.S. Geological Survey (used to structure digital line graph (DLG) data). Since ICARAS was modified and EDCIDS was developed at the EROS Data Center, both produce an output data structure suitable for loading directly into the relational data structure.

Tools such as RIMNET have been successfully used in at least two interfaces (GES and AMS). In the more polished interfaces, the various tools are tied together with the VAX/VMS DCL language. These interfaces include the interfaces with ARC/INFO, interfaces with raster data, and the loading of DLG-3 standard data into the relational data structure.

4.8 Concluding Remarks

The procedures developed during the study have become better refined as more data structures have been analyzed in detail. The intermediate relational data structure developed during the study has worked well. However, concurrently, the DLG-3 optional format emerged as a vector interchange data structure. One drawback of the DLG-3 optional format has been the rigid attribute handling which inhibits the storage of complex attribute data. Recent developments concerning an enhanced DLG data structure, such as the Federal Geographic Exchange Format (FGEF), should help alleviate attribute handling problems. In general, vector data structure transfer between analysis systems has recently become more mature throughout the vector processing community.

The strength of the relational vector data structure within a relational database management system for data structure conversion is the explicit data organization, ready and total access to the data, and the use of minimal input as a data source criterion, which provides control of the topology generated. This control has proved vital in adjusting to the idiosyncrasies of the topologies in the different vector data structures, since clean topology in one data structure has often proven not to be clean topology in others.

The simplicity and elegance of the relational approach is reflected in the simple relations of our intermediate spatial data structure, such that the spatial operations can take full advantage of the relational method.

The development of a good set of interfacing, diagnostic, and improvement tools will provide an excellent capability to tackle diverse interfacing problems.

5 References

- Breithart, Y. J., and Hartweg, L. R., 1984, RIM as an implementation tool for a distributed heterogeneous database: NASA Integrated Program for Aerospace - Vehicle Design Symposium II, Denver, Colorado.
- Callahan, George M., and Broome, Fredrick R., 1984, The joint development of a national 1:100,000-scale digital cartographic database, Proceedings of the American Congress of Surveying and Mapping, Washington, D.C.
- Codd, E. F., 1970, A relational model of data for large shared data banks: Communications of the ACM, v. 13, no. 6, p. 377-387.

- Codd, E. F., 1971, Relational completeness of data base sublanguages: R. Rustin (Ed.), Data Base Systems (Courant Computer Science Symposia VI), Prentice-Hall, Englewood Cliff, New Jersey.
- Cox, N. J., Aldred, B. K., and Rhind, D. W., 1980, A relational data base system and a proposal for a geographical data type: Geo-Processing, v. 1, no. 3, p. 217-229.
- Dijkstra, E.W., 1959, A note on two problems in connexion with graphs: Numerische Mathematik 1, pp 269-271.
- Fegeas, Robin, G., 1982, Modularization of digital cartographic data capture: Proceedings of the Fifth International Symposium on Computer Cartography, Arlington, Virginia, August 22-28, 1982.
- Garrett, M. T., and Foley, J. D., 1982, Graphics programming using a database system with dependency declarations: ACM Transactions on Graphics, v. 1 no. 2, p. 109-128.
- Go, A., Stonebraker M., and Williams, C., 1975, An approach to implementing a geo-data system: Memorandum No. ERL-M259, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, California.
- IBM, 1979, IBM interactive management and planning system under IMS/VS: User guide SB11-5220, IBM European Publications Center, Uithoorn, The Netherlands.
- Kent, W., 1983, A simple guide to five normal forms in relational database theory: Communications of the ACM, v. 26, no. 2. p. 120-125.
- Martin, J., 1977, Computer Data-Base Organization, O. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Nyerges, T., 1984, ed., Digital Cartographic Data Standards: Alternatives in Data Organization: Progress Report of the Workgroup on Data Organization of the National Committee for Digital Cartographic Data Standards, Report 4, January.
- Shapiro, L. G., and Haralick, R. M., 1980, A spatial data structure: Geo-Processing, v. 1, no. 3,
- Shapiro, L. G., 1980, Design of a spatial information system in map data processing: edited by H. Freeman and G. G. Pieroni, Academic Press.
- van Roessel, J.W., 1982, Short term plan for the continuation of spatial data handling development at EDC: EROS Data Center, memorandum.
- van Roessel, J.W. and E.A. Fosnight, 1984, A relational approach to vector data structure conversion, in: D. Marble, et al. (eds), Proceedings of the International Symposium on Spatial Data Handling, Zurich (Switzerland).

A System Descriptions

The purpose of system descriptions is to provide the reader with some familiarity with the most important systems used by the EROS Data Center. The goal is to describe the priority 1 systems in some depth, while the lesser priority systems are described with less detail.

The supporting documentation provided with each of the descriptions may not be the best or the most up-to-date for the system concerned, but it is available at the EROS Data Center.

A.1 ADS

ADS is the data entry module of the BLM Geographic Information System, ADS/MOSS/COS. It was developed specifically for entry of townships, ranges, sections, and their attributes.

ADS was designed to be the data entry module to an automated land record system. It was the first concept of what is now known as ALMRS (Automated Land and Minerals Record System). In its early development, ADS was linked to MOSS, and as the field became more aware of other agency applications of various geographic information systems for resource management, ADS/MOSS evolved as the BLM GIS. This evolution involved several enhancements to increase and streamline its capability for entering a variety of resource data types, and to decrease its emphasis on land records.

ADS consists of several modules performing separate tasks related to data input, including data capture, editing, output, data base maintenance, and links to the analysis software. The software has both public-domain and in-house components.

ADS uses several registration routines to accomodate a variety of source materials, including 7.5-minute and 15-minute quads, sources with geographic control in quad format, and diagrams with no control. Two methods are used to predict geographic coordinate values. One, used specifically with 7.5-minute and 15-minute quads, is a geometric fit in which the program assumes a conic projection. The second uses a least-squares method to predict coordinates. It allows the operator to input as many known control points as desired, breaks the map into patches with four known geographic corner coordinates, and then the corresponding table coordinates are used to derive coefficients for two first-order polynomials: one for predicting latitude and the other for predicting longitude. This method is more useful for small-scale source materials.

The editing module has more than 40 commands that allow interactive editing of lines, points, and attributes, and a variety of display commands to allow plotting, windowing or zooming, overlaying, and edgematching.

Formation of polygons is handled by a separate routine in ADS, called CLOSEPOLY. CLOSEPOLY identifies intersections in digitized line data, creates nodes, forms a file of the nodes and their joining arcs, and finally forms polygons, tagging each with its attribute or attributes. CLOSEPOLY can be performed interactively or in batch processing. This allows the operator to digitize the linework randomly and without regard to direction, intersections, or attributes (referred to as free form or spaghetti digitizing).

Users of the ADS system find the extensive editing capability of tremendous benefit for maintaining quality control. The commands are easy to use with sufficient on-line help documentation to allow the new user to use even the most obscure editing command. The method of entering areal data

by digitizing lines in the spaghetti method is extremely efficient in terms of human labor. A typical 7.5-minute quad with soils-type data with an average of 150 polygons can be digitized in 4 hours. Another 1/2 hour would be required to run the CLOSEPOLY macro. A landnet theme for a 7.5-minute quad can be digitized, encoded, and polygons formed in less than 1 hour. Users and project managers also appreciate the ease with which ADS allows updating of existing map files, and the variety of methods for labelling and tagging features.

Several weaknesses have been identified through intensive use of the system:

1. The registration routines are limited. It has been suggested that geometric fits for predicting coordinate values are not valid for many projections. The system also requires that all sources be in a quad format so that patching can be accomplished. This restriction eliminates many source materials with the most up-to-date information.
2. The ADS to MOSS conversion routine is limited. Data can be converted to MOSS format in the UTM projection only. This limits the user's capability in terms of output products.
3. ADS has no projection capability.
4. System and programming documentation are sometimes out-of-date, sketchy, or non-existent, making system updates, corrections, and enhancements difficult. There is also a lack of documentation regarding the limits of the system.
5. There is a limit of 999 polygons per map.
6. ADS does not handle multiple attributes.
7. Labels are sometimes misplaced.

A study was completed in cooperation with the BLM and the Fish and Wildlife Service to analyze the capabilities of the two data entry systems used by these agencies. Its goal was to identify areas in which the two systems were similar and dissimilar and make recommendations for the improvement of each system. It is the BLM's intention to enhance the the Bureau's system to reflect the best of both systems. The ADS/MOSS/COS system is also being investigated for certain tasks in cartography, such as compilation and negative plate production. Some specific plans are:

1. To improve the registration routines to handle nonformatted source materials.
2. To enhance its compilation capabilities. Users would like to see some capability for building data from existing data, such as aliquot parts from sections, and keyboard entry of coordinates.
3. To incorporate projection capability to make output more versatile.
4. To improve the availability of all types of documentation.

Reference:

1. Mogg, M., BLM's Automated Digitizing System (ADS), in Proceedings of the Workshop on Methods and Techniques for Digitizing Data: U.S. Geological Survey Open-File Report 85-307.

The primary contact for ADS is: Branch of Graphics Systems Development, BLM, Denver Service Center D-441, P.O. Box 25047 Denver, CO 80225.

A.2 AGIS/GRAM

AGIS/GRAM is a product of Interactive Systems Corporation (ISC), Littleton, Colorado. The system is in a developmental mode and only test projects have been conducted. No user community exists at this time.

AGIS is the acronym for Automated Geographic Information System. GRAM is the acronym for GBT Record Access Manager, and GBT stands for Generalized Balanced Ternary. The latter is a unique system for addressing two-dimensional space, and is a two-dimensional extension of the balanced ternary notation system which uses the "trits" -1,0,+1. This extension employs a pyramidal hexagonal structure in which hexagons are organized into aggregates, and aggregates are organized into higher level aggregates, and so on. Gibson and Lucas developed the GBT system and the accompanying algebra, in which points in two-dimensional space are located by GBT addresses. These are unsigned integer numbers, in which each digit represents an aggregate level. The use of GBT addresses in lieu of Cartesian coordinates allows for more efficient calculation of relative addresses and distances between features.

ISC has developed a spatial database manager (GRAM) based on these principles, and the AGIS system calls on this database manager. Other AGIS subsystems are ODM (Overlay Display Manager) and RIM (Relational Information Management System). The public domain version of RIM (version 5.0) is used as the attribute database management system.

While the ISC AGIS system was originally oriented toward the management of line type data, in September 1983, the EROS Data Center contracted with ISC for the development of a region-oriented AGIS. This system was installed at the EROS Data Center in July 1985. This version of AGIS has a query capability with a syntax that allows simultaneous attribute retrieval and spatial overlay. The system is highly interactive and uses a Megatek monochromatic CRT unit for the display of high-resolution maps.

An important AGIS concept is that of a virtual map base, without mapsheet boundaries. Because of the hierarchical structure of the system, map generalization and scale change operations can be readily performed. Because of the elegance of this structure in handling troublesome problems in processing large, multisource, multiscale databases, it was believed that this system would have potential for special applications at the EROS Data Center.

With AGIS/GRAM, most user interaction is performed using a graphics workstation, consisting of a graphics CRT and a pointing device. At the EROS Data Center, a Megatek 1024 x 1024 monochromatic display and a digitizer tablet are employed. Some other devices are now supported, including Sanders, Rastertek, and Evans-Sutherland displays. All input tasks (digitizing and editing), and graphics-oriented queries, are performed using the graphics workstation. Other analysis tasks can be performed using a VT100-like terminal.

Functional characteristics of AGIS/GRAM are:

1. Data Capture - A digitizing sub-menu may be selected from the menus available during an AGIS session. Input and editing operations will then be possible from a table digitizer connected to the host via a separately configured port. A menu pad is registered in the table coordinate space to enable function control from the digitizer. Alternatively it is possible to register small map segments to the tablet digitizer at the workstation. This enables the editing menus to be used as entry menus.

2. Map Editing - Editing of map data is accomplished at the workstation by display of the graphic data and graphic interaction on the tablet. Menu options allow for node and line removal, movement or, insertion. Data typing of the lines can also be manipulated to cause lines to be associated with map categories and data classes. This feature is useful for lines/polygons which must exist in multiple map layers in exact registry.

Map attribute data for symbols, lines and/or regions are entered interactively, unless the data are loaded from the Vector Data Interface. RIM format files interface to the user for attribute data display, entry, or editing.

3. Format Conversion - The only external format of data which is readily accepted into AGIS is the RIM format for Vector Data Interface.
4. Topological Structuring - Symbols (point data), lines, and regions (polygons) may be defined in AGIS. The storage methodology is designed around nodes stored in GRAM with linkages and pointers necessary to build lines and regions. Line and region verification programs will check the linkages for correctness and completeness as well as creation of isthmus connections for nested regions.
5. Map Projection Conversion - AGIS uses the USGS GCTP routines for projection of maps to any of 21 options.
6. Other Map Transformations - The system supports a user-defined projection option.
7. Map Joining - Maps may be entered in segments by creating the AGIS database definition of the coordinate system large enough to accommodate all segments, and by digitizing the segments in register with the database coordinate system. Any matching or manipulation of the map graphics at the edges have to be done interactively with the zoomed display and editing capabilities.
8. Selective Retrieval by Attribute - The query syntax allows any of the RIM attributes for a line or region to be specified and restricted in a RIM-like clause. The AGIS query also supports the map layer names, feature types, and data classes maintained in the data structure.
9. Statistics and Mensuration - Line distance, region area, and perimeter are supported.
10. Map Overlay - Query operations on map layer/data class combinations with operators AND, OR, and NOT may be employed to generate a display for hardcopy or to OUTPUT a temporary file for further query operations.
11. Map Modelling - The assignment of weights and mathematical modelling is not supported at this time.
12. Buffer Zone Generation - A PROXIMITY clause is available in the query syntax that allows buffer zones to be incorporated in analyses.
13. Map Display and Plotting - The display that is generated at the workstation may be built by selection of any combination of map categories or data classes that exist in the database or by query retrieval. The scale is controllable with coverage correspondingly dependent on the graphics capability of the workstation. Hardcopy to dot-matrix and pen-plotter devices are supported. GCTP coordinate transformations may be applied prior to hardcopy.

Supporting documentation:

1. AGIS, Computer Mapping: Interactive Systems Corporation, Littleton, Colorado.
2. Final Report on the Development of N-dimensional GBT Software Under Contract NAS5-26227, For the National Aeronautics and Space Administration, Goddard Space Flight Center, December 1980: by Laurie Gibson and Sandra Fitzgerald, Interactive Systems Corporation.
3. User Guide for the N-dimensional Data Base System, Prepared for the National Aeronautics and Space Administration, Goddard Space Flight Center, Greenbelt, Maryland, Contract NAS 5-24300, Task Assignment 19200, April 1982, Computer Sciences Corporation.
4. Vectorization of Raster Images Using Hierarchical Methods, by L. Gibson, and D. Lucas, Reprinted from Computer Graphics and Image Processing, v. 20, no. 1, September 1982, Academic Press Inc.
5. A Hierarchical Pattern Extraction System for Hexagonally Sampled Images, by Laurie D. Gibson and Charles Lenzmeier, Interactive Systems Corporation, October 1981, Final Report for Period 1 March 1981 - September 1981. Prepared for the Air Force Office of Scientific Research, Building 410, Bolling Air Force Base, D.C. 20332.
6. Introduction to GRAM, Interactive Systems Corporation, 5500 South Sycamore Street, Littleton, Colorado 80120.
7. Automated Geographic Information System (AGIS), Version 2.2A, Interactive Systems Corporation, 5500 South Sycamore Street, Littleton, 80120.
8. A System for Hierarchical Addressing in Euclidean Space, by Dean Lucas and Laurie Gibson.

A.3 AHDS

The Alaska Hydrologic Digitizing System (AHDS) was designed primarily for the digitizing of surface water features, hydrography, and lakes. It is currently in use by the Alaska BLM. The system provides input to MOSS.

A.4 ALMRS

ALMRS (Automated Land and Minerals Record System) is a system being designed by the BLM. The development of ALMRS will have three distinct stages, systems concepts, development, and implementation. It is currently in the systems concepts stage. The goals of the development and implementation stages follow.

ALMRS will manage both automated and manual data. Automated data will include land description, ownership, and use, which may be displayed in written or graphic form. Serial number log, stipulations, and regulation references (abstracts) will be displayed in written form. All of these are used in case processing, reporting, and management. The manual data (case files, survey plats and notes, Control Document Index, and reference information) are more efficiently left in their present formats.

Six automated BLM files have data that will be heavily used or updated by ALMRS. Geographic coordinates and resource data files may be displayed in written or graphic form, while the bond, surety, qualification, and master name files will be in written form.

Nine BLM and non-Bureau systems will share data with ALMRS. BLM's Geographic Information System and Public Land Survey System will graphically display land description, ownership and use, geographic coordinates, and other data. The Financial Management System, Mineral System 1 (MS-1), Solid Leasable Minerals System (SLMS), and the Simultaneous Oil and Gas Leasing System (SIMO), will be displayed in written form. All of these systems will be used in BLM case processing, reporting, and management. Whenever possible, common data entries will be made only once. The non-BLM systems, Bonus and Rental Accounting Support System (BRASS), Production Accounting and Auditing System (PAAS), and Auditing and Financial System (AFS) which involves financial management matters of the Minerals Management Service (MMS) will be displayed in written form.

Supporting documentation:

1. ALMRS Automated Land and Mineral Record System, ALMRS staff, ALMRS Project Office, BLM Service Center, Denver Federal Center.
2. ALMRS Prototype, ALMRS staff, ALMRS Project Office, BLM Service Center, Denver Federal Center.
3. Automated Land and Mineral Record System Graphic Requirements, ALMRS staff, ALMRS Project Office, BLM Service Center, Denver Federal Center, June 20, 1985.

A.5 AMS

In the fall of 1977, Autometric, Inc., under contract to the U.S. Fish and Wildlife Service, began the development of the Wetlands Analytical Mapping System (WAMS). The system was to be used for building a wetlands database for the National Wetlands Inventory Project, whose function was to locate and classify all wetlands in the United States. The final system was installed in Ft. Collins, Colo., in February 1979, where it is now used for the creation and maintenance of geographic databases. Since then, the system has been enhanced in follow-on efforts with the USFWS and U.S. Army Engineer Topographic Laboratory and is called the Analytical Mapping System.

Supporting documentation:

1. The Analytical Mapping System, System Manager's Guide (Data General Eclipse with AOS), Revision 2. Autometric, Inc. 5202 Leesburg Pike, Suite 1308/Skyline 1, Falls Church, Virginia 22041. Prepared by Harry Niedzwiadek, Bill Brooks, Dr. Carl Reed, Susan Codega, Jim Krammer, George Hoffman.

A.6 ARC/INFO

ARC/INFO is a vector-based GIS developed by the Environmental Systems Research Institute (ESRI), Redlands, Calif. It was developed to replace the Polygon Information Overlay System (PIOS) which ESRI used to process contracted project work and had installed for several clients. ARC/INFO was first released in 1982, and has since been installed at over 100 sites. It uses the INFO relational DBMS for managing attributes, and for all query syntax. INFO is produced and marketed by Henco, Inc., and is used for many non-GIS database management applications. ARC/INFO also makes use of the Interactive Graphics Language (IGL) from Tektronix, for controlling display and

plotter graphic output to most terminals and plotters. It has been installed on PRIME, VAX, Data General, and IBM computers and work is presently under way to develop a PC standalone workstation using PC-INFO and selected ARC/INFO functions for analysis and display.

ARC/INFO Version 2.2 was installed at the EROS Data Center in June 1984. In October 1984, Version 2.3 was installed, with some enhancements made to digitizer device drivers, and patches made to several minor bugs. Two-week training courses were held at the EROS Data Center in August 1984 and in May 1985, in which over 25 users were trained. Concurrent with the May 1985 training, a pre-Beta Version 3.1 was installed. It included substantial enhancements to ARC/INFO, including ARCEDIT, a digitizing and editing program, query enhancements to ARCPLOT, the primary display and map output subsystem, and several network analysis programs. An annual software maintenance fee paid to ESRI entitles the EROS Data Center to new software and documentation releases, and telephone consultation for site-specific software maintenance problems and user problems. Because of the diversity of our data sources and project work, we have used the user hotline frequently, and received help in solving problems and in interfacing data from several sources. The system has been used with remotely sensed cover classes, data derived from Digital Elevation Models (DEM), and several forms of digitized cartographic map information. Projects performed using ARC/INFO include:

- Acid Rain Building Material Inventory which intersected census tract data (STF3), with USGS land use (LUDA) and New York land use (LUNR) data sources. In this project, census tracts were assigned to cultural zones, using land use area within tracts calculated by ARC/INFO.
- Connecticut Hydrologic Information Project which resulted in a demonstration to the annual convention of the American Association of State Geologists, held in Mystic, Conn. For this project, the following data were input to ARC/INFO from USGS 1:24,000 DLG, 1:2,000,000 DLG, USGS Geographic Names Information System (GNIS), National Wetlands Inventory Data, and cover types extracted from Landsat multispectral scanner (MSS) data.
- Federal Mineral Lands Information System pilot studies in the Silver City 1:250,000 quadrangle (New Mexico and Arizona). In this data base, USGS 1:250,000 land use data and BLM 1:100,000 land status, ownership, and mineral rights data were input to ARC/INFO.
- John Day basin boundaries and drainage lines extracted from DEM. This project was a co-operative effort between USGS NMD and WRD and involved conversion of raster basin and lines to vector arcs, and inputting the arcs into the ARC/INFO system as Digital Line Graphs (DLG).

The design philosophy of ARC/INFO is to provide a tool-box of generic GIS functions for processing maps. This tool-box contains implements for creating, analyzing, displaying, and managing computerized maps in vector format. Functional characteristics of ARC/INFO are:

1. Data Capture - The DIGITIZE program allows maps to be captured and edited using one of several supported digitizers and display devices. This program can be used in a blind mode (with no graphic display), and allows for simple edits (deletion of arcs or vertices).
2. Map Edit - The ARCEDIT program allows for more sophisticated graphic editing than is possible using DIGITIZE, including command entry by tablet menu, node snapping, smooth arc generation by spline interpolation, and selectable line and annotation style, font, and color.

3. **Format Conversions** - Several vector formats are supported, including DLG-3 (standard and optional forms), GIMMS, DIME, ARC/INFO IMPORT/EXPORT, and the EROS Data Center RIM format (for vector data interface). Vector-to-raster and raster-to-vector conversions are supported for ESRI's Single Variable File (SVF) and Multi-Variable File (MVF) structures. The SVF and MVF files are created from the EROS Data Center's image processing and topographic analysis software.
4. **Topological Structuring** - The CLEAN and BUILD programs allow for weak input of lines and (optionally) area labels. The CLEAN program resolves line intersection and dangling arcs caused by overshoot in digitizing. The CREATELABELS program finds an interior label point (usually but not always the centroid) for areas with no labels attached in data capture.
5. **Map Projection Conversion** - ARC/INFO uses the USGS GCTP routines that allow systematic conversion to 20 projections.
6. **Other Map Transformations** - The TRANSFORM program takes user-specified points, and calculates an affine transformation by the method of least-squares fit. This procedure allows rotation, scale, and shift of points, lines, and areas.
7. **Map Joining** - Map joining or edge matching operations are performed using the EDGE-MATCH program which performs automatic node matching (snapping) to a user-specified tolerance, and optionally allows supervised adjustments to be made in joining and splitting arcs. The MAPJOIN program (or APPEND and CLEAN commands) is used to perform the map join operation.
8. **Selective Retrieval by Attribute** - Attribute operations can be performed within the INFO subsystem or within the ARCEDIT and ARCPLOT programs at display time. The INFO syntax is used consistently throughout the system and contains a complete set of boolean and arithmetic operations.
9. **Statistics and Mensuration** - Distance measurements can be performed within ARCEDIT and tabular sums can be calculated with INFO. INFO can optionally SAVE tabular data as ASCII files that could be input to user-supplied statistical subsystems (for example, SPSS, SAS, IMSL).
10. **Map Overlay** - Logical AND, OR, and NOT operations can be performed using one of eight available overlay functions.
11. **Map Modelling** - Modelling of attributes is performed using INFO and may or may not require an INFO program to be used to derive model weights. INFO programs are user written, are compiled and executed by INFO, and the results are used in subsequent map display and tabular reports.
12. **Buffer-Zone Generation** - Buffer zones can be generated using the BUFFER program and allow for a user-specified zone width applied to each point, or area, or optionally input as a table of distances to be applied by feature attributes or classes.
13. **Map Display and Plotting** - The ARCPLOT subsystem uses Tektronix IGL to display and plot map graphics. Query operations are supported using INFO syntax to selectively display by attributes. IGL contains a rich set of line types and character fonts, and the user can augment the set with user-defined types. Color graphics are fully supported for the 4100; series terminals and color pen plotters are supported if they are Calcomp-call-compatible, as most are.

Supporting documentation:

1. ARC/INFO Users Manual, DEC VAX, Version 3.1, Prepared by Environmental Systems Research Institute, 380 New York Street, Redlands, Calif. 92373, January 1985.
2. INFO-VAX, Revision 9, Reference Manual, Henco Software Inc, 100 Fifth Avenue, Waltham, MA 02154, October 1984.
3. INFO Primer, by Frances D. Brooks, 1980, Henco Inc.
4. Interactive Graphics Library (IGL), Users Manual, Tektronix Inc., P.O. Box 500, Beaverton, Oregon 97077, 1982.
5. ARC/INFO: A Geo-Relational Model for Spatial Information, by Scott Morehouse, ESRI, in Proceedings of Sixth International Symposium on Computer-Assisted Cartography (Auto-Carto 6), Ottawa, 1984.

A.7 COS

COS (Cartographic Output System) is the map-making system used in conjunction with MOSS. It was developed at the U.S. Fish and Wildlife Service, Western Energy and Landuse Team (WELUT), Ft. Collins, Colorado, by Randy Frosh, Jim Kramer, Pat Burcky, Eric Strand, and others.

COS allows for the entry and manipulation of cell, polygon, text, line, and point data, and produces maps and graphics in various forms. COS can use a variety of output display devices through the implementation of a device-independent graphics package.

COS is used by USFWS and BLM on Data General Equipment. BLM, ETL, and USFWS are interested in developing a more interactive version of COS.

Supporting documentation:

1. COS User's Manual, by Randy Frosh and Mike Walsh. Project Officer Eric Strand, Western Energy and land Use Team, 2627 Redwing Road, Fort Collins, Colorado 80526-2899. Prepared for Computer Systems Services Unit, Western Energy and Land Use Team, Division of Biological Services, Research and Development, Fish and Wildlife Service, U.S. Department of Interior, Washington D.C. 20240. (Revised September 1983).
2. Cartographic Output System, Systems Manual, By Patrick L. Burcky, Randy Frosh, and Jim Kramer. Project - 5189. Submitted to U.S. Department of the Interior Fish and Wildlife Service, Office of Biological Services, Fort Collins, Colorado, Attn: Eric Strand, Contract No: 14-16-0009-78-011. Submitted by HRB-Singer, Inc. Energy and Natural Resource Systems, State College, PA 16801.

A.8 CRAIGES

CRAIGES (Cartographic Reproduction and Interactive Graphics Editing System) is an Intergraph-based system used in conjunction with UCLGES for the production of DLG data and can be used to produce DLG-2 data directly on the Intergraph. The program was developed by Craig Skalet

at the Western Mapping Center of the USGS in 1981. The program is currently maintained at the Western Mapping Center and has been installed at the EROS Data Center.

Supporting documentation:

1. Craiges Manual, Programmed by Craig Skalet, Documented by Jay Fry, Tested by Richard Angel, Shirley Hokins, Pauline Jackson, Peg Rawson, Linda Scott, March, 1982.
2. Cartographic Reproduction and Interactive Graphics Editing System, (Craiges), Version 2.0, User's Manual, U.S. Geological Survey, National Mapping Division, 345 Middlefield Road, Menlo Park, CA 94025.
3. Cartographic Reproduction and Interactive Graphics Editing System, (CRAIGES), Version 3, Operator's Procedure Manual, U.S. Geological Survey, National Mapping Division, 345 Middlefield Road, Menlo Park, CA 94025.
4. Cartographic Reproduction and Interactive Graphics Editing System, (CRAIGES), Version 3, User's Manual, U.S. Geological Survey, National Mapping Division, 345 Middlefield Road, Menlo Park, CA. 94025.

A.9 DCASS

DCASS (Digital Cartographic Software System) was developed at the Midcontinent Mapping Center and was designed to produce compilation manuscripts, color separates, DLG-2 files, DEM-2 files and other statistical information from cartographic files.

DCASS was created to help complete the 1:24,000-scale map program using digital means to decrease the manual scribing of maps; it takes up to three scribing operations for each overlay to create a map. Raw data for each overlay comes from two sources: some from the Altek digitizer, and most from a stereo plotter.

A.10 DLG100K

This system was developed to aid in the production of 1:100,000 DLG's for the 1990 Census, and accepts SCITEX DIGIT files scanned from 1:100,000-scale maps. Topology is then defined from vectorized line elements. These lines, with node and area topology, are registered to the Universal Transverse Mercator (UTM) projection system. A quality check of the data is made to verify the topological structuring. The system outputs a Digital Line Graph 3 (DLG-3) machine-readable data set, and optional Versatec plots.

The system has been designed for batch job submittal and is operational on the Perkin-Elmer 3230 minicomputer, operating system version 6.2.

The system is used for production operations by all mapping centers of the National Mapping Division.

Supporting documentation:

1. DLG100K, Scitex-to-Network DLG, Software Documentation, Draft, Department of the Interior, U.S. Geological Survey, National Mapping Division.

A.11 GES

GES (Geographic Entry System) is the map digitizing system associated with ESL's IDIMS image processing system, and is currently used for most applications-oriented map digitizing at the EROS Data Center. Maps digitized through GES are collected in vector format. The system constructs the topology of the digitized features during the actual digitizing process, a feature not frequently found in other systems.

Once an area has been digitized, it can be converted to raster format using the IDIMS stratification utility function STRATA. The term stratification used by ESL in this context is probably a misnomer, and seems to be a hold-over from earlier applications of raster-based maps in sampling applications. More recently the ESRI function GRIPS is used for the vector to raster conversion. This function avoids some of the sliver problems associated with STRATA.

The current version of GES implemented at the EROS Data Center is version 8.6, which has a plotting capability for displaying the digitized maps.

GES has one main internal data structure that is primarily of interest. It makes abundant use of pointers, and allocates free file space using a free space list. Several utility programs access the main data structure. One such program is ALLCOORD, intended to convert coordinates from one system to another system. In the process of doing so however, it also produces a polygonal output data structure and is therefore sometimes used for this purpose.

Supporting documentation:

1. Geographic Entry System (GES) Manual, ESL Technical Memorandum ESL-TM1510, 25 Aug 1982.
2. Bruce Wicinas, GES Program Documentation (Proprietary Data) August 2, 1978 GES5C (marked "obviously out of date," but the only documentation available at ESL)
3. GES 7 File Format Diagram, dated 8/4/80.
4. HP-GES (Geographic Entry System), Manual - Version 8.6, ESL Incorporated, A Subsidiary of TRW Incorporated, Sunnyvale, Calif.

A.12 GIMMS

GIMMS is a user-oriented general-purpose geocartographic processing system for the analysis of geographic data through the generation of maps, graphs, and tabular information. It can also be used as a graphics package. GIMMS can deal with points, lines and areas, as well as with numeric and alphanumeric attribute data. The most-developed function in GIMMS is thematic mapping, but it has several other powerful capabilities, most notably the ability to convert from polygonal to gridded data, and to perform attribute modeling. It also has a polygon-creation and topological-checking capability. GIMSS can input points, arcs, or polygons.

GIMMS was primarily developed at the University of Edinburgh by Tom C. Waugh, and J. McCalden, and is now marketed by GIMMS Ltd.

Supporting documentation:

1. GIMMS Reference Manual, Release/Edition 4.5, T.C. Waugh, J. McCalden, Copyright GIMMS Ltd, 1983, 30 Keir Street, Edinburgh EH3 9EU, Scotland, U.K..

A.13 GIRAS

GIRAS (Geographic Information Retrieval and Analysis System) is a system for handling land use and land cover data, and has been specifically designed for the land use-land cover mapping program of the U.S. Geological Survey. This program provides systematic and comprehensive collection and analysis of land use and land cover map data at scales of 1:250,000 and 1:100,000. GIRAS's three basic subsystems are data input, data retrieval, and data output. It incorporates a database structure designed specifically for handling of spatial data, is based on the arc-node concept, and provides comprehensive facilities for data editing, data manipulation, and graphic and statistical output.

GIRAS is described in detail in USGS Professional Paper 1059. The usual mode of data input is that data are first compiled manually and then scanned by IOM-Towill in accordance with exact scanning specifications. The output consists of unlabelled digitized arcs and a set of polygon label points. The format is commonly referred to as the "contractor" format.

The GIRAS-archived format is binary format and can be converted through the function GRADLG to DLG binary, and from there through the function DLGBTA to DLG ASCII-optional. Another path is from GIRAS binary to GIRAS pure-polygon.

Supporting documentation:

1. GIRAS: A Geographic Information Retrieval and Analysis System for Handling Land Use and Land Cover Data, by William B. Mitchell, Stephen C. Guptill, K. Eric Anderson, Robin G. Fegeas, and Cheryl A. Hallam, U.S. Geological Survey Professional Paper 1059, for sale by the Branch of Distribution, U.S. Geological Survey, 1200 South Eads Street, Arlington, Va 22202.
2. Alpha Component Description, April, 1981.
3. Land Use and Land Cover Digital Data, by Robin G. Fegeas, Robert W. Claire, Stephan C. Guptill, K. Eric Anderson, and Cheryl A. Hallam. USGS Digital Cartographic Data Standards. U.S. Geological Survey Circular 895-E, 1983.

A.14 GPPU

GPPU is Intergraph's Graphic Polygon Processing Utility. This package makes it possible for the CAD/CAM Intergraph system to be used as a resource-oriented geographic information system. GPPU provides the capability to deal with complex polygon and region data.

A.15 ICARAS

Raster-to-vector processing at the EROS Data Center is performed in the LAS and IDIMS systems by algorithms obtained from the ICARAS system developed by Dave Nichols at the Jet Propulsion Laboratory in Pasadena.

The LAS implementation accepts a BYTE, INTEGER*2, or INTEGER*4 input image. Raster-to-vector processing is performed on the input image to generate a RIM data base with an ARCHDR, ARCXYS, and PRIMATT relation. This data base is considered to be weak input for full topological structuring.

Raster-to-vector processing in LAS employs five steps. The first three are modeled from the ICARAS system. The last two are added to the raster-to-vector process to load the specified RIM data base with the raster-to-vector processing results.

In the first step, the input image is scanned to locate unique regions. Each pixel of a region must have four-way (+) connectivity with the other pixels of that region, otherwise, the pixel is assigned to a new region. Each unique region is assigned a label number and an output image is generated with label numbers replacing the input pixel values. A region description file is built that links each region number with its old pixel value, and a count of the number of times the region number was assigned to pixels in the output image.

The second step scans the labeled image to locate the boundary segments of the labeled regions. A segment description file is built to store these boundary segments.

The third step reads the segment description file and the region description file. A cycling process is used to traverse the boundary segments for each region. A coordinate file, an arc header file, and an attribute file are generated.

The fourth step defines the data base schema and loads the relations that are defined through the cycling process. When the data base schema has been defined, the files built in step three are loaded to their respective relations.

The fifth step of the LAS implementation prepares the RIM relations for full topological structuring.

A.16 INTERGRAPH

In this document Intergraph refers to the EROS Data Center Intergraph Digitizing System (EDCIDS), because the system is referenced in this manner at the EROS Data Center.

The system was developed at the EROS Data Center after an analysis of several input systems including CRAIGES, Intergraph's GPPU, MIZAR Graphics' Natural Utility Package, Altek DLG Data Capture, and ESL's Geographic Entry System. The system has been designed to interface with the VDI system and relies on this system for topological structuring.

Supporting documentation:

1. EDC Intergraph Digitizing System (EDCIDS) Technical Review and Status Report, 8/8/84, by Dave Bankers and Diane Stoick.

A.17 LAS

LAS (Land Analysis System) is a raster-based image analysis system currently under development by the Goddard Space Flight Center and the EROS Data Center, under TAE (Transportable Applications Executive). LAS will be the replacement system for IDIMS and other image processing

systems at the Data Center (GIPS, RIPS, LAMS).

The raster-to-vector interface residing in LAS is from ICARAS.

Supporting documentation:

1. LAS Implementation Plan, Memorandum from Chief, Computer Services Branch (R. J. Thompson) to Senior Staff, February 14, 1985.
2. TAE and LAS related documentation, containing: Primer, User's Reference Manual, Programmer's Reference Manual, Utilities, Release Notes, TAE Newsletter, Catalog Manager User's Manual, Catalog Manager's Release Notes, LAS User's Guides, DMS, RCJM.
3. LAS Software Documentation Volume IX, A-J.
4. LAS Software Documentation Volume IX, K-Z.
5. Land Analysis System (LAS) Facility Guide, Revision 3, Prepared for LAS Project, Information Extraction Division, Goddard Space Flight Center, 6811 Kenilworth Avenue, Riverdale, Maryland 20737.

A.18 MOSS

MOSS (Map Overlay and Statistical System) is a vector-based GIS. Sometimes the acronym MOSS refers to the set of systems MOSS/MAPS/(W)AMS/COS in which MOSS is the vector analysis subsystem. In the following, however, MOSS refers only to the vector analysis subsystem unless otherwise noted.

MOSS design was begun in 1976 and was first implemented in 1977 on the Colorado State University CDC Cyber. USFWS-WELUT then perceived the utility of this system, and subsequently the main implementor of the system, Carl Reed, worked for FWS for a year to convert the system to Data General equipment. Thereafter he became employed by Autometric Inc. and this corporation still performs most of the system maintenance.

In the fall of 1977, under contract to the U.S. Fish and Wildlife Service, Autometric began the development of the Wetlands Analytical Mapping System (WAMS). The system was to be used for building a wetlands database for the National Wetlands Inventory Project, whose function it was to locate and classify all wetlands in the United States. The final system was installed in Ft. Collins, Colo., in February 1979, where it is now used for the creation and maintenance of geographic databases. Since then, the system has been enhanced in follow-on efforts with the USFWS and U.S. Army Engineer Topographic Laboratory and is now called the Analytical Mapping System (AMS).

A new Data General version of MOSS, completed for SCS and USFWS/WELUT in November 1984, has not yet been released.

The BLM/Denver obtained one of the earlier MOSS versions, and made significant improvements, thereby creating the BLM version. A number of different versions are still in use, although an attempt has been made to converge to a common version. For example, some BLM State offices are still using the BLM version in existence prior to the October 1983 "common" version developed by BLM and USFWS. Other offices and agencies are using the October common version as modified

through May 1984. Many users are also using raster overlay and neighborhood functions in MAPS to circumvent problems in MOSS.

In 1983, the EROS Data Center signed a contract with Autometric to convert MOSS to the VAX under VMS. This VAX version is based on the October common version, which is several releases behind the Data General version and does not include MAPS, AMS, or COS. This system is installed on the VAX at the EROS Data Center.

A MOSS User Group formed in 1982 holds annual meetings in Ft. Collins, Colo., at which MOSS users have made many suggestions for enhancements. In 1984, the EROS Data Center contracted with Autometric for a design study to investigate the feasibility of conversion to an arc-node data structure, and the incorporation of a relational DBMS. The functional specifications for this design have been completed and are under evaluation. In addition, the study has been forwarded to the Interior Department's Digital Cartographic Coordinating Committee (IDCCC) as well as to the MOSS Steering Committee.

The design philosophy of MOSS is to provide input, analysis, and output of cartographic data using existing public domain software. Other than the terminal control software for Plot-10 compatible devices (such as Tektronix 4014), which is supplied by each user site, this philosophy has been adhered to throughout its evolution. Functional characteristics of MOSS/MAPS/(W)AMS/COS are:

1. Data Capture - Data capture is performed by AMS, WAMS, or ADS (BLM only) programs. These programs support 4014 type terminals, and convert to polygon structured files which can be input to MOSS.
2. Map Edit - Map editing is performed by the capture programs. Arc-structured files must be completely edited prior to conversion to MOSS because editing is not performed after full polygons are formed.
3. Format Conversion - MOSS import/export and DLG-3 (optional) vector formats can be input to MOSS by converting to a MOSS ADD format. MOSS vector files can be converted to MOSS raster format, but not the reverse.
4. Topological Structuring - Polygon structuring is performed by the ADD function in MOSS. It is limited to 2,000 subjects for DG systems (1,000 for VAX), 16,000 features per map, 32,767 coordinates per polygon, and 350 islands per polygon.
5. Map Projection Conversion - The USGS GCTP routines are used to convert to one of 20 map projections using the PROJECT command.
6. Other Map Transformations - The TRANSLATE command allows for rotation, scale, or translation by angles, distances, scales, or by control-point matching.
7. Map Joining - Map joining is not supported in MOSS, but can be done manually at data capture time, provided file size limitations are met.
8. Selective Retrieval by Attribute - The BSEARCH command can be used to activate features which meet the criteria of a boolean expression. Once activated, the features can be displayed and listed.

9. Statistics and Mensuration - The STATISTICS program can be used to summarize mean and variance, and produce bar graphs. The SPSS program allows files to be output for subsequent input to a user supplied statistical package (SPSS).
10. Map Overlay - Map overlay is performed by the OVERLAY command. This overlay is the logical intersection (AND) of two polygon maps. The VAX version of MOSS is limited to 1,000 output subjects, and 30 characters maximum for each subject.
11. Map Modelling - Modelling of polygonal data is performed by converting to the raster domain. The raster program MODELG is used to perform boolean and arithmetic overlay operations.
12. Buffer Zone Generation - Buffer zones may be generated using the BUFFER command. Polygon islands are not processed, and the VAX version is limited to 1,000 points per feature processed.
13. Map Display and Plotting - MOSS has several commands for display of data as lines and shaded polygons using 4014 emulation (monochrome only). In addition, COS allows maps to be output on Calcomp compatible plot devices.

Supporting documentation:

1. MOSS User's Manual, Version 83.10, Editor John E. Lee, Technicolor Government Services, Inc. P.O. Box 9076, Fort Collins, CO 80526, Contract 14-16-0009-83-001, Prepared for Computer System Services Unit, Western Energy and Land Use Team, U.S. Fish and Wildlife Service, In cooperation with the Bureau of Land Management and U.S. Geological Survey, U.S. Department of the Interior, Washington D.C. 20240.
2. COS User's Manual, by Randy Frosh and Mike Walsh. Project Officer Eric Strand, Western Energy and Land Use Team, 2627 Redwing Road, Fort Collins, CO 80526-2899. Performed for Computer Systems Services Unit, Western Energy and Land Use Team, Division of Biological Services, Research and Development, Fish and Wildlife Service, U.S. Department of Interior, Washington D.C. 20240. (Revised September 1983).
3. Cartographic Output System, Systems Manual, By Patrick L. Burcky, Randy Frosh, and Jim Kramer. Project 5189. Submitted to U.S. Department of the Interior, Fish and Wildlife Service, Office of Biological Services, Fort Collins, Colorado, Attn: Eric Strand, Contract 14-16-0009-78-011. Submitted by HRB-Singer, Inc. Energy and Natural Resource Systems, State College, PA 16801.
4. The Analytical Mapping System, System Manager's Guide (Data General Eclipse with AOS), Revision 2. Autometric, Inc. 5202 Leesburg Pike, Suite 1308/Skyline 1, Falls Church, VA 22041. Prepared by Harry Niedzwiadek, Bill Brooks, Dr. Carl Reed, Susan Codega, Jim Krammer, and George Hoffman.
5. Feasibility and Design Study for the Enhancement of MOSS, Final Draft: Functional Specification. Prepared for the U.S. Geological Survey by Autometric, Inc., 2629 Redwing Road, Suite 340, Fort Collins, CO 80526, December 1984.

A.19 SDL/SGL

SDL/SGL (Surface Display Library/ Surface Gridding Library) is a surface-display system consisting of two sets of subroutine libraries, written and developed by Dynamic Graphics Incorporated.

ISM (Interactive Surface Modeling System), is an interactive system based on SDL/SGL, and also composed of the Faulting Surface Display Library, the Faulting Surface Gridding Library, and the Trend Surface Library.

Supporting documentation:

1. User Manual for the Surface Display Library, Prepared by The Software Development Group, Dynamic Graphics, Inc. 2855 Telegraph Ave, Berkeley, CA 94705.
2. User Manual for the Surface Gridding Library, Prepared by The Software Development Group, Dynamic Graphics, Inc. 2855 Telegraph Ave, Berkeley, CA 94705.
3. ISM Interactive Surface Modeling User's Guide, Dynamic Graphics, Inc. 2855 Telegraph Ave, Berkeley, CA 94705.

A.20 RIPS

RIPS (Remote Information Processing System) was developed at the EROS Data Center. The commercial version, fully compatible with the U.S. Government-furnished software, available for \$ 100, is now manufactured by Spectral Data Corp. Spectral Data also has its own software, including a parallelepiped classifier, that is a subset of the Government software. A variety of peripherals can be obtained with the RIPS system. Interfacing is facilitated through the use of an S-100 bus. Basic image data are stored on dual density 8-inch floppy disks, which can hold the equivalent of 20 7.5-minute U.S. Geological Survey quadrangles at Landsat MSS resolution. The U.S. Government software is quite diverse: in addition to a set of image processing functions, a number of GIS-type functions are available.

RIPS is included in this report because work is currently under way to develop a RIPS vector output interface.

Supporting documentation:

1. RIPS User Guide Applications Reference, Volumes I and II.
2. Hardware Guide to RIPS, Vol III.
3. Software Development Guide. Vol IV.

A.21 TIGER

The Bureau of the Census, U.S. Department of Commerce, has developed the Topologically Integrated Geographic Encoding and Referencing (TIGER) system, to support the geographic analysis and mapping needed for the 1990 Census. The term TIGER file refers to the computer file that contains all the data representing the position of roads, rivers, railroads, political and statistical

boundaries and other census-required features along with their attributes, such as name, address ranges, geographic code, and feature class.

The TIGER file structure is a series of interconnected files and programs accessible through a master control program. The system is made up of a number of subsystems which handle the major functions and currently resides on a Sperry Univac mainframe computer.

The spatial framework for the TIGER file is being provided through a cooperative program between the U.S. Geological Survey and the Bureau of the Census (Callahan, 1984). Each USGS 1:100,000-scale map is scanned and vectorized by USGS to produce four logical computer files: a road file, a hydrography file, a railroad file, and an other transportation file containing among other things, pipelines and powerlines. The Census Bureau is tagging the road file with the seven-character USGS attribute codes. USGS is tagging the other files with similar codes. Then the files are combined at USGS to form part of the National Digital Cartographic Data Base. From this combined, tagged data base, files of all transportation and hydrography are provided to the Census for use as the TIGER spatial framework.

The TIGER file structure is topologically structured and is similar to the DLG-3 structure, but additional information is carried by adding nontopologically structured elements in the file. Additional topological structure, such as pointers from nodes and areas to lines will also be included.

The TIGER system is an in-house Bureau of the Census system. The interfaces needed to interchange TIGER files with DLG files are being developed by the Geography Division of the Bureau of the Census, and the National Mapping Division of the USGS. These are not expected to be part of the Vector Data Interface process at the EROS Data Center.

Supporting documentation:

1. Broome, Frederick R., "TIGER Preliminary Design and Structure Overview: The Core of the Geographic Support System for 1990." Geography Division, U.S. Bureau of the Census. Prepared for Presentation at 1984 Annual Meeting of Association of American Geographers.
2. Callahan, George M. and Broome, Frederick R., "The Joint Development of a National 1:100,000-Scale Digital Cartographic Data Base," Proceedings of the American Congress on Surveying and Mapping, Washington, D.C., 1984.
3. Marx, Robert W. and Broome, Frederick, "A Topologically Based Data Structure for a Computer-Readable Map and Geographic System." in Proceedings of the Workshop on Methods and Techniques for Digitizing Data. U.S. Geological Survey Open-file Report 85-307, 1985, p. 59-85.

A.22 UCLGES

UCLGES (Unified Cartographic Line Graph Encoding System) is a system implemented by the USGS National Mapping Division under a national mapping program initiated in 1975, to collect and process Digital Cartographic Files (DCF's). Programs were initially implemented on an IBM 370/155, but have since been moved to an Amdahl 470 V7. Plans are being formulated to move all or portions of UCLGES to Gould 9780 minicomputers.

The data structures of interest are an internal on-line DCF format, and the input and output data structures termed DLG's (digital line graphs). The DLG's are assigned to either one of three classes:

DLG-1: Line map information that has been collected and coded to prescribed standards and edited to remove data acquisition blunders.

DLG-2: Line map information that has been interactively edited to add additional attribute codes and to remove visible errors and inconsistencies.

DLG-3: DLG-2 data that have been spatially structured to define all topological relationships.

The purpose of UCLGES is to process DLG-2's into DLG-3's through topological editing and assigning of topological relationships. UCLGES currently consists of 18 separate functions that range from file initialization to the integration of DLG's.

A major plan exists for the revision of UCLGES according to five phases:

1. Enlarge the data handling capacity (serious limitations on number of elements and map size exist currently).
2. Improve edge matching and joining of DLG's.
3. Provide capability to divide DLG's into two or more separate files.
4. Incorporate a new set of topological structuring functions (DLGCOP) in the PROSYS version of UCLGES. These include: (a) node generation, (b) area determination, (c) enhanced plotting, (d) topological error detection, (e) automatically generate point in area, and (f) command processor orientation of functions.
5. Display a DLG for validation and interactive editing.
6. Modify the plotting routines to accomodate larger file sizes.

The DLG data structures are available in an IBM binary format, and a 144-byte-record ASCII format. An 80-character ASCII format with expanded area definition is also available. Format documentation is available in USGS Circular 895.

The DLG-1,2,3 binary and 144-byte ASCII format have been described as one data structure; there is a separate description for the revised ASCII format.

Supporting documentation:

1. Untitled document, stamped DRAFT, with specifications for proposed UCLGES changes (author Joan Sziede ?), 5/10/83
2. Computer Files and Attribute Codes for Digital Line Graphs (January, 1980).
3. Joan Sziede, ASCII Representation of DLG 3 Format.
4. Digital Line Graphs from 1:2,000,000-Scale Maps, User Guide Interim Edition, National Mapping Program, USGEODATA, U.S. Department of the Interior, Geological Survey, National Cartographic Information Center (NCIC).

5. Untitled document, Attachment 1, contains DLG attribute codes and record formats.
6. NCIC User Guide, Digital Cartographic Data Files, Digital Line Graphs for 7.5- and 15-Minute Topographic Quadrangle Maps, Draft, Revised 11/16/81.
7. A Proposal for a Revised DLG ASCII Release Format, 9/29/82.
8. Appendix A.2, Optional DLG Record Formats, Draft.

A.23 VDI(RIM)

VDI (Vector Data Interface) is the system to which this document pertains. The word system, however, should be interpreted with caution, as it consists of a set of tools which work with the Relational Information Management System (RIM). The term system may be used because of this common database system and may be further justified because the tools, in conjunction with RIM, are developed according to the philosophy set out in this document. Furthermore, they are dedicated to a single purpose: vector data structure interfacing.

RIM was developed by NASA under contract NAS1-14700 with the Boeing Commercial Airplane Company. Principal developers were Wayne J. Erickson, Frederic P. Gray, and Geoffrey von Limbach. This system has progressed to version 5.0, and is in the public domain. Boeing subsequently began maintenance and improvements for a commercial version, and this version (BCS RIM) is currently available (6.0). The Boeing version includes a report writer and some graphic capabilities.

Different versions of RIM have been created by the University of Washington (UWRIM) and by the AMOCO company (Bfile).

Tools used in the VDI system can be assigned to two basic categories: (1) relational operators, and (2) independent programs. In the latter category there are five classes: (1) input, (2) general restructuring, (3) topological restructuring, (4) coordinate conversion, (5) display. The following table contains the names of the tools in these categories as well as a short functional description.

Table A1: Vector Data Interfacing Tools

Category 1 = input; 2 = general restructuring, 3 = topological restructuring, 4 = coordinate conversion, and 5 = display

TOOL	FUNCTIONAL DESCRIPTION	CATEGORY
RIMIN	Binary input to RIM	1
RIMSEQ	Add sequence number to relation	2
RIMDUP	Remove sequential duplicates in a relation	2
RIMTRF	Transfer a relation from one schema to another	2
RIMSCA	Multiply an attribute by a value	2
RIMAPD	Append one relation to another	2
RIMTRP	Transpose a relation	2
RIMRNG	Creates a range vector from two attribute values	2
RIMNET	Normalization, removal of linked lists, linking of separate relations	2
RIMTRM	Insert a termination symbol in a vector	2
RIMRDA	Removal of duplicate arcs	3
RIMCTSX	Topological restructuring of arc data	3
RIMXCK	Arc intersection checking	3
RIMPROJ	RIM interface to GCTP for coordinate conversions	4
RIMELX	Tektronix IGL compatible device previewing and editing	5
RIMQLT	Versatec plotting	5
RIMWDB	To and from World Data Bank I format	4

Supporting documentation:

1. User Guide: RIM 5.0, VAX VMS. Boeing Commercial Airplane Company, P.O. Box 3707, Seattle, Washington 98124.
2. BCS RIM Relational Information Management System, General Information Manual, Boeing Computer Services Company.
3. BCS RIM - Relational Information Management System, Version 6.0, Report Writer User Guide, Boeing Computer Services Company.
4. BCS RIM - Relational Information Management System Version 6.0, User Guide, Boeing Computer Services Company.
5. BCS RIM - Relational Information Management System Version 6.0, Host Dependent Supplement VAX VMS, Boeing Computer Services Company.
6. UWRIM Reference Manual, Instructions and examples for using UWRIM, the University of Washington version of the Relational Information Management System (RIM), by Wayne J. Erickson, University of Washington.

B Data Structures

The objective for the third phase of the workplan was to identify and concisely describe each of the vector data structures. The results of this phase are presented here.

Each data structure is introduced with a short description, followed by a detailed description for those data structures which are, or have been, of primary interest to the EROS Data Center.

Three methods of description are presented: (1) BNF; (2) tree structure, and (3) characteristics.

B.1 BNF Notation

BNF (Backus Naur Form) notation was developed to describe certain language grammars. Cox, Aldred and Rind (1980) have suggested the use of BNF to describe the syntax of geographical data types.

A BNF expression is called a production, and consists of a left side and a right side. The right side is separated from the left side by the symbol ::= which means "consists of". The right side contains one or more statements. Each statement may be terminal or non-terminal. Non-terminal statements are enclosed in pointed brackets, <>, while terminal statements are not. Each non-terminal statement may be further defined, in which case the character string within the pointed brackets is matched by a character string on the left side of another production.

Statements on the right side may be separated by either an exclamation mark, |, or an ampersand, &. The exclamation mark indicates that either the condition to the left or the right of the mark follows in the sequence of the data structure, while the ampersand indicates that both conditions exist simultaneously, although not in any specific order. For instance, a certain code may be either 1 or 2, indicated by 1|2, or two files may simultaneously exist, indicated by <file1> & <file2>. The absence of any demarcation indicates that the conditions follow in sequence. For instance, an x coordinate is followed by a y coordinate, is indicated by <x coordinate><y coordinate>. Repetition of a condition is indicated by enclosing the definition in curly brackets, { }, while optional conditions are enclosed in square brackets, []. Nested brackets of the same type are avoided by inserting intermediate definitions.

The BNF notation conventions used here deviate slightly from those commonly encountered in the literature. The exclamation mark is used instead of a vertical bar and the ampersand has been added to cope with non-sequential situations.

BNF definitions can be recursive, but for the purposes of this report most definitions will be hierarchical. One exception is the definition of BNF given in "bootstrap" BNF as shown in the following section, where "right side" is defined in terms of "right side". Possible confusion between the usage and definition of the BNF symbols is avoided by enclosing defined symbols in quote marks in this section.

In describing the data structures using the BNF notation the following rules were also applied:

1. Multiple conditions in repeat brackets are avoided. Instead, multiple situations are redefined as a single condition.
2. The exact number of repetitions for a condition is not a concern. The descriptions in the

following sections merely indicate that an item is repeated a number of times. At a later stage the notation may be modified to include repeat qualifiers.

3. Explanatory productions are avoided. An example of an explanatory production is:

```

elementcode ::= < linecode > < polygoncode >
      linecode ::= 1
      polygoncode ::= 2

```

Instead, a production of the following type is preferred:

```

elementcode ::= 1|2

```

4. Productions facilitating the recognition of spatial entities are emphasized, and so the following production

```

vectorfile ::= { < polygon > }
      polygon ::= < pointrecord > < attributerecord >

```

is preferred over:

```

vectorfile ::= { < recordset > }
      recordset ::= < pointrecord > < attributerecord >

```

B.1.1 BNF Defined in BNF

```

BNF ::= { < production > }
production ::= < left side > '::=' < right side >
left side ::= < character string >
right side ::= < definition >
right side ::= < right side > [< separator >] < definition >
separator ::= '|' | '&'
definition ::=
  < repeated condition > | < optional condition > | < condition >
repeated condition ::= '{' < condition > '}'
optional condition ::= '[' < repeated condition > ']'
optional condition ::= '[' < condition > ']'
condition ::= < terminal > | < non-terminal >
terminal ::= < character string >
non-terminal ::= '<' < character string > '>'

```

B.2 Tree Structures

Another type of description of a data structure is the tree structure. This description is derived directly from the BNF by processing the description through a program capable of matching right character strings with left character strings, in this instance the program PDL (Program Design Language). The result is a listing in which the tree structure of the data structure definition is made

more visible by adjusting the beginning of each definition to a column position corresponding to the tree level. Further breakdowns are then immediately printed offset and underneath the previous level. Thus, by scanning a certain column position, the definition of the data structure for that level can be read without being encumbered by further detail.

B.3 Characteristics Description

The BNF and tree structure methods of description deal primarily with the organization of the data structure. Certain characteristics may be immediately evident from these representations, while others cannot be expressed. For instance, the BNF and tree structure methods may show that a polygon consists of a list of vertices, each of which has an x and y coordinate, but they will not reveal whether the order of these vertices is clockwise or counter-clockwise, although the latter distinction may be vital to the correct processing of the polygon.

Therefore, in an attempt to cover all aspects of a data structure, a set of comprehensive characteristics of vector data structures has been compiled, of which, for each data structure the relevant sub-set has been extracted.

This set of characteristics has also been hierarchically structured with left and right definitions in a BNF-like manner. However the symbol `:` is used rather than `::=` to indicate that a certain characteristic or characteristic category has the following sub-characteristics. The following section shows the entire set of comprehensive characteristics.

B.3.1 Vector Data Structure Characteristics

```

characteristics: physical
                  logical
                  logical spatial (topology)
                  locational
                  attributal
physical: tape format
          disk format
          memory format
          calling sequence
tape format: number of files
            record type
            record length
            machine wordsize
number of files: single
                multiple
record type: ASCII
            binary
            EBCDIC
            mixed binary-ASCII
record length: fixed
                variable
machine wordsize: 16 bit

```


- 32 bit
 - other
- disk format:
 - number of files
 - access method
 - record length
 - record type
 - machine wordsize
- number of files:
 - single
 - multiple
- access method:
 - sequential
 - direct
- record length:
 - fixed
 - variable
- record type:
 - ASCII
 - binary
 - EBCDIC
 - mixed binary-ASCII
- machine wordsize:
 - 16 bit
 - 32 bit
 - other
- memory format:
 - wordsize
 - variable type
 - memory limitations
- wordsize:
 - 16 bit
 - 32 bit
 - other
- variable type:
 - byte
 - halfword
 - integer
 - real
 - double precision
 - mixed
- calling sequence:
 - wordsize
 - variable type
 - memory limitations
- logical:
 - number of logical files
 - file structure
 - record structure
- number of logical files:
 - single
 - multiple
- file structure:
 - sequential
 - hierarchical
 - network
 - relational
- sequential:
 - physical
 - linked list
- hierarchical:
 - binary tree
 - B-tree

- AVL tree
- GBT tree
- other
- record structure: single physical record/logical record
 - multiple physical records/logical record
- logical spatial: point
 - line
 - region
- point: dimensionality
 - representation
 - dimensionality: 2D
 - 3D
 - representation: true point
 - degenerate line
- line: spaghetti
 - non-intersecting
 - node snapped
 - networked
- region: basic entities
 - island handling
 - label points
- basic entities: polygon
 - arc-node
 - dime line segment
- polygon: boundary representation
 - boundary representation: point dictionary
 - line segment lists
 - location lists
- arc-node: element type groups
 - element type referencing
 - element type groups: nodes
 - arcs
 - regions
 - element type referencing: direct linkage
 - indirect reference
 - direct linkage: node-arc
 - node-region
 - arc-node
 - arc-region
 - region-node
 - region-arc
 - outer polygon-outer polygon
 - outer polygon-island
 - island-outer polygon
 - indirect reference: node-arc
 - node-region
 - arc-node
 - arc-region

- region-node
 - region-arc
- island handling: no islands
 - islands
- islands: connectors
 - polarity
 - hole-plug storage
 - hole-plug correspondence
 - outer polygon-islands coincidence
- connectors: artificial connectors
 - no artificial connectors
 - artificial connectors: flagged
 - not-flagged
- polarity: clockwise-counter clockwise polarity
 - no polarity
- hole-plug storage: single storage
 - double storage
- hole-plug correspondence: one to one
 - one to many
- outer polygon-island coincidence: hole
 - separate polygon
- label points: absent
 - one
 - multiple
- locational: reference system
 - encoding method
 - geo-block structure
 - resolution
 - tolerances
 - edge matching
- reference system: Generalized Balanced Ternary
 - Cartesian
- Cartesian: ground system
 - digitizing system
 - image system
- ground system: projection
 - measurement units
- projection: lat-lon
 - UTM
 - Albers
 - State-plane
 - Lambert
- measurement units: angular
 - metric
 - English
 - internal
- digitizing system: reference system
 - measurement units

- reference system: map
 - tablet
 - arbitrary
- measurement units: inches
 - user defined
- image system: reference system
 - measurement units
- reference system: line sample
 - other
- measurement units: pixels
 - other
- encoding method: absolute coordinates
 - relative coordinates
- relative coordinates: offset relative to origin geoblock
 - offset relative to spatial element
 - offset relative to previous point
- geo-block structure: number of geoblocks
 - geoblock size
 - geoblock origin
 - geoblock alignment
 - geoblock shape
- number of geoblocks: one (virtual map)
 - multiple
- geoblock size: spatially limited
 - filesize limited
 - unlimited
- geoblock origin: center
 - lower left corner
 - upper left corner
 - user definable
- geoblock alignment: north-south
 - with map projection
 - image alignment
- geoblock shape: rectangular
 - arbitrary
- resolution: fixed
 - element variable
 - geoblock-variable
 - sys-gen variable
- edge matching: edge matched
 - not edge matched
- edge matched: spatial elements transcend geoblock
 - spatial elements limited by geoblock
 - geoblock boundary in border elements
- attribute: attribute type
 - attribute structure
- attribute type: standard
 - user definable


```

attribute structure: interleaved
                    sparate file
interleaved: number of attribute records
              data types
              correpondence with spatial elements
              class groupings
number of attribute records: zero
                             one
                             multiple

data types: signed integer
            unsigned integer
            floating point
            character string
            vector
            matrix

correspondence with spatial elements: one to one
                                     one to many

class groupings: linked list
                 other pointer scheme

```


C Data Structure Descriptions

C.1 AGIS/GRAM

The internal data structure of this system is not well described in the AGIS/GRAM documentation. All data are designed to be entered either through the digitizing function or in the form of the interchange format.

The interchange format was specified by the EROS Data Center and is based on the early relational formats used for the VDI, but can easily be constructed from the currently used intermediate data structure.

C.1.1 AGIS/GRAM BNF

```
agis interchange format ::= <map interchange format database>
map interchange format database ::= <schema database>
                                   {<coordinate database>}
schema database ::= <coordinate database name relation>
                   <layers and classes relation>
coordinate database name relation ::= {<FGLODREL relation row>}
FGLODREL relation row ::= <directory specification>
                        <coordinate database name>
                        <user password>
                        <entity>
                        entity ::= <line> <region> <point>
layers and classes relation ::= {<LAYCLS relation row>}
LAYCLS relation row ::= <layer identification number>
                      <layer name>
                      <class identification number>
                      <class name>
coordinate database ::= <point coordinate database>&
                      <line coordinate database>&
                      <region coordinate database>
point coordinate database ::= <points coordinates relation>
                             <points attributes relation>
points coordinates relation ::= {<COORDS points relation row>}
COORDS points relation row ::= <nodenumber>
                              <latitude>
                              <longitude>
points attributes relation ::= {<ATTRIBTS points relation row>}
ATTRIBTS points relation row ::= <nodenumber>
                                <layer number>
                                <class identification number>
                                <symbol number>
                                <subject text string>
line coordinate database ::= <line coordinates relation>
                           <line attributes relation>
```



```

line coordinates relation::= {<COORDS line relation row>}
  COORDS line relation row::= <arc number>
    <latitude>
    <longitude>
line attributes relation::= {<ATTRIBTS lines relation row>}
  ATTRIBTS lines relation row::= <arc number>
    <layer number>
    <class identification number>
    <subject text string>
region coordinate database::= <region coordinates relation>
  <region attributes relation>
  <archheader relation>
  <area arc relation>
region coordinates relation::= {<COORDS regions relation row>}
  COORDS region relation row::= <arc number>
    <latitude>
    <lonitude>
region attributes relation::= {<ATTRIBTS regions relation row>}
  ATTRIBTS regions relation row::= <area number>
    <layer number>
    <class identification number>
    <subject text string>
archheader relation::= {<ARCHDR relation row>}
  ARCHDR relation row::= <arc number>
    <left area>
    <right area>
area arc relation::= {<AREAARC relation row>}
  AREAARC relation row::= <area number>
    <island indicator>
    <arc number>

```

C.2 AMS

C.2.1 AMS BNF

```

ams::= <normal/temporary node file (A file)>&
  <segment header file (B file)>&
  <edge node file (C file)>&
  <polygon file (D file)>&
  <segment coordinate file (E file)>&
  <classification file (F file)>&
  <multiple attribute file (G file)>
normal/temporary node file (A file)::= {<node file record>}
node file record::=
  <number of node subrecords>
  <record number of previous record with nodes in area>
  {<node subrecord>}

```



```

        <record number of following record with nodes in area>
        <filler>
node subrecord ::= <relative latitude of node>
                  <relative longitude of node>
                  <number of associated segments in extent>
                  <segment header record or extent pointer>
                  {<segment header record number>}
segment header file (B file) ::= {<segment header record>}
segment header record ::= <segment status>
                        <first coordinate record pointer>
                        <last segment indicator>
                        {<node record pointer>}
                        <segment attributes>
                        <segment polygon pointers>
                        <segment box>
                        <scale factor>
                        <last coordinate record pointer>
                        <triplet indicator>
                        <first scheme prefix>
                        <number of schemes>
                        {<additional scheme>}
                        <picture ID for graphics superposition>
segment attributes ::= <left><center><right>
segment polygon pointers ::= <left pointer>
                            <center pointer>
                            <right pointer>
segment box ::= <min latitude><max latitude>
               <min longitude><max longitude>
additional scheme ::= <prefix additional scheme>
                    <segment attributes>
edge node file (C file) ::= {<edge node record>}
edge node record ::= <edge number>
                   <record number of preceding record>
                   <record number of following record>
                   <number of edge node subrecords>
                   <edge node subrecord>
edge node subrecord ::= <latitude>|<longitude>
                      <new old node indicator>
                      <created in adjacent geounit flag>
polygon file (D file) ::= {<polygon record>}
polygon record ::= <polygon number>|<island polygon pointer>
                 <polygon attribute>
                 <polygon box>
                 <area>
                 <attribute display status word>
                 {<attribute location>}
                 <single multiple attribute index>
                 {<segment header record number>}

```



```

        <zero sequence>
polygon box::= <max latitude><max longitude>
               <min latitude><min longitude>
attribute display status word::=<left byte><right byte>
  left byte::= <lettering scale x 100>
               <attribute display code>
               attribute display code::= 0|1|2|3|4
attribute location::= <attribute point>|<centroid>|
                     <end point of lead line>
single multiple attribute index::= 0|1
zero sequence::= <end of polygon record zero>|
                 <end of polygon squence zeros>|
                 <next island pointer>
segment coordinate file (E file)::= {<coordinate record>}
coordinate record::= <number of coordinate sets in record>
                    <curve mode flag>
                    <record box>
                    {<coordinate set>}
record box::= <min latitude><max latitude>
              <min longitude><max longitude>
classification file (F file)::= <scheme record>
scheme record::= {<scheme description>}
scheme description::= <scheme name>
                    <scheme type>
multiple attribute file (G file)::= {<attribute record>}
attribute record::= {<16 character polygon attribute>}

```

C.2.2 AMS Tree Structure

```

ams
  <normal/temporary node file (A file)>
    {<node file record>}
    <number of node subrecords>
    <record number of previous record with nodes in area>
    {<node subrecord>}
    <relative latitude of node>
    <relative longitude of node>
    <number of associated segments in extent>
    <segment header record or extent pointer>
    {<segment header record number>}
    <record number of following record with nodes in area>
    <filler>
  <segment header file (B file)>
    {<segment header record>}
    <segment status>
    <first coordinate record pointer>
    <last segment indicator>

```



```

    {<node record pointer>}
    <segment attributes>
        <left>
        <center>
        <right>
    <segment polygon pointers>
        <left pointer>
        <center pointer>
        <right pointer>
    <segment box>
        <min latitude>
        <max latitude>
        <min longitude>
        <max longitude>
    <scale factor>
    <last coordinate record pointer>
    <triplet indicator>
    <first scheme prefix>
    <number of schemes>
    {<additional scheme>}
        <prefix additional scheme>
        <segment attributes>
        <picture ID for graphics superposition>
<edge node file (C file)>
    {<edge node record>}
        <edgenumber>
        <record number of preceding record>
        <record number of following record>
        <number of edge node subrecords>
        {<edge node subrecord>}
            <latitude or longitude>
            <new old node indicator>
            <created in adjacentgeounit flag>
<polygon file (D file)>
    {<polygon record>}
        <polygon number>
        <island polygon pointer>
        <polygon attribute>
        <polygon box>
            <max latitude>
            <max longitude>
            <min latitude>
            <min longitude>
        <area>
        <attribute display status word>
            <left byte>
                <lettering scale x 100>
                <attribute display code>

```



```

        tion>}
    <single multiple attribute index>
        0!1
    {<segment header record number>}
    <zero sequence>
        <end of polygon record zero>!
        <end of polygon sequence zeros>!
        <next island pointer>
<segment coordinate file (E file)>
    {<coordinate record>}
        <number of coordinate sets in record>
        <curve mode flag>
        <record box>
            <min latitude>
            <max latitude>
            <min longitude>
            <max longitude>
        {<coordinate set>}
<classification file (F file)>
    <scheme record>
        {<scheme description>}
            <scheme name>
            <scheme type>
<multiple attribute file (G file)>
    {<attribute record>}
        {<16 character polygon attribute>}
<attribute location>
    <attribute point>
    <centroid>
    <end point of lead line>

```

C.3 ARC/INFO

C.3.1 ARC/INFO BNF

```

arc/info database ::= {<coverage sub-directory>} & <INFO sub-directory>
coverage sub-directory ::=
    [<AAT - arc attribute file>] &
    [<ALT - list of arcs at each node>] &
    [<ARC - arc coordinates and topology>] &
    [<ARF - arc cross reference file>] &
    <BND - coordinate extremes of coverage> &
    [<CCT - index of label points by label grid>] &
    [<CNT - list of labels by polygon>] &
    [<LAB - label point coordinates and topology>] &
    [<NCT - index of nodes by index grid>] &
    [<PAL - list of arcs by polygon>] &

```



```

        [<PAT - polygon attribute file>] &
        [<PRF - polygon cross reference file>] &
        <TIC - tic coordinates and ids> &
        [<XAT - point attribute file>]
AAT - arc attribute file::= {<AAT record>}
  AAT record: <from node>
    <to node>
    <left polygon>
    <right polygon>
    <length of arc>
    <arc record number>
    <arc identifier>
ALT - list of arcs at each node::= {<ALT record>}
  ALT record::= {<indexed arc-id grup>}
    indexed arc-id group::= {<arc-id>}
ARC - ARC coordinates and topology::= {<ARC record>}
  ARC record::= <record identifier>
    <arc name>
    <from node>
    <to node>
    <left polygon>
    <right polygon>
    <number of points>
    {<arc vertex>}
    arc vertex::= <x coordinate> <y coordinate>
ARF - arc cross-reference file::= <AAT - arc attribute file>
BND - coordinate extremes of coverage::= <BND record>
  BND record::= <minimum x coordinate>
    <minimum y coordinate>
    <maximum x coordinate>
    <maximum y coordinate>
CCT - index of label points by label grid::= {<CCT record>}
  CCT record::= {<grid cell list>}
    grid cell list::= {<label point>}
      label point::= <x coordinate> <y coordinate>
CNT - list of labels by polygon::= {<CNT record>}
  CNT record::= <geometric centroid of polygon>
    <number of label points>
    {<label point>}
    geometric centroid of polygon::= < x coordinate> <y coordinate>
    label point::= <x coordinate> <y coordinate>
LAB - label point coordinates and topology::= {<LAB record>}
  LAB record::= <polygon name>
    <polygon identifier (surrounding polygon id)>
    <interior point>
    <label box>
    interior point::= <x coordinate> <y coordinate>
    label box::= <minimum x> <minimum y>

```



```

        <maximum x> <maximum y>
NCT - index of nodes by index grid::= {<NCT record>}
NCT record::= {<index square linked list record>}
    index square linked list record::= {<node entry>}
        node entry::= <node number> <x coordinate> <y coordinate>
PAL - list of arcs by polygon::= {<PAL record>}
PAL record::= <polygon range>
    <number of arcs>
    <arcs, nodes and polygons around polygon>
    arcs,nodes and polygons around polygon::= {<entry>}
        entry::=<arc> <node> <polygon>
PAT - polygon attribute file::= {<PAT record>}
PAT record::= <area>
    <perimeter>
    <polygon id>
    <input polygon id>
    <overlay polygon id>
    {<additional user defined item>}
PRF - polygon cross reference file::= {<PRF record>}
PRF record::= <area>
    <perimeter>
    <polygon id>
    <input polygon id>
    <overlay polygon id>
TIC - tic coordinates and ids::={<TIC record>}
TIC record::= <tic identifier>
    <tic x coordinate>
    <tic y coordinate>
XAT - point attribute file::= undefined
info sub-directory::= <info directory (ARCD8)>
    {<INFO file>}

```

C.4 DLG

C.4.1 DLG BNF

```

DLG::= <header records> {<overlay>} [<accuracy record>]
header records::= <identification record (type A)>
    <transformation record (type B)>
    <overlays description record (type C)>
identification record (type A)::= <unit name> <level code>
    <ground reference system code>
    <ground reference system zone code>
    <map projection parameters>
    <ground units of measure code>
    <resolution> <accuracy code>
    <number of sides in coverage polygon>

```



```

        [<filler>]
        {<coverage vertex point>} [<filler>]
unit name::= <name of digital cartographic unit>
        <date of original source material>
        <latest photorevision date>
        <scale of source material>
        <unused characters>
level code::= 1|2|3
ground reference system code::= 0|1|2|<3-20>
ground reference system zone code::= 9999|
        <state plane code>|
        <UTM code>
map projection parameters::= <semi-major axis of ellipsoid>
        <eccentricity squared of ellipsoid>
        <latitude first standard parallel>
        <latitude second standard parallel>
        <longitude of central meridian>
        <latitude of projected origin>
        <false easting> <false northing>
        {<unused parameter>}
ground units of measure code::= 0|1|2|3|4
accuracy code::= 0|<not zero>
number of sides in coverage polygon::= 4
coverage vertex point::= <longitude> <latitude>
transformation record (type B)::=
        {<file-to-map transformation parameter>}
        <number of registration points>
        [<filler>] {<registration point>}
        [<filler>]
registration point::= <identification> <x coordinate>
        <y coordinate>
identification::= SW|NW|NE|SE
overlays description record (type C)::=
        <number of overlays> [<filler>]
        {<contents array>} [<filler>]
contents array::= <network name> <maximum number of nodes>
        <actual number of nodes>
        <maximum number of areas>
        <actual number of areas>
        <maximum number of line segments>
        <actual number of line segments>
overlay::= <DLG3 overlay>|<DLG1,2 overlay>
DLG3 overlay::= {<node >} {<area >} {<line >}
node::= <information record (type D.1)>
        [<attribute record (type F)>]
        [<text record (type G)>]
area ::= <information record (type D.1)>
        [<attribute record (type F)>]

```



```

        [<text record (type G)>]
line ::= <topology record (type D.2)>
        <coordinate record (type E)>
        [<attribute record (type F)>]
        [<text record (type G)>]
information record (type D.1) ::= <element type code>
        <element internal identification number>
        <representative point>
        <number of attributes>
        <number of pairs of text characters>
        [<filler>]
        element type code ::= N|A
attribute record (type F) ::= {<classification codes>}
        [<filler>]
        classification codes ::= <major code> <minor code>
text record (type G) ::= {<character pair>}
        character pair ::= <character> <character>
        character ::= <non-sentence delimiter>|
        <sentence delimiter>
        sentence delimiter ::= /
topology record (type D.2) ::=
        <line segment graph element code>
        <element internal identification number>
        <internal id number of starting node>
        <internal id number of ending node>
        <internal id number of left area>
        <internal id number of right area>
        <number of vertex points in line segment>
        <number of attributes>
        <number of pairs of text characters>
        [<filler>]
coordinate record (type E) ::= {<vertex point>}
        [<filler>]
        vertex point ::= <x coordinate> <y coordinate>
DLG1,2 overlay ::= {<DLG1,2 line>}
DLG1,2 line ::= <DLG1,2 information record (type D.3)>
        <coordinate record (type E)>
        [<attribute record (type F)>]
        [<text record (type G)>]
DLG1,2 information record (type D.3) ::=
        <number of vertex points n line segment>
        <number of attributes>
        <number of pairs of text characters>
accuracy record (type H) ::= <statistics availability switch>
        <RMSE of file datum relative to absolute>
        <sample size>
        <statistics availability switch>
        <RMSE of DLG relative to files datum>

```


<sample size>

C.4.2 DLG Tree Structure

DLG

```
<header records>
  <identification record (type A)>
    <unit name>
      <name of digital cartographic unit>
      <date of original source material>
      <latest photorevision date>
      <scale of source material>
      <unused characters>
    <level code>
    <ground reference system code>
      0 | 1 | 2
    <ground reference system zone code>
      9999 | <state plane code> | <UTM code>
    <map projection parameters>
      <semi major axis of ellipsoid>
      <eccentricity squared of ellipsoid>
      <latitude first standard parallel>
      <latitude second standard parallel>
      <longitude of central meridian>
      <latitude of projected origin>
      <>false easting>
      <>false northing>
      {<unused parameter>}
    <ground units of measure code>
      0 | 1 | 2 | 3 | 4
    <resolution>
    <accuracy code>
      0 | <not zero>
    <number of sides in coverage polygon>
      4
    [<filler>]
    {<coverage vertex point>}
      <lngitude>
      <latitude>
  <transformation record (type B)>
    {<file to map transformation parameter>}
    <number of registration points>
    [<filler>]
    {<registration point>}
      <identification>
        SW | NW | NE | SE
      <x coordinate>
```



```

        <y coordinate>
    <overlays description record (type C)>
        <number of overlays>
        [<filler>]
        {<contents array>}
            <network name>
            <maximum number of nodes>
            <actual number of nodes>
            <maximum number of areas>
            <actual number of areas>
            <maximum number of line segments>
            <actual number of line segments>
    {<overlay>}
    <DLG3 overlay>
        {<node>}
            <information record (type D.1)>
                <element type code>
                <element internal identification number>
                <representative point>
                <number of attributes>
                <number of pairs of text characters>
                [<filler>]
            [<attribute record (type F)>]
                {<classification codes>}
                    <major code>
                    <minor code>
                [<filler>]
            [<ext record (type G)>]
                {<character pair>}
                    <character>
                        <non sentence delimiter>
                        <sentence delimiter>
        {<area>}
            <information record (type D.1)>
                <element type code>
                <element internal identification number>
                <representative point>
                <number of attributes>
                <number of pairs of text characters>
                [<filler>]
            [<attribute record (type F)>]
                {<classification codes>}
                    <major code>
                    <minor code>
                [<filler>]
            [<text record (type G)>]
                {<character pair>}
                    <character>

```



```

        <non sentence delimiter
        <sentence delimiter>
    {<line>}
    <topology record (type D.2)>
        <line segment graph element code>
        <element internal identification number>
        <internal id number of starting node>
        <internal id number of ending node>
        <internal id number of left area>
        <internal id number of right area>
        <number of vertex points in line segment>
        <number of attributes>
        <number of pairs of text characters>
        [<filler>]
    <coordinate record (type E)>
        {<vertex point>}
            <x coordinate>
            <y coordinate>
        [<filler>]
    <attribute record (type F)>
        {<classification codes>}
            <major code>
            <minor code>
        [<filler>]
    <text record (type G)>
        {<character pair>}
            <character>
                <non sentence delimiter>
                <sentence delimiter>
<DLG1,2 overlay>
    {<DLG1,2 line>}
        <DLG1,2 information record (type D.3)>
            <number of vertex points in line segment>
            <number of attributes>
            <number of pairs of text characters>
        <coordinate record (type E)>
            {<vertex point>}
                <x coordinate>
                <y coordinate>
            [<filler>]
        <attribute record (type F)>
            {<classification codes>}
                <major code>
                <minor code>
            [<filler>]
        <text record (type G)>
            {<character pair>}
                <character>

```



```

        <non sentence delimiter>
        <sentence delimiter>
    [<accuracy record (type H)>]
        <statistics availability switch>
        <RMSE of file datum relative to absolute>
        <sample size>
        <RMSE of DLG relative to files datum>

```

C.4.3 DLG Characteristics

```

characteristics: physical
                  logical
                  logical spatial (topology)
                  locational
                  attribute
physical: tape format
    tape format: number of files
                  record type
                  record length
                  machine wordsize
    number of files: multiple
    record type: ASCII
                  binary
    record length: fixed (144-byte ASCII)
                  variable (IBM binary)
    machine wordsize: 32 bit
logical: number of logical files
          file structure
          record structure
    number of logical files: multiple
    file structure: sequential
    sequential: physical
    record structure:
        single physical record/logical record (binary)
        multiple physical records/logical record (ASCII)
logical spatial: point
                  line
                  region
point: dimensionality
      representation
      dimensionality: 2D
      representation: degenerate line
line: non-intersecting (DLG-1,2)
      node snapped (DLG-3)
region: basic entities
        label points
        basic entities: arc-node

```


- arc-node: element type groups
 - element type referencing
- element type groups: nodes
 - arcs
 - regions
- element type referencing: direct linkage
 - indirect reference
- direct linkage: arc-node
 - arc-region
- indirect reference: node-arc
 - node-region
 - region-node
 - region-arc
- island handling: no islands
- label points: one (may be outside region)
- locational: reference system
 - encoding method
 - geo-block structure
 - resolution
 - tolerances
 - edge matching
- reference system: Cartesian
 - Cartesian: ground system
 - ground system: projection
 - measurement units
 - projection: Albers equal area (1:2,000,000)
 - UTM (quad maps)
 - measurement units: English (thousands of an inch)
- encoding method: relative coordinates
 - relative coordinates: offset relative to origin geoblock
- geo-block structure: number of geoblocks
 - geoblock size
 - geoblock origin
 - geoblock alignment
 - geoblock shape
- number of geoblocks: multiple (1:2,000,000 maps, 7.5', 15')
- geoblock size: spatially limited
 - filesize limited
- geoblock origin: center
- geoblock alignment: with map projection
- geoblock shape: rectangular
- resolution: fixed
- edge matching: not edge matched
- attribute: attribute type
 - attribute structure
- attribute type: standard
- attribute structure: interleaved
 - interleaved: number of attribute records

data types
 correspondence with spatial elements
 number of attribute records: one
 data types: signed integer
 character string
 correspondence with spatial elements: one to one

C.5 FGEF

C.5.1 FGEF BNF

Acronyms in the following BNF statements are to be interpreted as:

UD - subfield delimiter
 FD - field delimiter
 RD - record delimiter
 SD - section delimiter
 BL - begin level symbol
 EL - end level symbol
 IS - include symbol

```

FGEF ::= {<file + EOF>} <EOF>
file + EOF ::= <file> <EOF>
file ::= <delimiter section> <SD>
        <general information section> <SD>
        <geographic reference section>
        [<data section>]
        <SD>
delimiter section ::= <section identification record> <RD>
                     <delimiter record>
section identification record ::= D <UD> DELIMITER
delimiter record ::= <UD> <FD> <RD> <SD> <BL>
                   <EL> <IS> <IS> <IS>
general information section ::=
        <section identification record> <RD>
        <file description record>
        {<RD + comment record>}
section identification record ::= I <UD> GENERAL INFORMATION
file description record ::=
        FEDERAL GEOGRAPHIC EXCHANGE FILE <FD>
        <version, date or code> <FD>
        <file name> <FD>
        <documentation reference> <FD>
        <format details> <FD>
        <attribute data type codes> <FD>
        <include symbol use flag> <FD>
        <number of data sections> <FD>
  
```



```

        {<data section description>} <RD>
attribute data type codes::= {<format character>}
    format character::= X|H|I|R|E|T
include symbol use flag::= YES, INCLUDE SYMBOL USED|
                        NO, INCLUDE SYMBOL NOT USED|
                        YES|NO|Y|N|T|F
data section description::=
    <section code> <UD>
    <data type code> <UD>
    <attribute data type codes> <UD>
    <number of records>
    data type code::= G|T|A|N|S|W|E|L|P|C|R|I|J|K|M
RD + comment record::= <RD> <comment record>
comment record::= [<record id + FD>]
    <contents>
    record id + FD::= IQ1|IQ2|IQ3|IQ4|IQ5
    <FD>
geographic reference section::=
    <data coordinate definition record>
    [<RD + map projection record>]
    [<RD + coverage record>]
    {[<RD + registration point record>]}
data coordinate definition record::=
    <entry id> <FD>
    <data coordinate type> <FD>
    <data coordinate format> <FD>
    [<spatial address type>] <FD>
    [<horizontal datum>] <FD>
    <vertical datum> <FD>
    <horizontal coordinate units> <FD>
    <vertical coordinate units> <FD>
    <estimate of coordinate accuracy> <FD>
    <transformation parameters>
    [<FD + coordinate resolution>]
    [<FD + map scale denominator>]
    [<D+ number of registration points>]

entry id::= R_1
data coordinate type::= Long-Lat|Map-Proj|Digitizer
data coordinate format::= I|E|R
horizontal coordionate units::= DD|DMS|FT|IN
vertical coordinate units::= FT|MT
estimate of coordinate accuracy::= <X accuracy> <UD>
                                <Y accuracy>
                                [<UD + Z accuracy>]

    UD + Z accuracy::= <UD> <Z accuracy>
transformation parameters::= <A1> <UD>
                                <A2> <UD>
                                <A3> <UD>

```



```

                                <A4>
                                [<UD + A5>]
                                [<UD + A6>]

UD + A5 := <UD> <A5>
UD + A6 := <UD> <A6>
FD + coordinate resolution := <FD> <coordinate resolution>
FD + map scale denominator := <FD> <map scale denominator>
FD + number of registration points :=
                                <FD> <umber of registration points>
RD + map projection record :=
                                <RD>
                                <entry id> <FD>
                                <definition reference> <FD>
                                [<zone>] <FD>
                                <number of parameters> [<FD>]
                                {<parameter>}

entry id := R_2
definition reference := GCTP| <other reference>
RD + coverage record :=
                                <RD>
                                <entry id> <FD>
                                <minimum local coordinate values> <FD>
                                <maximum local coordinate values> <FD>
                                <minimum ground reference coordinate values>
                                <FD>
                                <maximum ground reference coordinate values>
                                <FD>
                                <SW corner of coverage area> <FD>
                                <NE corner of coverage area>

entry id := R_3
minimum local coordinate values := <minimum X> <UD>
                                <minimum Y>
                                [<UD + minimum Z>]

UD + minimum Z := <UD> <minimum Z>
maximum local coordinate values := <maximum X> <UD>
                                <maximum Y>
                                [<UD + maximum Z>]

UD + maximum Z := <UD> <maximum Z>
minimum ground reference coordinate values :=
                                <minimum X> <UD>
                                <minimum Y>
                                [<UD + minimum Z>]
maximum ground reference coordinate values :=
                                <maximum X> <UD>
                                <maximum Y>
                                [<UD + maximum Z>]
SW corner of coverage area := <longitude of corner> <UD>
                                <latitude of corner>

```



```

NE corner of coverage area ::= <longitude of corner> <UD>
                                <latitude of corner>
RD + Registration point record ::=
    <RD>
    <entry id> <FD>
    <label/comment> <FD>
    [<ground reference type code>] <FD>
    <ground reference coordinates> <FD>
    <local data coordinates>

entry id ::= R+ <digit> <digit>
ground reference type code ::= <blank>|<zero>|
                                <non-zero,non-blank>
ground reference coordinates ::= <long-lat>|
                                <X,Y>

    long-lat ::= <long> <UD> <lat>
    X,Y ::= <X> <UD> <Y>
local data coordinates ::= <x> <UD> <y>
data section ::= [<SD + point section>]|
                 [<SD + line section>]|
                 [<SD + polygon section>]|
                 [<SD + feature section>] |
                 [<SD + grid section>]|
                 [<SD + attribute description section>]
SD + point section ::= <SD>
    <section identification record>
    {<RD + point record>}
section identification record ::= P <UD> POINT
RD + point record = <RD> <point record>
point record ::= <entry id> <FD>
    <location> <FD>
    <number of first level attributes> <FD>
    <number of line pointers> <FD>
    <attributes> <FD>
    <line identifiers>
entry id ::= P <object type> <integer>
    object type ::= G|T|A|N
location ::= <X> <Y>
attributes ::= {<attribute node>}
    attribute node ::= <triplet>|
        <subtree>
subtree ::= <BL> <attributes> <EL>
triplet ::= <attribute name> <UD>
    <attribute format> <UD>
    <attribute value> [<FD>]
attribute format ::= I|R|E|T|
attribute value ::= <value>
    {<UD + value>}
UD + value ::= <UD> <value>

```



```

line identifiers::=<line identifier>
                {<UD + line identifier>}
line identifier::= L <object type> <integer>
UD + line identifier::= <UD> <line identifier>

SD + line section::= <SD>
                    <section identification record>
                    {<RD + line record>}
section identification record::= L <UD> LINE
RD + line record::= <RD> <line record>
line record::= <entry id> <FD>
               <area ids> <FD>
               <node ids> <FD>
               <number of first level attributes> <FD>
               <number of coordinates> <FD>
               <attributes> <FD>
               <coordinates>
entry id::= L <object type> <integer>
object type::= S|W|E|L
area ids::= <left area identifier> <UD>
            <right area identifier>
left area identifier::= <section code>
                       <object type>
                       <integer>
section code::= P|A
right area identifier::= <section code>
                       <object type>
                       <integer>
node ids::= <starting node identifier>
            <ending node identifier>
starting node identifier::= P <object type>
                           <integer>
ending node identifier::= P <object type>
                           <integer>
coordinates::= <x,y>
               {<FD + x,y>}
x,y::= x <UD> y
FD + x,y::= <FD> <x,y>
SD + polygon section::= <SD>
                       <section identification record>
                       {<RD + polygon record>}
section identification record::= A <UD> POLYGON
RD + polygon record::= <RD> <polygon record>
polygon record::= <entry id> <FD>
                  <number of first level attributes> <FD>
                  <number of boundaries> <FD>
                  <attributes> <FD>
                  {<boundary>}

```



```

entry id ::= A <object type> <integer>
object type ::= P|C|R
boundary ::= <number of fields>
             {<FD + boundary field>}
FD + boundary field ::= <FD> <boundary field>
boundary field ::= <coordinate field>|
                  <identifier field>
coordinate field ::= <x> <UD> <y>
identifier field ::= <section code>
                   <object type>
                   <integer>
section code ::= L|A
SD + feature section ::= <SD>
                       <section identification record>
                       <feature records>|
                       {<feature + data records>}
section identification record ::= F <UD> FEATURE
feature records ::= {<RD + feature record>}
RD + feature record ::= <RD> <feature record>
feature record ::=
    <entry id> <FD>
    <number of first level attributes> <FD>
    <number of data record identifiers> <FD>
    <attributes> <FD>
    [<data record identifiers>]
entry id ::= FF <integer>
data record identifiers ::=
    <data record identifier>
    <FD + data record identifier>
data record identifier ::= <section code>
                        <object type>
                        <integer>
section code ::= P|L|A|F|G|T
feature + data records ::= <feature record>
                        {<RD + data record>}
RD + data record ::= <RD>
                  <point record>|
                  <line record>|
                  <area recrd>|
                  <feature record>
SD + grid section ::= <SD>
                    <section identification record>
                    {<grid>}
section identification record ::= G <UD> GRID
grid ::= <RD + grid information record>
        {<RD + grid data record>}
RD + grid information record ::= <RD>
                              <grid information record>

```



```

grid information record ::=
    <encoding type> <FD>
    <number of layers in grid> <FD>
    <layer names> <FD>
    <layer value formats> <FD>
    <cell encoding types> <FD>
    <origin coordinates> <FD>
    <offsets of first cell> <FD>
    <orientation angle> <FD>
    <column direction> <FD>
    <row direction> <FD>
    <grid resolution elements> <FD>
    <minimum grid values> <FD>
    <maximum grid values> <FD>
    <number of rows> <FD>
    <number of columns> <FD>
    <number of grid data records> <FD>
    <number of attributes> <FD>
    <attributes> <FD>
encoding type ::= I|J|K|M
layer names ::= <layer name>
                {<UD + layername>}
    UD + layername ::= <UD> <layername>
layer value formats ::= <format>
                        {<UD + format>}
format ::= I|R|E|T
    UD + format ::= <D> <format>
cell encoding types ::= <encoding type>
                        {<UD + encoding type>}
encoding type ::= B|D|F|P|C
origin coordinates ::= <origin x> <UD> <origin y>
offsets of first cell ::= <offset x> <UD> <offset y>
column direction ::= LR|RL
row direction ::= TB|BT
grid resolution elements ::= <x resolution> <UD>
                             <y resolution> <UD>
                             <z resolution> <UD>
minimum grid values ::= <layer minimum>
                        {<UD + layer minimum>}
    UD + layer minimum ::= <UD> <layer minimum>
maximum grid values ::= <layer maximum>
                        {<UD + layer maximum>}
    UD + layer maximum ::= <UD> <layer maximum>
RD + grid data record ::= <RD> <grid data record>
grid data record ::= <entry id>
                    <row, column>
                    <x,y,z at row,column>
                    <number of first level attributes>

```



```

        <attributes>
        {<FD> <cell values>}
entry id::= <G> <object code> <integer>
  object code::= I|J|K|M
row,column::= <row number> <UD> <column number>
x,y,z at row column::= <x> <UD> <y> <UD> <z>
FD + cell values::= <FD> <cell values>
  cell values::= <layer value>
    {<UD + layer value>}
    UD + layer value::= <UD> <layer value>
SD + attribute description::= <SD>
    <section identification record>
    {<RD + attribute description record>}
section identification record::=
    T <UD> ATTRIBUTE DESCRIPTION
RD + attribute description record::=
    <RD>
    <attribute description record>
attribute description record::= <attributes>

```

C.6 GES

C.6.1 GES BNF

```

GES::= <geoblock directory file> & <overlay file> &
  <area file> & <line file> & <junction point file> &
  <control point file>
Geoblock directory file::= {<geoblock record>}
geoblock record::=
  <geoblock definition>
  {<overlay id>}
  {<overlay change date>}
  {<overlay jp cells on axis>}
  <available space pointers>
  <control point overlay record number>
  <geoblock name>
  <version number>
  <hole indicator>
  <minimum tolerance used>
  <file limits>
geoblock definition::= <geoblock center>
  <geoblock width>
  <geoblock height>
geoblock center::= <longitude> <latutude>
available space pointers::=
  <pointer to free list for area file>
  <pointer to eof +1 for area file>

```



```

        <pointer to free list for jp file>
        <pointer to eof +1 for jp file>
        <pointer to free list of lines file>
        <pointer to eof +1 for lines file>
geoblock name::= {<character>}
version number::= 0|<ge 500>
hole indicator::= 0|1
file limits::= <area file limit>
               <junction point file limit>
               <lines file limit>
               <control point file limit>
overlay file::= {<overlay record>}
overlay record::= <area overlay record>|
                  <lines overlay record>|
                  <control point overlay record>
area overlay record::= <pointer to first jp first cell>
                       {<class id>}
                       {<pointer to first area for class>}
                       {<pointer to last area for class>}
                       <pointer to first primary area>
                       <pointer to last primary area>
                       <area overlay type>
                       <overlay name>
area overlay type::= 0
overlay name::= {<character>}
lines overlay record::=<pointer to first jp first cell>
                      {<class id>}
                      {<pointer to first line for class>}
                      {<pointer to last line for class>}
                      <filler> <filler>
                      <line overlay type>
                      <overlay name>
line overlay type::= 2
control point overlay record::=
    <eof +1 for control point file>
    {<class id>}
    {<pointer to first record for class>}
    {<pointer to last record for class>}
    <filler><filler>
    <point overlay type>
    <overlay name>
point overlay type::= 1
area file::= {<area file record>}
area file record::= <logical area record>|<label record>
logical area record::= {<physical area record>}
physical area record::=
    <class pointers>
    <hole pointers>

```



```

    <same level pointers>
    <total number of vertices in area>
    <filler>
    <rectangular envelope>
    {<pointer to area junction point>}
    {<index of line segment out of jp>}
    <pointer to next physical record for area>
    <class id>
    {<label point>}
    <perimeter>
    <area>
    <number of label points>
    <pointer to additional labels record>
    <filler>
class pointers::= <pointer next area in class>
                  <pointer previous area in class>
hole pointers::= <pointer to first hole>
                  <pointer to containing area>
same level pointers::=
    <pointer next area same level>
    <pointer previous area same level>
rectangular envelope::= {<envelope vertex point>}
envelope vertex point::= <x coordinate>
                        <y coordinate>
label point::= <x coordinate>
               <y coordinate>
label record::= {<label point>}
junction point file::= {<junction point record>}
junction point record::=
    <pointer next junction point in cell>
    <junction point>
    {<line segment pointer>}
junction point::= <x coordinate> <y coordinate>
line file::= {<line segment logical records>}
line segment logical record::=
    {<line segment physical record>}
line segment physical record::=
    <area record>|
    <line segment record>
area record::=
area record::= <pointer next record in line segment>
               <total vertices in line segment>
               <adjacent areas>
               {<line segment point>}
               <overlay number>
               <line segment record sequence number>
adjacent areas::= <first area record number>
                  <second area record number>

```



```

    line segment point ::= <x coordinate> <y coordinate>
line segment record ::=
    <pointer next record in line segment>
    <total vertices in line segment>
    <class pointers>
    {<line segment point>}
    <overlay number>
    <line segment record sequence number>
control point file ::= {<control point record>}
control point record ::= <pointer next record in class>
    <number of points in record>
    {<control point>}
control point ::= <x coordinate> <y coordinate>
    <point name>
point name ::= {<character>}

```

C.6.2 GES Tree Structure

GES

```

<geoblock directory file>
{<geoblock record>}
    <geoblock definition>
        <geoblock center>
            <longitude>
            <latitude>
        <geoblock width>
        <geoblock height>
    {<overlay id>}
    {<overlay change date>}
    {<overlay jp cells on axis>}
    <available space pointers>
        <pointer to free list for area file>
        <pointer to eof +1 for area file>
        <pointer to free list for jp file>
        <pointer to eof +1 for jp file>
        <pointer to free list of lines file>
        <pointer to eof +1 for lines file>
    <control point overlay record number>
    <geoblock name>
        {<character>}
    <version number>
        0|<ge 500>
    <hole indicator>
        0|1
    <minimum tolerance used>
    <file limits>
        <area file limit>

```



```

        <junction point file limit>
        <lines file limit>
        <control point file limit>
<overlay file>
  {<overlay record>}
    <area overlay record>
      <pointer to first jp first cell>
      {<class id>}
      {<pointer to first area for class>}
      {<pointer to last area for class>}
      <pointer to first primary area>
      <pointer to last primary area>
    <area overlay type>
      0
    <overlay name>
      {<character>}
  <lines overlay record>
    <pointer to first jp first cell>
    {<class id>}
    {<pointer to first line for class>}
    {<pointer to last line for class>}
    <filler>
    <line overlay type>
      2
    <overlay name>
      {<character>}
  <control point overlay record>
    <eof +1 for control point file>
    {<class id>}
    {<pointer to first record for class>}
    {<pointer to last record for class>}
    <filler>
    <point overlay type>
      1
    <overlay name>
      {<character>}

<area file>
  {<area file record>}
    <logical area record>
      {<physical area record>}
        <class pointers>
          <pointer next area in class>
          <pointer previous area in class>
        <hole pointers>
          <pointer to first hole>
          <pointer to containing area>
        <same level pointers>

```



```

        <pointer next area same level>
        <pointer previous area same level>
    <total number of vertices in area>
    <filler>
    <rectangular envelope>
        {<envelope vertex point>}
            <x coordinate>
            <y coordinate>
        {<pointer to area junction point>}
        {<index of line segment out of jp>}
    <pointer to next physical record for aea>
    <class id>
    {<label point>}
        <x coordinate>
        <y coordinate>
    <perimeter>
    <area>
    <number of label points>
    <pointer to additional labels record>
<label record>
    {<label point>}
        <x coordinate>
        <y coordinate>
<line file>
    {<line segment logical record>}
    {<line segment physical record>}
        <area record>
            <pointer next record in line segment>
            <total vertices in line segment>
            <adjacent areas>
                <first area record number>
                <second area record number>
            {<line segment point>}
                <x coordinate>
                <y coordinate>
            <overlay number>
            <line segment record sequence number>
        <line segment record>
            <pointer next record in line segment>
            <total vertices in line segment>
            <class pointers>
                <pointer next area in class>
                <pointer previous area in class>
            {<line segment point>}
                <x coordinate>
                <y coordinate>
            <overlay number>
            <line segment record sequence number>

```



```

<junction point file>
  {<junction point record>}
    <pointer next junction point in cell>
    <junction point>
      <x coordinate>
      <y coordinate>
    {<line segment pointer>}
<control point file>
  {<control point record>}
    <pointer next record in class>
    <number of points in record>
    {<control point>}
      <x coordinate>
      <y coordinate>
      <point name>
      {<character>}

```

C.6.3 GES Characteristics

```

characteristics: physical
                  logical
                  logical spatial (topology)
                  locational
                  attribute
physical: disk format
  disk format: number of files
                access method
                record length
                record type
                machine wordsize
  number of files: multiple
  access method: direct
  record length: fixed
  record type: binary
  machine wordsize: 16 bit (HP)
logical: number of logical files
         file structure
         record structure
  number of logical files: multiple
  file structure: sequential
    sequential: linked list
  record structure: multiple physical record/logical record
logical spatial: point
                 line
                 region
point: dimensionality
      representation

```


- dimensionality: 2D
- representation: true point
- line: non-intersecting
 - node snapped
- region: basic entities
 - island handling
 - label points
- basic entities: arc-node
 - arc-node: element type groups
 - element type referencing
 - element type groups: nodes
 - arcs
 - regions
 - element type referencing: direct linkage
 - indirect reference
 - direct linkage: node-arc
 - arc-region
 - region-node
 - outer polygon-outer polygon
 - outer polygon-island
 - island-outer polygon
 - indirect reference: node-region
 - arc-node
 - region-arc
- island handling: islands
 - islands: connectors
 - polarity
 - hole-plug storage
 - hole-plug correspondence
 - outer polygon-islands coincidence
 - connectors: no artificial connectors
 - polarity: no polarity
 - hole-plug storage: single storage
 - hole-plug correspondence: one to one
 - outer polygon-island singularity: hole
 - separate polygon
- label points: multiple
- locational: reference system
 - encoding method
 - geo-block structure
 - resolution
 - tolerances
 - edge matching
- reference system: Cartesian
 - Cartesian: ground system
 - ground system: projection
 - measurement units
 - projection: lat-lon


```

    measurement units: internal
encoding method: relative coordinates
    relative coordinates: offset relative to origin geoblock
geo-block structure: number of geoblocks
                        geoblock size
                        geoblock origin
                        geoblock alignment
                        geoblock shape

number of geoblocks: multiple
geoblock size: filesize limited
geoblock origin: center
geoblock alignment: north-south
geoblock shape: rectangular
resolution: variable (with geoblock size)
edge matching: not edge matched
attribute type: standard
attribute structure: interleaved
    interleaved: number of attribute records
                  data types
                  correspondence with spatial elements
                  class groupings
    number of attribute records: zero
    data types: signed integer
                character string
    correspondence with spatial elements: one to one
    class groupings: linked list

```

C.7 IDIMS

C.7.1 IDIMS ICARAS BNF

```

idims icaras::=<polygon file>&
                <edge file>
polygon file (file d)::={<polygon>}
polygon::=<polygon header record>
        {<polygon coordinate record>}
polygon header record::=<polygon number>
                        <number of vertexes>
                        <polygon class>
polygon coordinate record::=<vertex number>
                        <x coordinate>
                        <y coordinate>
edge file(file f)::={<edge>}
edge::=<edge header record>{<edge coordinate record>}
edge header record::=<edge number>
                    <left polygon number>
                    <left polygon class>

```



```

        <left polygon area>
        <right polygon number>
        <right polygon class>
        <right polygon area>
        <number of vertexes>
edge coordinate record::={<vertex point>}[<filler>]
vertex point::=<x coordinate><y coordinate>

```

C.7.2 IDIMS ICARAS Tree Structure

```

idims icaras
  <polygon file>
    {<polygon>}
      <polygon header record>
        <polygon number>
        <number of vertexes>
        <polygon class>
      {<polygon coordinate record>}
        <vertex number>
        <x coordinate>
        <y coordinate>
  <edge file>
    {<edge>}
      <edge header record>
        <edge number>
        <left polygon number>
        <left polygon class>
        <left polygon area>
        <right polygon number>
        <right polygon class>
        <right polygon area>
        <number of vertexes>
      {<edge coordinate record>}
        {<vertex point>}
          <x coordinate>
          <y coordinate>
        [<filler>]

```

C.8 ISO 8211

C.8.1 ISO 8211 BNF

```

ISO 8211 file::= <ISO standard file labels>
                <data descriptive file>
                <ISO file termination indicator>
data descriptive file::= <data descriptive record>

```



```

                                {<data record>}
data descriptive record ::= <leader>
                                <directory>
                                <FT>
                                <data descriptive area>
                                <FT>

leader ::= <record length>
            <interchange level>
            <leader identifier>
            <inline code extension indicator>
            <reserved>
            <application indicator>
            <field control length>
            <base address of data descriptive area>
            <extended character set indicator>
            <entry map>
record length ::= {<digit>}
interchange level ::= 1|2|3
leader identifier ::= L
inline code extension indicator ::= SP|E
application indicator ::= <SPACE character>
field control length ::= 00|06|09
base address of data descriptive area ::= {<digit>}
extended character set indicator ::= (2/0)(2/0)(2/0)|
                                    (2/0)(2/1)(2/0)|
                                    <truncated escape sequence>

entry map ::= <size of field length>
            <size of field position>
            <reserved>
            <size of field tag>
            size of field length ::= 1|2|3|4|5|6|7|8|9
            size of field position ::= 1|2|3|4|6|7|8|9
            reserved ::= 0
            size of field tag ::= 1|2|3|4|5|6|7
directory ::= <header entries> {<entry>} <FT>
header entries ::= [<file control field identifier>
                  <record identifier field identifier>
                  [<optional user field identifier>]
                  {<reserved entry>}]
file control field identifier ::= <entry>
record identifier field identifier ::= <entry>
optional user field identifier ::= <entry>
entry ::= <field tag>
            <field length>
            <field position>
            field tag ::= {<character>}
            field length ::= <right justified integer>
            field position ::= <right justified integer>

```



```

    right justified integer ::= {0} <integer>
data descriptive area ::=
    <file control field> <FT>
    <data description of record identifier field> <FT>
    [<optional user field>]
    {<data descriptive field + FT>}
file control field ::= <field control field>
    [file title field]
    [<list of tag pairs for hierarchy>]
FT ::= <(RS)1/14 ISO 646 information separator>
field control field ::= {0}|{SPACE}
    <(RS)1/14,(US)1/15 graphics symbols>
file title field ::= {<character>}
list of tag pairs for hierarchy ::= {<tag pair>} <UT>
tag pair ::= <field tag> <field tag>
UT ::= <(US)1/15 ISO 646 information separator>
data description of record identifier field ::= <field controls>
    <data field name>
data descriptive field + FT ::= <data descriptive field> <FT>
data descriptive field ::= <field controls>
    <data field name>
    <UT>
    [<vector or array information>]
    <UT>
    [<format controls>]
field controls ::= <structure type code>
    <reserved>
    <(RS)1/14,(US)1/15 graphics symbols>
structure type code ::= <digit><digit>
reserved ::= <0> <0>
1/14,1/15 systems delimiters graphics symbol ::= <character>
    <character>
datafield name ::= {<character>}
vector or array information ::= <vector label>|
    <cartesian label>
vector label ::= {<label> <|>} <label>
cartesian label ::= {<composite label> <*>} <composite label>
composite label ::= {<label> <|>} <label>
label ::= <(1/15) (2/10)>
label ::= {<character>}
format controls ::= <(> <expression> <)>
expression ::= [k]<format controls>|
    <expression> <,> <Y>|
    [m]<Y>
k ::= <positive integer>
m ::= <positive integer>
Y ::= <Z>|<Z(*)>|<Z(n)>
Z ::= A|I|R|S|C|B|X

```



```

        Z(*) ::= <Z> <character>
        Z(n) ::= <Z> <positive integer>
data record ::= <leader>
                <directory>
                <FT>
                <user data area>
                <FT>
leader ::= <record length>
            <reserved>
            <leader identifier>
            <reserved>
            <base adress of data area>
            <reserved>
            <entry map>
record length ::= {<digit>}
leader identifier ::= D|R
user data area ::= <record identifier field> <FT>
                    [<array information>]
                    {<user data field + FT>}
array information ::= <array dimension>
                    {<dimension extent>}
array dimension ::= <positive integer>
dimension extent ::= <positive integer>
user data field + FT ::= <user data field> <FT>
user data field ::= {<elementary character data field>}|
                    {<compound data field>}
elementary character data field ::= {<character>} <FT>
compound data field ::= <elementary field>|
                        <vector>|
                        <array>|
                        elementary field ::= <character field>|
                                                <implicit point field>
                                                <explicit point field>
                                                <explicit point scaled field>
                                                <character mode bit string field>
                                                <bit field>
vector ::= {<elementary field>}
array ::= {<elementary field>}

```

C.9 MOSS

C.9.1 MOSS ADD FORMAT

```

MOSS ADD Format ::= <standard format>|<soils data format>
standard format ::= {<polygon>}|{<line>}|{<point>}
polygon ::= <header record>{<point record>}
header record ::= [<item number>]

```



```

        <filler>
        <subject name>
        <filler>
        <number of points>
        <ASCII new line character>
item number::=<regular item number>
        <lat long item number>
    regular item number::=<integer>
    lat long item number::=<negative integer>
    subject name::={<character>}
point record::=<regular point>|<island starting point>
regular point::=<x coordinate>
        <y coordinate>
        <ASCII new line character>
island starting point::=<negative x coordinate>
        <y coordinate>
        <ASCII new line character>
line::=<header record>{<point record>}
point::=<header record><point record>
soils data format::={<polygon>}|{<line>}|{<point>}<eof record>
polygon::=<parent polygon>{<island polygon>}
parent polygon::=<header record>{<point record>}
    header record::=[<item number>]
        <filler>
        <subject name>
        <filler>
        <number of points>
        <filler>
    item number::=<regular item number>
        <lat long item number>
    regular item number::=<integer>
    lat long item number::=<negative integer>
    subject name::={<character>}
point record::=<point>
    point::=<x coordinate>
        <y coordinate>
        <filler>
    island polygon::=<header record>{<point record>}
line::=<header record>{<point record>}
point::=<header record><point record>
eof record::='999999'

```

C.9.2 MOSS IMPORT/EXPORT BNF

```

MOSS External::= {<feature record pair>}
feature record pair::= <header record> <coordinate record>
    header record::= <filler>

```



```

        <subject>
        <filler>
        <number of points>
coordinate record ::= {<point>}
point ::= <regular point>
        <island starting point>
regular point ::= <UTM easting> <UTM northing>
island starting point ::= <negative UTM easting>
                        <UTM northing>

```

C.9.3 MOSS IMPORT/EXPORT Tree Structure

```

MOSS External
{<feature record pair>}
  <header record>
    <filler>
    <subject>
    <number of points>
  <coordinate record>
    {<point>}
      <regular point>
        <UTM easting>
        <UTM northing>
      <island starting point>
        <negative UTM easting>
        <UTM northing>

```

C.10 VDI(RIM)

```

map ::= <spatial relations> <attribute relations>
spatial relations ::= [<region-polygon relation>]
                    [<polygon-arc relation>]
                    [<arc-header relation>]
                    [<arc-xy relation>]
                    [<node-arc relation>]
                    [<node-xy relation>]
                    [<label-xy relation>]
region-polygon relation ::= {<REGPOL relation row>}
  REGPOL relation row ::= <region number>
                        <polygon number>
polygon-arc relation ::= {<POLARC relation row>}
  POLARC relation row ::= <polygon number>
                        <arc number>
arc-header relation ::= {<ARCHDR relation row>}
  ARCHDR relation row ::= <arc number>
                        <start node>

```



```

                                <end node>
                                <left region>
                                <right region>
arc-xy relation::= {<ARCXY relation row>}
  ARCXY relation row::= <arc number>
                        <x coordinate>
                        <y coordinate>
```


D Input Interfaces

All interfaces used by the EROS Data Center are listed below, and information on their history, use and status is provided if available. Interfaces are categorized as INPUT from different sources (maps, DLG files, RIM relations, etc.) into systems described in Appendix A, or as OUTPUT interfaces from those systems in Appendix A to maps, files, or other systems.

D.1 ADS from MAP

The map to ADS interface consists of the data entry module of the BLM's ADS/MOSS/COS. ADS was originally designed to be the data entry module to an automated land record system (ALMRS), and consists of several modules performing separate tasks related to data input, including data capture, editing, output, data base maintenance, and links to the analysis software. Point, line, or areal data can be digitized with ADS, using one of several types of digitizing modes, including point-to-point, stream, or increment. Formation of polygons is handled by a separate routine in ADS, called CLOSEPOLY, which identifies intersections in digitized line data, creates nodes, forms a file of the nodes and their joining arcs, and finally forms polygons, tagging each with its attribute(s). CLOSEPOLY can be performed interactively or in batch processing and allows the operator to digitize the linework randomly and without regard to direction, intersections, or attributes (referred to as free form or spaghetti digitizing).

One of the elements of ADS that makes it versatile is its editing capability. The editing module has more than 40 commands that allow interactive editing of lines, points, and attributes, as well as a variety of display commands to allow plotting, windowing or zooming, overlaying, and edgematching.

Users of ADS find the extensive editing capability of tremendous benefit for maintaining quality control. The commands are easy to use with sufficient on-line help documentation to allow the new user to use even the most obscure editing command. The spaghetti method of digitizing is extremely efficient in terms of labor. Users and project managers also appreciate the ease with which ADS allows updating of existing map files, and the variety of methods for labelling and tagging features.

Several weaknesses have been identified through intensive use of the system:

1. The registration routines are limited. It has been suggested that geometric fits for predicting coordinate values is not valid for many projections. The system also requires that all sources be in a quad format so that patching can be accomplished. This restriction eliminates many source materials with the most up-to-date information.
2. The ADS to MOSS conversion routine is limited. Data can be converted to MOSS format in the UTM projection only. This limits the user's capability in terms of output products.
3. ADS has no projection capability.
4. System and programming documentation is sometimes out-of-date, sketchy, or non-existent, making system updates, corrections, and enhancements difficult. There is also a lack of documentation regarding the limits of the system.

A study in cooperation with the BLM and the Fish and Wildlife Service analyzed the capabilities of the two data entry systems used by these agencies to identify areas in which the two systems were

similar and dissimilar and make recommendations for the improvement of each. It is the BLM's intention to enhance its system to reflect the best of both systems.

Some specific plans are:

1. To improve the registration routines to handle nonformatted source materials.
2. To enhance its compilation capabilities. Users would like to see some capability for building data from existing data, such as aliquot parts from sections, and keyboard entry of coordinates.
3. To incorporate projection capability to make output more versatile.
4. To improve the availability of all types of documentation.

Supporting documentation:

1. "BLM's Automated Digitizing System: ADS", by M. Mogg, in Workshop on Methods and Techniques for Digitizing Data, U.S. Geological Survey Open File Report 85-307.

D.2 AGIS/GRAM from MAP

Digitizing a hardcopy map is accommodated primarily by the functions on the digitizing menu which operate on the table digitizer input. Small maps can be entered via the graphics control tablet.

For table digitizing, a display control graphics menu can be located on the table for easy operation of the graphics display. Map projection and registration to the database is accomplished first, allowing digitization of modes, lines, labels, symbols, or regions. Digitizing can be interrupted with all session status information saved to facilitate resuming digitizing at a later time.

For tablet digitizing, the editing menus for graphics data interaction from the tablet can be used if the map segments are small enough to fit the active map portion of the tablet. Several display functions must be used to scale the graphics display area to the map, and the map must be manually registered on the tablet. The EDIT LINE/NODES menu can be used for the data entry.

D.3 AGIS/GRAM from VDI(RIM)

The description of this interface can be found in the system description for the AGIS/GRAM system.

D.4 AHDS from MAP

The Alaska Hydrologic Digitizing System (AHDS) digitizing interface is inherently contained in AHDS. AHDS was developed by the Alaska Bureau of Land Management (BLM). Originally it was used to digitize surface water features, hydrography, and lakes. The system has evolved and is used for various purposes, such as the generation of Master Title Plats (1:250,000) from protraction diagrams.

The system is maintained by Bob Caughey and Jim Braham, Box 13, BLM, 701 C St., Alaska, 99513. 907-271-5057.

D.5 AMS from MAP

The EROS Data Center has not directly used the digitizing method, but will be using digital data captured with AMS. AMS is one of the vector input systems for MOSS. In addition, AMS data will likely be used as input to any of several other interfaces, depending upon the desired output system. The "AMS" in this description refers to the Autometric version of AMS used by CMD. The contact at CMD is Paul D. Harrison, FTS 323-5343 or (303) 226-6183.

Two species of AMS exist, one described above and another that has been modified and/or maintained by the Goddard Space Flight Center (GSFC). The Goddard system was initially designated the official digitizing system of LAS.

AMS has no automatic line thinning capability; this must be performed in the conversion process or in the target output system if required.

There are no limits to the number of polygons that can be contained within an AMS file; however, a single polygon is restricted to 32,000 arcs. This limitation should not be a concern for many projects. Attribute assignments are limited to 17 alphanumeric characters.

Of the files that AMS creates, those that are required for an eventual VDI conversion contain: normal/temporary nodes, segment indices (headers), edge nodes, polygons, and segment coordinates. There is a knot-removal option in AMS. This should be exercised if the data are headed for the VDI conversion process.

The digitizing subsystem is comprised of six programs:

DIGS0, the main program, sets up the digitizing process.

DIGS1 controls digitizing and editing.

ATRPOS positions attributes within polygons for a final plot.

VERF performs data verification and forms polygons.

POLY performs a point-in-polygon test.

CROS1 checks to see if a line crosses itself.

A map that is to be digitized can be subdivided into geo units of any size. Each geo unit is digitized and treated as a separate entity, and is enclosed within a boundary line. Each geo unit accesses edge node files of adjoining geo units for edge node extraction purposes and polygons are closed at the geo unit boundary.

Edge fitting in AMS is performed as part of the initial map set-up. In AMS, edge fitting only occurs with respect to segments (AMS does not compare attributes assigned to polygons that fall on two or more maps). Edge segments are forced to tie by being required for successful verification. Attributes are checked as part of the initial map set-up procedure via a manual process. Attribute consistency between quads can, however, be verified after the map has been moved into MOSS and merged with adjacent maps.

Segments in AMS are defined by beginning and ending nodes with a series of points established between those nodes. Nodes exist at all segment intersections, and therefore force mathematical equality at all junction points. Nodes that fall on the edge of the map are called edge nodes, and are

considered separately with respect to file structure. In the digitizing process, beginning nodes are created, points are then established along the entire length of the line, and an ending node is created as the terminal point in the segment. AMS provides for 'point' and 'stream' modes of digitizing; however, users such as CMD do not use the stream mode.

When the digitizer crosses the map edge while digitizing a segment, an edge node is automatically established. If the map adjacent to the edge that was crossed was previously digitized, an extracted edge node must be tied to it. If the segment being digitized does not cross the edge where the extracted edge node is located, then either a new edge node must be created or topological modifications need to be made to the segment in question.

When digitizing a segment in AMS, the attributes of the polygons for which that segment forms a boundary must be entered. Also, any center attribute that defines a linear or point feature is entered. As attributes are entered, any attribute not found in the classification scheme is identified as a possible error. Also, polygons with multiple or no labels or two adjacent polygons with the same labels are identified as errors.

Topological quality control is conducted after all segments on a map have been digitized. The digitizer creates a plot of raw lines at the same scale as the source data if possible. This plot is overlaid on the base map and the segments on the plot are compared with those on the source map. The plot is checked for missing segments, segments digitized twice, and segments that have missing points. All errors are then corrected in the job file using normal digitizing methods.

Polygon verification in AMS involves various processes. The primary function is for all lines to be combined into coherent groups (polygons) according to their attributes. Additionally, verification identifies each feature as a line, point, or polygon and checks to ensure that attributes entered for all segments forming a polygon have been coded the same. It also checks all edges to identify any extracted edge nodes that are detached and checks to see if any segments have crossed into adjacent polygons. It then assigns island polygons to parent polygons.

Errors detected during the verification procedure cause failure of the verification process. Factors causing verification failure may be missing segments, segments digitized twice, attributes entered incorrectly, edge nodes not tied to, or crossed segments. A listing of all errors is generated at the end of the verification process and printed out on the line printer, and the digitizer uses the editing capabilities to make necessary corrections. An AMS map is not considered verified until it has passed through the verification process from the beginning without error. TGS-CMD digitizers verify digitized maps in batch mode, thus freeing their tables for further digitizing.

AMS data can be quality checked for topology by comparing the plot at scale with the source map, but can be quality checked for attribute entries only by comparing a listing of all the polygon attributes associated with a given map with a valid classification scheme. Any illegal attributes can be readily identified in this fashion.

It is very inefficient and cost ineffective to evaluate every polygon on a map for accuracy of labeling. This may seem to be a major detriment to using AMS. However, since polygon attributes have to be entered when each segment is digitized, it is unusual that a digitizer will miscode the attribute the same for every segment of the polygon.

After successful verification, the digitizer runs a final plot and an attribute summary. The attribute summary provides the digitizer with information with respect to total acres in the map, total number of polygons, total number of unique attributes used, and a list of all unique attributes with total

polygons and total acreages associated with each. Polygons with unusually small acreages (indicating that a segment was inadvertently given a center attribute) are checked for accuracy. The digitizer is responsible for checking both plot and summary for accuracy. After the digitizer is satisfied that all data are correct, the supervisor is responsible for rechecking both plot and summary as a final quality checking step. Though this may appear to be duplication of effort, experience has indicated that after a digitizer has been working on the same map for several days, it is easy to overlook mistakes. Secondary quality control has proven as a valuable investment and provides consistent evaluation of all data digitized throughout a project.

Editing in AMS is a simple and versatile process that may be performed either in a job file at any point in the digitizing process, or within the database. If necessary, any node (including edge nodes) or segment can be deleted by entering the appropriate node or segment number. These numbers can be identified by using a command where the digitizer measures the node, or any point on a segment with the cursor, and is given the number of the feature in question. In the case where an edge node is deleted, a flag will signal that the adjacent map requires an update.

Attribute editing can be performed on individual segments and on polygons using the editing commands in AMS. These commands can be used whether or not the map is registered to the table. However, any segment attribute modifications must be followed by a successful verification before the map is considered completed. Polygon attribute modifications can be made without further verification if a successful verification has already been completed.

Editing line segments is accomplished in two different ways. The first is to delete a segment by number and redigitize it. The second is to remove the portion of the segment that is inaccurate and replace it without redigitizing the entire segment. In cases where segments need to be added that would tie to an existing segment in an area where no node exists, a node can be added to the existing segment and the new segment can then be attached to the new node.

A more involved procedure is required for maps that were previously completed and need updating as a result of source data modifications. If the map to be updated has been archived, it (and adjacent maps if the updating affects edges) has to be retrieved from archive and loaded onto the system. All status words of the target and adjacent maps have to be modified to allow the digitizer access to the map to be updated. The digitizer then defines the project, theme, map center, and indicates that it is an update in order to copy the map file from the database to the working directory for modification. The new data are then incorporated into the map using processes identical to those used in new map production.

Digitizing errors, as discussed in this section, refer to problems associated with the data files, rather than errors that result from digitizing mistakes such as attribute miscoding. Serious data file errors can occur as a result of abnormal terminations of the system, user "collisions", and a variety of other uncontrollable factors. Inability to perform file modifications after error conditions are discovered would require that a map be started over. This is a costly and risky way to approach a production digitizing environment. A utility program called RESCUE is routinely used to modify files internally when aborted digitizing sessions create errors in the data sets.

A software system designed specifically for analyzing problem data sets in AMS is DAP (Dataset Analysis Program) which evaluates AMS data files and reports to the operator all error and warning conditions that exist in a problem dataset. This utility can be performed within minutes so the operator need only to correct the file errors in order to continue work on the map.

When maps are archived, only the node, segment, edge node, polygon (which includes the first six

records of the restart file), and coordinate files are transferred.

D.6 ARC/INFO from ARC/INFO IMPORT/EXPORT

ESRI provides a two-way interface to other ARC/INFO systems using the Import/Export format. This format was introduced in release 3.1, to treat the problems of complete exchange of attributes and topology that occur through the use of DLG-3 interfaces. As with the DLG-3 optional format, it uses all ASCII encoding, which minimizes machine to machine conversion problems. In addition, it converts all ARC/INFO coverage files to be output and input, just as they exist, with the only conversion being to ASCII codes. It has been used occasionally especially for Vax to Prime move, and is a smooth flow interface. No problems have been encountered, and ESRI maintains it as part of the ARC/INFO system (no source code provided).

D.7 ARC/INFO from DIME

This interface is built into ESRI's ARC/INFO software, and is maintained by ESRI. These files are a primary source for the NETWORK subsystem of ARC/INFO. The conversion from GBF/DIME files to ARC/INFO coverages uses the DIMEARC command within ARC/INFO. The program is written in Fortran 77.

D.8 ARC/INFO from DLG

This interface is built into ESRI's ARC/INFO software and is maintained by ESRI. Files in this DLG format are converted to ARC/INFO coverages through the DLG3ARC command. Attribute data are transferred directly, but only to disk files. They must then be joined to each coverage with other subcommands. The program is written in Fortran 77.

D.9 ARC/INFO from DLG OPTIONAL

This interface is built into ESRI's ARC/INFO software, and is maintained by ESRI. Files in this DLG format are converted to ARC/INFO coverages through the DLGARC command. Attribute data are not transferred directly, but must be written to ASCII files, then copied to attribute tables. The program is written in Fortran 77, and represents a smooth flow for the data transferred.

D.10 ARC/INFO from GIMMS

ESRI provides an unsupported capability for ARC/INFO to read inputs from GIMMS.

D.11 ARC/INFO from MAP

ARC/INFO provides the DIGITIZE subsystem for map digitizing. This package is an integral part of the ARC/INFO system, and supports TALOS, ALTEK, Tektronics, GTCO, and Summagraphics

digitizers, as well as Tektronics displays (and emulators). The subsystem represents a smooth flow capability, is written in Fortran 77, and is maintained by ESRI (no source code supplied).

D.12 ARC/INFO from PIOS

ESRI has developed an interface from its predecessor system, PIOS, to ARC/INFO. The state of this software is unknown, since we have not used PIOS at the EROS Data Center, and have never attempted an interface. It is maintained by ESRI for users of PIOS, but is probably being used less and less, since few PIOS installations remain.

D.13 ARC/INFO from SVF

The Single Variable File (SVF) interface can be performed from ARC/INFO using the GRIDPOLY program, which converts raster data to arc vectors. The SVF is fairly versatile for topographic data. This interface is smooth flow, is written in Fortran 77, and is maintained by ESRI as part of the ARC/INFO system.

D.14 ARC/INFO from VDI(RIM)

Please reference section VDI(RIM) to ARC/INFO

D.15 CONTRACTOR FMT from MAP

The contractor format was developed by the USGS National Mapping Division, Reston, Va, which has a need to convert graphic data on categorical overlays of paper maps into digital form. Digitizing work has been contracted to meet increasing requirements for digital map data.

The interface can be termed smooth flow. Header file and arc file are produced. Categorical attributes are: land use, land cover, political boundaries, hydrologic units, census county subdivisions, Federal land ownership, and State land ownership. Data types are areas, lines, and labels.

Each contractor is responsible for developing the interface on site to produce the digital tapes. A sample FORTRAN program that can be used for verifying the data produced by the contractor is provided to the contractor.

There has been heavy use of the interface. Sample projects can be found in the USGS LUDA program. The interface has been in place since July 1980. Digitizing work is still being contracted.

D.16 COS from MOSS

COS (Cartographic Output System) can receive data from MOSS. This interface is well established and is described in the system description of MOSS and COS.

D.17 CRAIGES from MAP

The agency responsible for this interface is the USGS National Mapping Division, Reston, Va.

The Geological Survey must convert graphic data from categorical map overlays into Digital Line Graph (DLG) form. The CRAIGES system was developed to interactively create the digital data produced by the UCLGES software. The interface can be termed smooth flow.

An ALTEK raw file is produced. All DLG base categories are collected with this digitizing system. Data types collected are points, line, nodes and labels.

The interface consists of a FORTRAN program running on INTERGRAPH systems in each regional mapping center of the National Mapping Division.

The use of the interface is heavy. Sample projects for its use are the USGS DLG 1:24,000-, 1:100,000-, and 1:200,000-scale mapping programs.

A new version of this system is being developed at the EROS Data Center to eliminate certain deficiencies (see the INTERGRAPH system description).

D.18 GES from DLG and from DLG OPTIONAL

ESL provides three programs for input and output of DLG data. Regular DLG data are input to GES through the GES program DLG2GES, optional DLG data can be input through the utility DLG3IN, and data can be output in the form of an optional DLG through the GES program DLG3OUT.

DLG2GES will convert line and area features contained in the DLG data. It is the responsibility of the user, through a series of prompts, to identify what DLG overlays are wanted and where they should be stored. Individual features within a DLG overlay can be stored as separate classes depending on user selection.

Once the data are loaded, it is advisable for the user to review the data for obvious problems. This is especially important for area overlays that have holes. Apparently it is not possible for the hole-searching algorithm to identify every hole for all types of data, and some areas may not be closed even though the proper line segments exist. ESL advises that the data may have insufficient mathematical content and that visual judgement may be necessary. Especially problematic are figure 8's and holes that share a line segment with the primary boundary.

DLG2GES and DLG3IN both create ERIS files, while DLG3OUT reads an ERIS file.

The interfaces are definitely smooth interfaces. DLG2GES is not used frequently at the EROS Data Center. DLG3IN and DLG3OUT are not part of the GES installation at the EROS Data Center but can be purchased from ESL.

At the GMIS installation at Norton Air Force Base DLG3IN and DLG3OUT are used for exchanging data with ESRI's ARC/INFO.

D.19 GES from MAP

This digitizing system was developed and is distributed by Electromagnetic Systems Laboratory (ESL), Sunnyvale, California. The ESL product IDIMS required an interface to input digitized geographic data and convert it to raster form. GES (Geographic Entry System) was developed to augment the image processing capabilities of IDIMS.

Any attribute data can be captured by GES, but it must be used in conjunction with the tabular data handling system ERIS; GES itself only deals with class values.

GES was written in Fortran and runs on HP3000 and VAX machines in conjunction with the IDIMS system.

The system is used at the EROS Data Center for nearly all applications digitizing projects.

GES can produce data sets with bad topology. Lines can intersect at places other than nodes, and there may be duplicate lines or misplaced attributes.

D.20 GIMMS from ARC/INFO

ESRI provides an unsupported capability for ARC/INFO to produce input for GIMMS.

D.21 GIRAS from CONTRACTOR FMT

The agency responsible for this interface is the USGS National Mapping Division. The Survey must convert graphic data from categorical map overlays into digital form. Digitizing work has been contracted to meet increasing requirements for digital map data. The digitized data are delivered in the contractor format.

The interface consists of the FORTRAN program DIGITRD, which is used by the USGS for processing and testing delivered digital tapes. This program runs on the Amdahl mainframe computer at USGS headquarters in Reston, Va.

Usage of the interface is heavy, with frequent use for the USGS LUDA program.

The interface has been in place since July 1980. Digitizing work is still being contracted.

D.22 GIRAS from MAP

Reference the system description on GIRAS.

D.23 INTERGRAPH from DLG

The USGS National Mapping Division is responsible for this interface. The USGS/Census project requirements for 1:100,000-scale DLG data created a need for DLG input into the various Intergraph systems of the National Mapping Division. An interface was developed by Mid-Continent Mapping

Center and was later refined at the EROS Data Center to accomplish efficient loading of DLG data into Intergraph design files.

The interface is a smooth flow interface. It produces an Intergraph design file and supports all DLG base categories. Areas, lines, and labels can be handled.

The interface consists of a FORTRAN program that is part of the CRAIGES and EDCIDS systems. The interface can be used on all INTERGRAPH Systems with IGDS version 8.0 and higher.

The interface is frequently used. A sample project is the USGS 1:100,000-scale DLG project. The software runs much slower on a PDP 11/23-based system than on a PDP 11/70-based system. No data problems have been noted in transfers. The interface has been in place since mid-1984, and it has been used in full-scale DLG production.

D.24 INTERGRAPH from MAP

This interface runs on an Intergraph system from Intergraph Corporation, Huntsville, Ala. The interfacing software is provided by the USGS National Mapping Division, Reston, Va., and the EROS Data Center.

The USGS creates Digital Line Graphs from digitizing paper maps. The more complex paper maps to be digitized are automated on the National Mapping Division's Intergraph systems under the CRAIGES software. The CRAIGES software has also been refined at the EROS Data Center and has been incorporated into the EDCIDS (EROS Data Center Intergraph Digitizing System) software. Intergraph also markets a product called GPPU (Graphic Polygon Processing Utility) that can be used as a paper map digitizing system. The interface can be classified as a smooth flow interface.

Files produced are Intergraph design files. All DLG base categories and other thematic categories can be handled. Data types can be areas, lines, and labels.

The interface is a FORTRAN program, a part of the CRAIGES and EDCIDS packages. It runs on all Intergraph systems of IGDS version 8.0 or higher. GPPU is a combination of FORTRAN programs and Intergraph User Command procedures.

Usage of interface is frequent. Sample projects are the USGS 1:24,000-, 1:100,000-, and 1:2,000,000-scale DLG projects. GPPU has not been used for any large-scale digitizing projects at the EROS Data Center.

The programs run slowly on PDP 11/23-based systems as compared to a PDP 11/70. GPPU has problems in its overlay processing capabilities on the PDP machines.

GPPU has been under development since the late 1970's. The best version runs on the VAX-based Intergraph systems. CRAIGES was developed in the late 1970's by Craig Skalet at the Western Mapping Center. It is currently being supplemented by the PROSYS software under development by the NMD. EDCIDS has been in place since mid-1985. It has been used in full-scale DLG production.

D.25 INTERGRAPH from SIF

Standard Interchange Format (SIF) was developed by Intergraph Corporation for the exchange of CAD-CAM type data. As such it is not very suitable for the exchange of cartographic data, and cannot handle attribute data. The format has not been used for any exchange at the EROS Data Center.

D.26 MOSS from AHDS

The Alaska Hydrologic Digitizing System, which was designed primarily for the digitizing of surface water features, hydrography, and lakes, does have an input interface to MOSS, but the status of this interface is unknown.

D.27 MOSS from AMS

AMS (Analytical Mapping System) has capabilities for map capture or digitizing and format conversion to MOSS. Additional information on AMS can be found in the system description on AMS.

D.28 MOSS from DIME

In the course of reviewing MOSS data interfaces this interface was indicated to be a planned activity as described in documentation "Feasibility and Design Study for the Enhancement of MOSS". No specific agency or contact for the development has been identified.

D.29 MOSS from DLG OPTIONAL

This is actually a transformation of the DLG optional format into the MOSS import/export format, to be converted to the MOSS format by the Autometrics function ADD in MOSS. The conversion is a menu option in the MOSS utility command. Interactive prompts to the user request the name of the input DLG file; the input data projection (and possibly projection parameters); the desired output projection and parameters; the number of overlays; point, line, or polygon data type, and the output file name.

Only DLG3 ASCII optional format data can be accepted since these files contain topologically correct polygons with associated segment numbers.

D.30 MOSS from MAPS

Cell maps that are a part of the MAPS system can be used by the raster functions within MOSS. The general capability to convert cell data to polygon data is not known to exist.

D.31 MOSS from MOSS IMPORT/EXPORT

Conversion to MOSS format from IMPORT/EXPORT format is implemented as a MOSS function by Autometrics. Execution of the ADD function is possible from the MOSS session with a set of parameter arguments and is termed smooth flow. The function will prompt the user for the following parameters: input file name, desired output file name for the MOSS format data, an optional template file name for the map header default values, header information, and input data type (point, line, polygon, elevation).

Within any project on MOSS hardware/software, the usage of ADD to load import/export format data would likely be infrequent. Use might be made to transfer data from MOSS on dissimilar hardware or from an alternate GIS system with an adaptable export format capability.

D.32 MOSS from VDI(RIM)

The RIM-to-MOSS interface was developed at the EROS Data Center and is implemented as a program under VAX VMS (RIMMOSS).

Input is the following set of relations for each data type:

POINT : POINTXY and PRIMATT
LINE : ARCXYs and PRIMATT
REGION : REGXY and PRIMATT

The procedure generates a MOSS ADD file.

The creation of multiple attribute files is not directly supported, although they could be supplied. The Sandcrane data (BOR) and the ARP data have been sent through this conversion. It is anticipated that the interface will be used frequently.

D.33 TIGER from DLG

The data that are being gathered by the NMD for support of the Census project are sent to Census in a DLG format. Census has developed the capabilities to input this data into the TIGER system.

D.34 UCLGES from UCLGES UPDATE

The responsible agency for the interface is the National Mapping Division, Reston, Va. The production of DLG's has been under development by the National Mapping Division since the mid-1970's. A system called the Unified Coding of Line Graph Element System (UCLGES) was developed to produce DLG's from digitizer data. UCLGES runs on an Amdahl mainframe computer at National Mapping Division Headquarters. The original DLG was stored in IBM binary format. DLG's are currently distributed in ASCII for interchange with various computer systems.

The interface is a smooth flow interface.

1. Files produced - A binary form of the DLG; the Digital Cartographic File (DCF).

2. Attributes - All DLG base category attributes.
3. Data types - Areas, lines, nodes, degenerate lines, and attributes.

The interface consists of one or two FORTRAN programs (FRMDCF and optionally INTDCF). INTDCF is used when a new Digital Cartographic File is to be initialized. FRMDCF reads a transaction file from either the ALTEK or GRADICON digitizer systems. The software is executed by IBM Job Control Language. The program is normally run in a batch queue. The usage of the interface is heavy since all regional mapping centers send digitized data to Washington, D.C., to be processed into UCLGES.

Sample projects are the USGS 1:2,000,000-, 1:100,000-, and 1:24,000-scale national mapping programs. The software has been converted to run on Gould 9780 mini-computers under a new system entitled PROSYS.

D.35 VDI(RIM) from ALMRS

In support of the Federal Mineral Land Information System (FMLIS) project it is desired to use data that have been prepared by BLM from within ALMRS. The interface to VDI(RIM) is currently in the planning phases. The largest concern is the ability to relate legal description to geographic coordinates.

D.36 VDI(RIM) from AMS

The AMS-to-RIM interface was developed at the EROS Data Center. The data flow was designed using VDI tools and was implemented into a smooth flow using VAX DCL. Existing tools used in the interface include RIM, RIMIN, RIMSEQ, RIMDUP, RIMAPD, RIMNET, and RIMQLT. Tools designed specifically for the AMS interface include AMSREG, RIMSCA, RIMRNG, and RIMBND. An AMS data set containing the A-E files is required as input and those files must be named amsa.dat, amsb.dat, amsc.dat, amsd.dat, and amse.dat. The AMS data files are loaded into RIM where they are decomposed into the ARCXYS, ARCHDR, and PRIMATT relations needed for structuring into the complete vector interface data structure. The resultant coordinates are in map-block-centered lat/long coordinates that require further transformation prior to conversion into another map projection. The procedure currently assumes that the data were digitized using a Data General 16-bit computer. The AMS-to-RIM interface is expected to have extensive usage in the future. Only test data, including the Kenai data set, have been sent through the interface to date.

D.37 VDI(RIM) from ARC/INFO

The ARC/INFO-to-RIM interface was developed at the EROS Data Center. The flow is implemented in VAX DCL. Existing tools used were RIM and available ARC/INFO command files. Tools designed specifically for the ARC/INFO interface are ARCRIM and CRESCH. The procedure can be submitted from inside of ARC/INFO as well as from VMS.

Input is an ARC/INFO coverage. The current procedure generates the following set of relations for each data type:

1. POINT: POINTXY and replicates all INFO files as relations
2. LINE: ARCHDR, ARCXYs and replicates all INFO files as relations
3. REGION: ARCXYs, ARCHDR, LABXY, and replicates all INFO files as relations

The ARCHDR relation should be removed prior to structuring the data into the complete vector interface data structure. Several projects including FMLIS and SLAP have been sent through this interface. The procedure is frequently used.

D.38 VDI(RIM) from DLG

The DLG-to-RIM standard format interface was developed at the EROS Data Center. The flow is implemented in TAE Command Language and consists of three parts: a copy from tape to VMS file step (DLGDISC), a DLG load procedure in which the RIM data base is defined and the data is loaded into the vector data structure (DLGDBS), and a transformation from table coordinates to projection coordinates (DLGAFFINE). The VDI relations created are defined in the TAE documentation of the DLG data entry procedures. Several projects including Fort Berthold have used this interface. It is anticipated that the procedure will be used frequently.

D.39 VDI(RIM) from DLG OPTIONAL

The DLG OPTIONAL-to-RIM interface is under consideration for implementation under TAE whenever the demand will justify the development.

D.40 VDI(RIM) from FGEF

The Federal Interagency Coordinating Committee on Digital Cartography (FICCDC) Standards Working Group is in the process of developing the Federal Geographic Exchange Format (FGEF). It is planned that an interface will be developed to support data entry into VDI(RIM).

D.41 VDI(RIM) from GES

The GES-to-RIM interface was developed at the EROS Data Center. The data flow was designed using VDI tools and was implemented into a smooth flow using VAX DCL. Input is a GES GEOBLOCK, and output is a single complex RIM database containing all GES overlays. These overlays then need to be moved to separate RIM data bases where standard relational naming conventions are used. Most vector data entry for application project support uses this interface. Several projects have utilized this capability for converting data between GES and ARC/INFO and between GES and MOSS.

D.42 VDI(RIM) from INTERGRAPH

The INTERGRAPH-to-RIM interface is under development at the EROS Data Center under TAE. Input is an Intergraph design file. The current procedure generates the following set of relations for each data type:

POINT : POINTXY and PRIMATT
LINE : ARCHDR, ARCXYs and PRIMATT
REGION : ARCXYs, LABXY, and PRIMATT

A basic capability exists. The interface is anticipated to be in frequent use if the INTERGRAPH system becomes the primary data collection system for application projects.

D.43 VDI(RIM) from LAS

The LAS-to-RIM interface was developed at the EROS Data Center. The raster-to-vector interface for region data (ICARAS) is based on the algorithm developed by Azriel Rosenfield and implemented by Dave Nichols. A raster-to-vector interface for point data (RTP) is also available. The interfaces are implemented under TAE and read a LAS image. The capability for converting line data sets does not exist within a user environment at the EROS Data Center.

Input is LAS raster data set. The current procedure generates the following set of relations for each data type:

POINT : POINTXY and PRIMATT
REGION : ARCXYs, ARCHDR and PRIMATT

It is anticipated that the interface will be used frequently.

D.44 VDI(RIM) from LUNR

The LUNR-to-RIM interface was developed at the EROS Data Center within RIM. The interface was designed for low frequency use. Input is a LUNR data set and output data set consists of ARCXYs, PRIMATT, and LABXY relations.

E Output Interfaces

E.1 ADS to DLG OPTIONAL

The Bureau of Land Management (BLM), Denver, developed the ADSTODLG-optional interface. The EROS Data Center has attempted to use this interface but has not acquired the ADSTODLG-optional software.

It is unlikely that the EROS Data Center will pursue this capability due to the limitations (999 polygons max., misplaced labels, limit of single attribute per polygon) of the ADS system.

E.2 ADS to MAP

The ADS (Automated Digitizing System) was developed by the Bureau of Land Management. Development work was begun by Bob Green and was continued by Greg Hand, who has since left the BLM. The interface from ADS to a physical map is a hardcopy mapmaking process.

Data from ADS are normally used as input to MOSS. The system features spaghetti digitizing, automatic node creation, automatic polygon creation, specialized menus with different line types, and has a plotting capability. The interface is a standalone program (for example, is part of ADS). The EROS Data Center has not used ADS, but will be acquiring the system, and may use it for minor tasks.

There is a limit of 999 polygons per map. ADS does not handle multiple attributes. Misplaced labels apparently present a problem.

E.3 ADS to MOSS IMPORT/EXPORT

The ADS to MOSS IMPORT/EXPORT interface does not exist at the EROS Data Center. It was developed by the BLM to allow ADS digitized data to be used in MOSS. A text file that is in the MOSS IMPORT/EXPORT format may be read in via the MOSS ADD command. EXPORT refers to the MOSS command EXPORT, which allows the creation of a text file from an active MOSS map. The file can then be used as input to another GIS system or to another computer.

E.4 AGIS/GRAM to MAP

The primary level graphics menu offers hardcopy options for the current contents of the graphic display. Two devices are software supported, a pen plotter and an electrostatic printer/plotter. Other devices can be implemented with driver routines for which a template is provided. The user has control of the scale of the output.

A separate plot menu supports creation of index and title blocks, projection, data types and plot devices for producing hardcopy from all or a framed portion of the map database.

E.5 AHDS to MOSS

The Alaska Hydrologic Digitizing System, which was designed primarily for the digitizing of surface water features, hydrography, and lakes, has an input interface to MOSS, but the status is unknown.

E.6 AMS to MAP

The AMS to MAP interface (AMS plotting system) is part of the AMS system, primarily a means of providing verification plots at scale. It is used for this purpose regularly by TGS/CMD in Ft. Collins, Colo. The plotting of attributes is a time-intensive task for the AMS plotting program. Therefore, verification of attributes is done mostly by comparing text listings.

E.7 ARC/INFO to ARC/INFO IMPORT/EXPORT

ESRI provides a two-way interface to other ARC/INFO systems using the Import/Export format. This format was introduced in release 3.1, in order to bypass the problems of incomplete exchange of attributes and topology which occur through the use of DLG-3 interfaces. As with the DLG-3 optional, it uses all ASCII encoding, which minimizes machine-to-machine conversion problems. In addition, it converts all ARC/INFO coverage files to output and input, just as they exist, with the only conversion being to ASCII codes. It has been used frequently, especially for Vax-to-Prime moves, and is a smooth flow interface. No problems have been encountered and ESRI maintains it as part of the ARC/INFO system (no source code provided).

E.8 ARC/INFO to DLG OPTIONAL

This interface is built into ESRI's ARC/INFO software, and is maintained by ESRI. It was designed as the original interface format for ARC/INFO; later, the IMPORT/EXPORT format was added to treat DLG-3 deficiencies. The program outputs points, lines, and areas, with a unique "user-id" attribute key. No other attributes are carried with the file, and must be output in a separate ancillary file. No topology records are written. The program is written in Fortran 77, and source is supplied by ESRI. We have used the program very infrequently, especially since the IMPORT/EXPORT format was developed for ARC/INFO to ARC/INFO transfers. The program represents a smooth flow for the data transferred, as long as attribute recreation and topological structuring can be performed on the receiving system.

E.9 ARC/INFO to GIMMS

ESRI provides an unsupported capability for ARC/INFO to product input for GIMMS

E.10 ARC/INFO to MAP

ARC/INFO provides the ARCPLOT subsystem for map production. It is an integral part of the ARC/INFO system, and supports Tektronix displays (and emulators) as well as electrostatic and

pen plotters. The subsystem represents a smooth flow capability, is written in Fortran 77, and is maintained by ESRI (no source supplied). ARCPLOT outputs a "meta-file" which consists of an x,y, and pencode parameters. This allows the file to be displayed and/or plotted and saved as a reusable file. It is flexible and capable, rather than simple and friendly, but is used heavily.

E.11 ARC/INFO to PIOS

ESRI has developed an interface from ARC/INFO to its predecessor system, PIOS. The state of this software is unknown, since we have not used PIOS at the EROS Data Center, and have never attempted an interface. It is maintained by ESRI for users of PIOS, but is probably being used less and less, since few PIOS installations remain.

E.12 ARC/INFO to SVF

The Single Variable File (SVF) interface can be performed from ARC/INFO using the POLYGRID program, which converts arc vectors to raster data and adds necessary header data for the SVF. The SVF is fairly versatile for topographic data (its principal purpose), but is somewhat restrictive for remotely sensed data (multiple bands or variables), or extended range data (long integer or real), and may be wasteful for small integers (byte 0-255). This interface is a smooth flow, and except for the line and sample header record, can be used as a "transfer format" input. We have used the interface with moderate frequency, and have found no problems or limitations. It is written in Fortran 77 and is maintained by ESRI as part of the ARC/INFO system.

E.13 ARC/INFO to VDI(RIM)

Reference section RIM from ARC/INFO

E.14 CRAIGES to UCLGES UPDATE

The responsible agency is the USGS National Mapping Division, Reston, Va.

The Geological Survey has a need to convert graphic data from various categorical map overlays into DLG form. CRAIGES was developed to create interactively the digital data processed by the UCLGES software. The interface is required to transfer digitized data from CRAIGES to UCLGES.

The interface can be classified as a smooth flow interface.

1. Files produced - an ALTEK raw file that is input to the UCLGES UPDATE program.
2. Attributes - All DLG base categories.
3. Data types - Areas, lines, nodes, and labels.

The interface is a FORTRAN program running on Intergraph systems in each regional mapping center of the National Mapping Division. Usage of interface is heavy. Sample projects are: USGS

DLG 1:24,000-, 1:100,000-, 1:2,000,000- scale mapping programs. A new version is being developed at the EROS Data Center to eliminate deficiencies in the system. The failure rate of transmissions of data is not related to CRAIGES itself, but results from the DATAPOINT communications between the mapping centers and USGS Headquarters.

The system has been in place for several years. The 1:2,000,000 mapping program has used it extensively.

E.15 DLG to DLG OPTIONAL

The responsible agency is the USGS National Mapping Division, Reston, Va.

The National Mapping Division has received numerous requests to distribute DLG data in a more usable form; hence the DLG optional format was created. Topological relations are stored implicitly in DLG standard format. The implicit topology is read by the DLG standard to DLG optional software and explicit topology is generated and stored in the output file.

The classification of the interface is a smooth flow interface.

1. Files produced - An 80-byte card image ASCII file.
2. Attributes - All base category attributes as defined in USGS Circular 895.
3. Data types - Control points, conformal transformation parameters, descriptive information, node, area, and line topology.

The interface is a FORTRAN program running on the Amdahl 470 V7 at the USGS National Center in Reston, Va. Usage to date of interface is heavy. Sample projects are: USGS DLG 1:24,000-, 1:100,000-, 1:2,000,000-scale mapping programs.

The software has one problem when reading edited standard DLG's. If a DLG has been edited with line identification numbers out of sequence, a renumbering of the line identifiers is not carried through to the DLG optional format. In a few cases, incorrect topology is generated in the output file.

E.16 EXTELE to MAP

The responsible agencies are the USGS National Mapping Division, Reston, Va., the Rocky Mountain Mapping Center, Denver Colo., and the EROS Data Center.

The EROS Data Center and Rocky Mountain Mapping Center have requirements to plot Intergraph design files on either the Calcomp or Gerber plotters. Software was written to create an intermediate file from an Intergraph design file. This file is read by a plotter application program. The interface is a smooth flow interface.

1. Files produced - An EXTELE-formatted binary file.
2. Attributes - Line color and weight.

3. Data types - Text, text nodes, and all digitized linework.

The interface is a FORTRAN program running on Intergraph systems in each regional mapping center and the EROS Data Center. Usage of interface is heavy. Sample projects are: USGS DLG 1:24,000-, 1:100,000-, 1:2,000,000-scale mapping programs. A new version has been developed at the EROS Data Center to eliminate deficiencies.

The system has been in place for several years. Numerous proof plots and map collars have been created from EXTELE files.

E.17 GES to DLG OPTIONAL

As a part of GES, ESL can provide DLG3OUT which produces vector output files in the DLG 80-character optional format. This program has been used at the Norton Air Force Base GMIS installation. DLG3OUT reads an ERIS file to produce major and minor codes.

The program is a smooth flow program and functions reliably.

E.18 GES to MAP

GES provides the PLOTGES subsystem for plots of GES data items. This subsystem currently supports six display terminal types and six different plotters. All parameters required by the plotting software to set up the supported plotter and output data are contained in a plot transformation record that is saved for reuse. The PLOTGES subsystem can be run interactively or in a batch mode. It is simple and user friendly but not as flexible as ARC/PLOT in ARC/INFO.

E.19 GES to VDI(RIM)

The GES (Geographic Entry System) to RIM (Relational Information Management) interface was initially developed at the EROS Data Center as a prototype for using a relational data base for vector data interchange. The prototype was further refined and this vector data pathway now provides a means to move GES data to RIM and then to a variety of vector data processing systems.

The interface is a smooth flow interface.

1. Files produced - RIM relational data base files.
2. Attributes - All numeric attribute codes entered in GES.
3. Data types - Areas, lines, and labels.

The interface consists of a set of FORTRAN programs running on a VAX 11/780. VAX DCL is used to tie the programs together, as well as for the user interface. May be run interactively or batch. Usage of interface is moderate.

A sample project is the FMLIS API project on State of Colorado. The software runs somewhat slowly, so an overnight batch run, when machine resources are not in demand, is usually preferred.

Control points sometimes do not transfer properly. Otherwise the system provides a faithful representation of all points, lines, and attribute codes collected by the GES system.

E.20 GIMMS to ARC/INFO

ESRI provides an unsupported capability for ARC/INFO to produce input for GIMMS

E.21 GIRAS to DLG

The responsible agency is the USGS National Mapping Division, Reston, Va.

The National Mapping Division has received numerous requests to distribute land use/land cover data in a more usable form. This led to the development of software to translate GIRAS-internal format to DLG format. DLG-optional is also available by the same interface described in section C.3.13. Topological relationships are built in GIRAS and copied into the DLG-standard format.

The interface is a smooth flow interface.

1. Files produced - A standard or optional DLG.
2. Attributes - All land use/land cover attributes as defined in USGS Circular 895.
3. Data types - Control points, conformal transformation parameters, descriptive information, node, area, and line topology.

The interface is a FORTRAN program running on the Amdahl 470 V7 at the USGS National Center in Reston Va. Usage of the interface is heavy. A sample project is the USGS LUDA (Land Use Data Analysis) program. The software has had no problems in creating the DLG.

E.22 INTERGRAPH to DLG

The responsible agencies are the USGS National Mapping Division, Reston, Va. and the EROS Data Center. CRAIGES was developed to interactively create the digital data processed by the UCLGES software. EROS has enhanced the CRAIGES to evolve into EDCIDS which has more capability and flexibility. One of the many outputs of EDCIDS is an ALTEK raw tape for input into UCLGES UPDATE. The interface is smooth flow interface.

1. Files produced - An ALTEK raw file which is input to the UCLGES UPDATE program.
2. Attributes - All DLG base categories.
3. Data types - Areas, lines, nodes, and labels.

EDCIDS is a FORTRAN program running on an Intergraph system at the EROS Data Center. Usage of the interface is heavy. A sample project is the USGS DLG 1:100,000-scale mapping program. The software is a great improvement in digitizing time and efficiency over the older CRAIGES software.

Data problems noted in previous transfers: Failure rate of transmissions of data has nothing to do with EDCIDS itself, but with the DATAPOINT communications between EROS and the USGS Headquarters in Reston. The system has been in place for nearly 1 year with significant production of DLG's.

E.23 INTERGRAPH to EXTELE

The agencies responsible for the interface are the Rocky Mountain Regional Mapping Center (RMMC) and the EROS Data Center.

The EROS Data Center and Rocky Mountain Mapping Center have requirements to plot Intergraph design files on either the Calcomp or Gerber plotters. Software was written to create an intermediate file from an Intergraph design file. This file is read by several plotter application programs. The interfaces are particular drivers which read the EXTELE file and produce plot files for each plotting device. The interfaces are smooth flows. The files produced are:

1. A Gerber, Calcomp, Versatec, or Applicon plot file.
2. Attributes - Line color and weight.
3. Data types - Text, text nodes, and all digitized line work.

The various plotter application programs are:

1. Calcplot - FORTRAN program running on Intergraph PDP 11/23 at the EROS Data Center.
2. Gerber - FORTRAN program running on Intergraph PDP 11/70 at the Rocky Mountain Mapping Center.
3. RIMQLT - FORTRAN program that reads relational format converted from EXTELE file and plots on Versatec electrostatic black and white plotter.
4. ARCAP - FORTRAN program that reads relational format converted from EXTELE file and plots on Applicon color electrostatic plotter.

Usage is heavy. Sample projects are: USGS DLG 1:24,000-, 1:100,000-, 1:2,000,000-scale mapping programs. Each plotter program has been improved regularly and has become more efficient. The programs have been in place for several years. Numerous proof plots and map collars have been created from EXTELE files.

E.24 INTERGRAPH to VDI(RIM)

The responsible agency for the interfacing is the EROS Data Center.

The EROS Data Center has been involved in Vector Data Interface research. Several vector processing systems were designated initially to test the validity of VDI tools. The Intergraph was selected as one of the important systems to create an interface to and from RIM.

The interface is smooth flow interface.

1. Files produced - User-defined linkage file, arc output file, label output, node output file, point output file, and an attribute output file.
2. Attributes - Any coded or text attribute.
3. Data types - Areas, lines, nodes, labels, and attributes.

The interface consists of several FORTRAN programs running on a PDP 11/23. The programs are executed by running indirect command files on the PDP from MCR and may be run interactively or batch. Usage of the interface is slight to moderate.

A sample project is the AVHRR mosaic of the western United States. The USGS 1:2,000,000-scale DLG's were edited and transferred to RIM for rasterization and registration to the AVHRR imagery. The software runs slowly for large data sets since the PDP 11/23 is a slow machine. The system is fully documented but not rigorously tested in a full production mode.

E.25 MOSS to COS

COS (Cartographic Output System) can receive data from MOSS. This interface is used frequently and is described in the system description on MOSS and COS.

E.26 MOSS to MAP

The generation a map can be accomplished by the MOSS plotter function (CALCOM, HOUSTON, etc) or by the COS subsystem.

E.27 MOSS to MAPS

This was implemented as a MAPS subsystem function within MOSS by Autometrics. Execution of the function is possible from the MAPS subsession with a minimum set of arguments.

The user must specify the vector input map name, the desired cell output map name, and the desired cellularization by either specifying the width and height of the output cell or the area and aspect ratio for the cell. The session viewing window controls the spatial coverage of the operation with respect to the input map.

E.28 MOSS to MOSS IMPORT/EXPORT

This conversion is implemented as a MOSS function by Autometrics. Execution of the function is possible from a MOSS session with user-prompted parameters, hence the conversion is termed smooth flow. The MOSS data set to be exported must be an active session data set. The user must respond to the function prompts with the active data set number and the number of subjects to be exported.

An export file is created in the user's workspace and will supersede any previous export file. Any export file to be saved must be renamed or copied to an alternate directory or media.

Within any project on a particular MOSS hardware/software implementation the use of EXPORT would likely be infrequent. The most probable use is to transfer data between MOSS implementations on dissimilar hardware or to an alternate GIS.

E.29 TIGER to DLG

Data that have been tagged by Census and are in one of the NDCDB categories are sent back to the NMD in a DLG format. However, later phases of the project where additional information (features) will be defined cannot be supported with the DLG format. It is anticipated that the FGEF standard will be used.

E.30 UCLGES to DLG

The responsible agency for the interface is the National Mapping Division, Reston, Va.

DLG's have been produced by the National Mapping Division since the mid-1970's. A system called the Unified Coding of Line Graph Element System (UCLGES) was developed to produce the DLG from digitizer data.

UCLGES runs on an Amdahl mainframe computer at National Mapping Division Headquarters. The original DLG was stored in IBM binary format. DLG's are currently distributed in ASCII for interchange with various computer systems. The interface is smooth flow interface.

1. Files produced - A 144-byte record ASCII DLG.
2. Attributes - All DLG base category attributes.
3. Data types - Areas, lines, nodes, degenerate lines, and attributes.

The interface consists of a FORTRAN program (BINASC) executed by IBM Job Control Language. The program is normally run in a batch queue. The usage of the interface is heavy since all DLG are archived in IBM binary format. Customer orders are processed by converting the binary DLG to standard ASCII format.

Sample projects are the USGS 1:2,000,000-, 1:100,000-, and 1:24,000-scale national mapping programs. The software has been converted to run on Gould 9780 minicomputers under a new system entitled PROSYS.

E.31 VDI(RIM) to AGIS/GRAM

The RIM-to-AGIS/GRAM interface was developed to support a contract with Interactive Systems Corporation. Working from a preliminary definition of VDI structure within RIM, the algorithms permit point, line, and region (polygon) data to be loaded into AGIS in interactive or batch operations. Construction of VAX DCL procedures to prepare the AGIS-required format of VDI from the currently defined format of VDI have been developed.

The expectation of the AGIS loading routines for VDI data is as follows:

1. POINTS- Relations: COORDS with attributes NODENUM, LAT, LONG; ATTRIBTS with attributes NODENUM, LAYNUM, CLSNUM, SYMBOL, SUBJECT.
2. LINES- Relations: COORDS with attributes ARCNUM, LAT, LONG; ATTRIBTS with attributes ARCNUM, LAYNUM, CLSNUM, SUBJECT.
3. REGIONS- Relations: COORDS with attributes ARCNUM, LAT, LONG; ATTRIBTS with attributes AREANUM, LAYNUM, CLSNUM, SUBJECT; ARCHDR with attributes ARCNUM, LFTAREA, RGTAREA; AREAARC with attributes AREANUM, ISLAND, ARCNUM.

The process of adjusting point and line data from the current VDI format to the AGIS/GRAM format is relatively easy. The region data preparation is more extensive utilizing such tools as RIMSEQ, RIMDUP, CTSX, and RIMVDI. Since AGIS/GRAM is an experimental system without extensive project-oriented data loading requirements, these procedures have not been streamlined for the general user.

E.32 VDI(RIM) to ARC/INFO

The RIM-to-ARC/INFO interface was developed at the EROS Data Center. The flow is implemented in VAX DCL. Existing tools used were RIM and available ARC/INFO command files. Tools designed specifically for the ARC/INFO interface are RIMARC, SORTATT, and CREATT.

Input is from the following set of relations for each data type:

POINT : POINTXY and PRIMATT
LINE : ARCXYs and PRIMATT
REGION : ARCXYs, LABXY

The procedure generates an ARC/INFO coverage, a TIC relation containing a set of tic marks, and a BND relation containing min/max values will be transferred, if available. A complete ARC/INFO coverage is created. The procedure can be submitted from inside of ARC/INFO as well as from VMS. Several projects have made use of this interface, which is frequently used.

E.33 VDI(RIM) to DLG OPTIONAL

The RIM-DLG format is under consideration for implementation under TAE if a demand becomes evident.

E.34 VDI(RIM) to FGEF

The Federal Interagency Coordinating Committee on Digital Cartography (FICCDC) Standards Working Group is in the process of developing the Federal Geographic Exchange Format (FGEF). It is planned that an interface will be developed to convert data from VDI(RIM) into the FGEF standard.

E.35 VDI(RIM) to Intergraph

The RIM-to-Intergraph is under development at the EROS Data Center. The interface is being developed under TAE. A basic capability exists. Input is the following set of relations for each data type:

POINT : POINTXY and PRIMATT
LINE : ARCXYs and PRIMATT
REGION : ARCXYs, LABXY, and PRIMATT

The procedure generates an Intergraph design file.

The interface is anticipated to be in frequent use if the Intergraph system becomes the primary data collection system for application projects.

E.36 VDI(RIM) to LAS

The RIM-to-LAS interface was developed at the EROS Data Center. The vector-to-raster (VTR) interface is based on the algorithm developed by Mitch Modeleski. The interface is implemented under TAE and writes a LAS image. The capability exists for converting point, line, and region data sets.

Input is from the following set of relations for each data type:

POINT : POINTXY and POINTWGT
LINE : ARCXYs and ARCWGT
REGION : ARCXYs and ARCHDR

The procedure generates a LAS raster data set. The interface is anticipated to be in frequent use.

E.37 VDI(RIM) to MAP

Mapmaking within RIM is possible using either the Versatec electrostatic printer (RIMQLT) or the Applicon ink-jet plotter. RIMQLT requires a display relation containing an identification- attribute, x-coordinate, y-coordinate, and optional attributes that can be displayed as labels. Regions must be defined as full regions, and must have each polygon closed upon itself. Points, chains, and regions can be intermixed in the display relation. Versatec and Applicon procedures are frequently used. Both are still under development.

E.38 VDI(RIM) to MOSS

The RIM-to-MOSS interface was developed at the EROS Data Center and is implemented as a program under VAX VMS (RIMMOSS).

Input is from the following set of relations for each data type:

POINT : *POINTXY* and *PRIMATT*
LINE : *ARCXYS* and *PRIMATT*
REGION : *REGXY* and *PRIMATT*

The procedure generates a MOSS ADD file.

The creation of multiple attribute files is not directly supported, although they could be supplied. The Sandcrane data (BOR) and the ARP data have been sent through this conversion. The interface is anticipated to be in moderately frequent use.

F Data Flows

The concept of a data flow was presented in section 2.1 as a communication involving more than one interface. Interfacing two systems through an intermediate system—the approach recommended by the study—therefore creates data flows between the source and target system, through the intermediate system. In addition to the VDI system developed at the EROS Data Center, several other systems or data structures can act as conversion hubs, such as the DLG data structures. With two or more of these central systems, the reality of system connections becomes a network with a large number of possible data flows.

Tables F1 and F2 present data flows that might be possible, according to the interfaces presented earlier. For completeness, the basic interfaces are also listed.

The data flows in table F1 are arranged alphabetically by the originating data structure or system; the data flows in table F2 are ordered by the destination data structure or system. Table F2 is restricted to first priority systems as indicated in table 4 of the main report.

The following codes indicate the status of the individual interfaces. These codes are also shown in each data flow, so that the potential for a given data flow can be rapidly assessed.

Table F0: Status Codes

| CODE | DESCRIPTION | COST |
|------|--------------|------|
| S | smooth flow | 1 |
| T | tool bench | 5 |
| D | digitizing | 100 |
| M | hardcopy map | 1 |
| O | not in-house | 10 |
| P | planned | 20 |
| U | unknown | 30 |
| N | non-existing | 9999 |

F.1 Data Flows by Input Data Structure

ADS-M->MAP
ADS-O->DLG OPTIONAL
ADS-O->DLG OPTIONAL-S->ARC/INFO
ADS-O->DLG OPTIONAL-S->ARC/INFO-S->ARC IMP/EXP
ADS-O->DLG OPTIONAL-S->ARC/INFO-S->GIMMS
ADS-O->DLG OPTIONAL-S->ARC/INFO-S->PIOS
ADS-O->DLG OPTIONAL-S->ARC/INFO-S->SVF
ADS-O->DLG OPTIONAL-S->GES
ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)
ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-P->FGEF
ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-S->AGIS/GRAM
ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-S->LAS
ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH
ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG

ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE
 ADS-O->MOSS IMP/EXP
 ADS-O->MOSS IMP/EXP-S->MOSS
 ADS-O->MOSS IMP/EXP-S->MOSS-S->COS
 ADS-O->MOSS IMP/EXP-S->MOSS-S->MAPS

AGIS/GRAM-S->MAP

AHDS-U->MOSS
 AHDS-U->MOSS-M->MAP
 AHDS-U->MOSS-S->COS
 AHDS-U->MOSS-S->MAPS
 AHDS-U->MOSS-S->MOSS IMP/EXP

ALMRS-P->VDI(RIM)
 ALMRS-P->VDI(RIM)-M->MAP
 ALMRS-P->VDI(RIM)-P->FGEF
 ALMRS-P->VDI(RIM)-S->AGIS/GRAM
 ALMRS-P->VDI(RIM)-S->ARC/INFO
 ALMRS-P->VDI(RIM)-S->ARC/INFO-S->ARC IMP/EXP
 ALMRS-P->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 ALMRS-P->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 ALMRS-P->VDI(RIM)-S->ARC/INFO-S->GIMMS
 ALMRS-P->VDI(RIM)-S->ARC/INFO-S->PIOS
 ALMRS-P->VDI(RIM)-S->ARC/INFO-S->SVF
 ALMRS-P->VDI(RIM)-S->LAS
 ALMRS-P->VDI(RIM)-S->MOSS
 ALMRS-P->VDI(RIM)-S->MOSS-S->COS
 ALMRS-P->VDI(RIM)-S->MOSS-S->MAPS
 ALMRS-P->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP
 ALMRS-P->VDI(RIM)-T->INTERGRAPH
 ALMRS-P->VDI(RIM)-T->INTERGRAPH-S->DLG
 ALMRS-P->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 ALMRS-P->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 ALMRS-P->VDI(RIM)-T->INTERGRAPH-S->EXTELE

AMS-M->MAP
 AMS-O->DFAD
 AMS-O->DLG
 AMS-O->DLG-O->DCASS
 AMS-O->DLG-O->TIGER
 AMS-S->MOSS
 AMS-S->MOSS-S->COS
 AMS-S->MOSS-S->MAPS
 AMS-S->MOSS-S->MOSS IMP/EXP
 AMS-T->VDI(RIM)

AMS-T->VDI(RIM)-P->FGEF
 AMS-T->VDI(RIM)-S->AGIS/GRAM
 AMS-T->VDI(RIM)-S->ARC/INFO
 AMS-T->VDI(RIM)-S->ARC/INFO-S->ARC IMP/EXP
 AMS-T->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 AMS-T->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 AMS-T->VDI(RIM)-S->ARC/INFO-S->GIMMS
 AMS-T->VDI(RIM)-S->ARC/INFO-S->PIOS
 AMS-T->VDI(RIM)-S->ARC/INFO-S->SVF
 AMS-T->VDI(RIM)-S->LAS
 AMS-T->VDI(RIM)-T->INTERGRAPH
 AMS-T->VDI(RIM)-T->INTERGRAPH-S->EXTELE

ARC IMP/EXP-S->ARC/INFO
 ARC IMP/EXP-S->ARC/INFO-M->MAP
 ARC IMP/EXP-S->ARC/INFO-S->DLG OPTIONAL
 ARC IMP/EXP-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 ARC IMP/EXP-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
 ARC IMP/EXP-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->COS
 ARC IMP/EXP-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MAPS
 ARC IMP/EXP-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MOSS IMP/EXP
 ARC IMP/EXP-S->ARC/INFO-S->GIMMS
 ARC IMP/EXP-S->ARC/INFO-S->PIOS
 ARC IMP/EXP-S->ARC/INFO-S->SVF
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-P->FGEF
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-S->LAS
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE

ARC/INFO-M->MAP
 ARC/INFO-S->ARC IMP/EXP
 ARC/INFO-S->DLG OPTIONAL
 ARC/INFO-S->DLG OPTIONAL-S->GES
 ARC/INFO-S->DLG OPTIONAL-S->MOSS
 ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->COS
 ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MAPS
 ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MOSS IMP/EXP
 ARC/INFO-S->GIMMS
 ARC/INFO-S->PIOS
 ARC/INFO-S->SVF
 ARC/INFO-S->VDI(RIM)
 ARC/INFO-S->VDI(RIM)-P->FGEF
 ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM

ARC/INFO-S->VDI(RIM)-S->LAS
 ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
 ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE

 CONTRACTOR FMT-O->GIRAS
 CONTRACTOR FMT-O->GIRAS-O->DLG
 CONTRACTOR FMT-O->GIRAS-O->DL-O->DCASS
 CONTRACTOR FMT-O->GIRAS-O->DLG-O->TIGER
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->ARC/INFO
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->ARC/INFO-S->ARC IMP/EXP
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->ARC/INFO-S->DLG OPTIONAL
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->ARC/INFO-S->GIMMS
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->ARC/INFO-S->PIOS
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->ARC/INFO-S->SVF
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->GES
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->GES-M->MAP
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->INTERGRAPH
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->INTERGRAPH-S->EXTELE
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)-P->FGEF
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)-S->AGIS/GRAM
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)-S->LAS
 CONTRACTOR FM--O->GIRAS-O->DLG-S->VDI(RIM)-S->MOSS
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)-S->MOSS-S->COS
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)-S->MOSS-S->MAPS
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP

 CRAIGES-S->UCLGES UPDATE
 CRAIGES-S->UCLGES UPDATE-S->UCLGES
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-O->DCASS
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-O->TIGER
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->ARC IMP/EXP
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->GIMMS
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->PIOS
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->SVF
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->GES
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->GES-M->MAP
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->INTERGRAPH
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->INTERGRAPH-S->EXTELE
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-P->FGEF
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->AGIS/GRAM

CRAIGES-S->UCLGES UPDATE-S->UCLGES-->DLG-S->VDI(RIM)-S->LAS
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS-S->COS
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS-S->MAPS
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP

DCASS-M->MAP

DIME-S->ARC/INFO
 DIME-S->ARC/INFO-M->MAP
 DIME-S->ARC/INFO-S->ARC IMP/EXP
 DIME-S->ARC/INFO-S->DLG OPTIONAL
 DIME-S->ARC/NFO-S->DLG OPTIONAL-S->GES
 DIME-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
 DIME-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->COS
 DIME-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MAPS
 DIME-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MOSS IMP/EXP
 DIME-S->ARC/INFO-S->GIMMS
 DIME-S->ARC/INFO-S->PIOS
 DIME-S->ARC/INFO-S->SVF
 DIME-S->ARC/INFO-S->VDI(RIM)
 DIME-S->ARC/INFO-S->VDI(RIM)-P->FGEF
 DIME-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
 DIME-S->ARC/INFO-S->VDI(RIM)-S->LAS
 DIME-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
 DIME-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 DIME-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 DIME-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 DIME-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE

DLG OPTIONAL-S->ARC/INFO
 DL OPTIONAL-S->ARC/INFO-S->ARC IMP/EXP
 DLG OPTIONAL-S->ARC/INFO-S->GIMMS
 DLG OPTIONAL-S->ARC/INFO-S->PIOS
 DLG OPTIONAL-S->ARC/INFO-S->SVF
 DLG OPTIONAL-S->GES
 DLG OPTIONAL-S->GES-M->MAP
 DLG OPTIONAL-S->GES-S->VDI(RIM)
 DLG OPTIONAL-S->GES-S->VDI(RIM)-P->FGEF
 DLG OPTIONAL-S->GES-S->VDI(RIM)-S->AGIS/GRAM
 DLG OPTIONAL-S->GES-S->VDI(RIM)-S->LAS
 DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH
 DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE
 DLG OPTIONAL-S->MOSS
 DLG OPTIONAL-S->MOSS-S->COS

DLG OPTIONAL-S->MOSS-S->MAPS
DLG OPTIONAL-S->MOSS-S->MOSS IMP/EXP

DLG-O->DCASS
DLG-O->TIGER
DLG-S->ARC/INFO
DLG-S->ARC/INFO-S->ARC IMP/EXP
DLG-S->ARC/INFO-S->DLG OPTIONAL
DLG-S->ARC/INFO-S->GIMMS
DLG-S->ARC/INFO-S->PIOS
DLG-S->ARC/INFO-S->SVF
DLG-S->GES
DLG-S->GES-M->MAP
DLG-S->INTERGRAPH
DLG-S->INTERGRAPH-S->EXTELE
DLG-S->VDI(RIM)
DLG-S->VDI(RIM)-P->FGEF
DLG-S->VDI(RIM)-S->AGIS/GRAM
DLG-S->VDI(RIM)-S->LAS
DLG-S->VDI(RIM)-S->MOSS
DLG-S->VDI(RIM)-S->MOSS-S->COS
DLG-S->VDI(RIM)-S->MOSS-S->MAPS
DLG-S->VDI(RIM)-S->MOSS-S-MOSS IMP/EXP

EXTELE-M->MAP

FGEF-P->VDI(RIM)
FGEF-P->VDI(RIM)-M->MAP
FGEF-P->VDI(RIM)-S->AGIS/GRAM
FGEF-P->VDI(RIM)-S->ARC/INFO
FGEF-P->VDI(RIM)-S->ARC/INFO-S->ARC IMP/EXP
FGEF-P->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
FGEF-P->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
FGEF-P->VDI(RIM)-S->ARC/INFO-S->GIMMS
FGEF-P->VDI(RIM)-S->ARC/INFO-S->PIOS
FGEF-P->VDI(RIM)-S->ARC/INFO-S->SVF
FGEF-P->VDI(RIM)-S->LAS
FGEF-P->VDI(RIM)-S->MOSS
FGEF-P->VDI(RIM)-S->MOSS-S->COS
FGEF-P->VDI(RIM)-S->MOSS-S->MAPS
FGEF-P->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP
FGEF-P->VDI(RIM)-T->INTERGRAPH
FGEF-P->VDI(RIM)-T->INTERGRAPH-S->DLG
FGEF-P->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
FGEF-P->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
FGEF-P->VDI(RIM)-T->INTERGRAPH-S->EXTELE

GES-M->MAP

GES-S->VDI(RIM)
 GES-S->VDI(RIM)-P->FGEF
 GES-S->VDI(RIM)-S->AGIS/GRAM
 GES-S->VDI(RIM)-S->ARC/INFO
 GES-S->VDI(RIM)-S->ARC/INFO-S->ARC IMP/EXP
 GES-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 GES-S->VDI(RIM)-S->ARC/INFO-S->GIMMS
 GES-S->VDI(RIM)-S->ARC/INFO-S->PIOS
 GES-S->VDI(RIM)-S->ARC/INFO-S->SVF
 GES-S->VDI(RIM)-S->LAS
 GES-S->VDI(RIM)-S->MOSS
 GES-S->VDI(RIM)-S->MOSS-S->COS
 GES-S->VDI(RIM)-S->MOSS-S->MAPS
 GES-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP
 GES-S->VDI(RIM)-T->INTERGRAPH
 GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 GES-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE

GIMMS-S->ARC/INFO
 GIMMS-S->ARC/INFO-M->MAP
 GIMMS-S->ARC/INFO-S->ARC IMP/EXP
 GIMMS-S->ARC/INFO-S->DLG OPTIONAL
 GIMMS-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 GIMMS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
 GIMMS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->COS
 GIMMS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MAPS
 GIMMS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MOSS IMP/EXP
 GIMMS-S->ARC/INFO-S->PIOS
 GIMMS-S->ARC/INFO-S->SVF
 GIMMS-S->ARC/INFO-S->VDI(RIM)
 GIMMS-S->ARC/INFO-S->VDI(RIM)-P->FGEF
 GIMMS-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
 GIMMS-S->ARC/INFO-S->VDI(RIM)-S->LAS
 GIMMS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
 GIMMS-S->ARC/INFO-S->VDI(RIM)-T->INTERGAPH-S->DLG
 GIMMS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 GIMMS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 GIMMS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE

GIRAS-O->DLG
 GIRAS-O->DLG-O->DCASS
 GIAS-O->DLG-O->TIGER
 GIRAS-O->DLG-S->ARC/INFO
 GIRAS-O->DLG-S->ARC/INFO-S->ARC IMP/EXP
 GIRAS-O->DLG-S->ARC/INFO-S->DLG OPTIONAL
 GIRAS-O->DLG-S->ARC/INFO-S->GIMMS

GIRAS-O->DLG-S->ARC/INFO-S->PIOS
 GIRAS-O->DLG-S->ARC/INFO-S->SVF
 GIRAS-O->DLG-S->GES
 GIRAS-O->DLG-S->GES-M->MAP
 GIRAS-O->DLG-S->INTERGRAPH
 GIRAS-O->DLG-S->INTERGRAPH-S->EXTELE
 GIRAS-O->DLG-S->VDI(RIM)
 GIRAS-O->DLG-S->VDI(RIM)-P->FGEF
 GIRAS-O->DLG-S->VDI(RIM)-S->AGIS/GRAM
 GIRAS-O->DLG-S->VDI(RIM)-S->LAS
 GIRAS-O->DLG-S->VDI(RIM)-S->MOSS
 GIRAS-O->DLG-S->VDI(RIM)-S->MOSS-S->COS
 GIRAS-O->DLG-S->VDI(RIM)-S->MOSS-S->MAPS
 GIRAS-O->DLG-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP

INTERGRAPH-S->DLG
 INTERGRAPH-S->DLG-O->DCASS
 INTERGRAPH-S->DLG-O->TIGER
 INTERGRAPH-S->DLG-S->ARC/INFO
 INTERGRAPH-S->DLG-S->ARC/INFO-S->ARC IMP/EXP
 INTERGRAPH-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
 INTERGRAPH-S->DLG-S->ARC/INFO-S->GIMMS
 INTERGRAPH-S->DLG-S->ARC/INFO-S->PIOS
 INTERGRAPH-S->DLG-S->ARC/INFO-S->SVF
 INTERGRAPH-S->DLG-S->GES
 INTERGRAPH-S->DLG-S->VDI(RIM)
 INTERGRAPH-S->DLG-S->VDI(RIM)-P->FGEF
 INTERGRAPH-S->DLG-S->VDI(RIM)-S->AGIS/GRAM
 INTERGRAPH-S->DLG-S->VDI(RIM)-S->LAS
 INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS
 INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS-S->COS
 INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS-S->MAPS
 INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP
 INTERGRAPH-S->EXTELE
 INTERGRAPH-S->EXTELE-M->MAP

LAS-S->VDI(RIM)
 LAS-S->VDI(RIM)-M->MAP
 LAS-S->VDI(RIM)-P->FGEF
 LAS-S->VDI(RIM)-S->AGIS/GRAM
 LAS-S->VDI(RIM)-S->ARC/INFO
 LAS-S->VDI(RIM)-S->ARC/INFO-S->ARC IMP/EXP
 LAS-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 LAS-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 LAS-S->VDI(RIM)-S->ARC/INFO-S->GIMMS
 LAS-S->VDI(RIM)-S->ARC/INFO-S->PIOS
 LAS-S->VDI(RIM)-S->ARC/INFO-S->SVF
 LAS-S->VDI(RIM)-S->MOSS

LAS-S->VDI(RIM)-S->MOSS-S->COS
 LAS-S->VDI(RIM)-S->MOSS-S->MAPS
 LAS-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP
 LAS-S->VDI(RIM)-T->INTERGRAPH
 LAS-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 LAS-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 LAS-S->VDI(RIM)-T->INERGRAPH-S->DLG-O->TIGER
 LAS-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE

LUNR-S->VDI(RIM)
 LUNR-S->VDI(RIM)-M->MAP
 LUNR-S->VDI(RIM)-P->FGEF
 LUNR-S->VDI(RIM)-S->AGIS/GRAM
 LUNR-S->VDI(RIM)-S->ARC/INFO
 LUNR-S->VDI(RIM)-S->ARC/INFO-S->ARC IMP/EXP
 LUNR-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 LUNR-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 LUNR-S->VDI(RIM)-S->ARC/INFO-S->GIMMS
 LUNR-S->VDI(RIM)-S->ARC/INFO-S->PIOS
 LUNR-S->VDI(RIM)-S->ARC/INFO-S->SVF
 LUNR-S->VDI(RIM)-S->LAS
 LUNR-S->VDI(RIM)-S->MOSS
 LUNR-S->VDI(RIM)-S->MOSS-S->COS
 LUNR-S->VDI(RIM)-S->MOSS-S->MAPS
 LUNR-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP
 LUNR-S->VDI(RIM)-T->INTERGRAPH
 LUNR-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 LUNR-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 LUNR-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 LUNR-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE

MAP-D->ADS
 MAP-D->AGIS/GRAM
 MAP-D->AHDS
 MAP-D->AMS
 MAP-D->ARC/INFO
 MAP-D->CONTRACTOR FMT
 MAP-D->CRAIGES
 MAP-D->GES
 MAP-D->GIRAS
 MAP-D->INTERGRAPH

MAPS-S->MOSS
 MAPS-S->MOSS-M->MAP
 MAPS-S->MOSS-S->COS
 MAPS-S->MOSS-S->MOSS IMP/EXP

MOSS IMP/EXP-S->MOSS

MOSS IMP/EXP-S->MOSS-M->MAP
MOSS IMP/EXP-S->MOSS-S->COS
MOSS IMP/EXP-S->MOSS-S->MAPS

MOSS-M->MAP
MOSS-S->COS
MOSS-S->MAPS
MOS-S->MOSS IMP/EXP

PIOS-S->ARC/INFO
PIOS-S->ARC/INFO-M->MAP
PIOS-S->ARC/INFO-S->ARC IMP/EXP
PIOS-S->ARC/INFO-S->DLG OPTIONAL
PIOS-S->ARC/INFO-S->DLG OPTIONAL-S->GES
PIOS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
PIOS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->COS
PIOS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MAPS
PIOS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MOSS IMP/EXP
PIOS-S->ARC/INFO-S->GIMMS
PIOS-S->ARC/INFO-S->SVF
PIOS-S->ARC/INFO-S->VDI(RIM)
PIOS-S->ARC/INFO-S->VDI(RIM)-P->FGEF
PIOS-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
PIOS-S->ARC/INFO-S->VDI(RIM)-S->LAS
PIOS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
PIOS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
PIOS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
PIOS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
PIOS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE

SIF-S->INTERGRAPH
SIF-S->INTERGRAPH-S->DLG
SIF-S->INTERGRAPH-S->DLG-O->DCASS
SIF-S->INTERGRAPH-S->DLG-O->TIGER
SIF-S->INTERGRAPH-S->DLG-S->ARC/INFO
SIF-S->INTERGRAPH-S->DLG-S->ARC/INFO-S->ARC IMP/EXP
SIF-S->INTERGRAPH-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
SIF-S->INTERGRAPH-S->DLG-S->ARC/INFO-S->GIMMS
SIF-S->INTERGRAPH-S->DLG-S->ARC/INFO-S->PIOS
SIF-S->INTERGRAPH-S->DLG-S->ARC/INFO-S->SVF
SIF-S->INTERGRAPH-S->DLG-S->GES
SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)
SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-P->FGEF
SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-S->AGIS/GRAM
SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-S->LAS
SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS
SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS-S->COS
SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS-S->MAPS

SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP
SIF-S->INTERGRAPH-S->EXTELE
SIF-S->INTERGRAPH-S->EXTELE-M->MAP

SVF-S->ARC/INFO
SVF-S->ARC/INFO-M->MAP
SVF-S->ARC/INFO-S->ARC IMP/EXP
SVF-S->ARC/INFO-S->DLG OPTIONAL
SVF-S->ARC/INFO-S->DLG OPTIONAL-S->GES
SVF-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
SVF-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->COS
SVF-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MAPS
SVF-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS-S->MOSS IMP/EXP
SVF-S->ARC/INFO-S->GIMMS
SVF-S->ARC/INFO-S->PIOS
SVF-S->ARC/INFO-S->VDI(RIM)
SVF-S->ARC/INFO-S->VDI(RIM)-P->FGEF
SVF-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
SVF-S->ARC/INFO-S->VDI(RIM)-S->LAS
SVF-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
SVF-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
SVF-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
SVF-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
SVF-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->EXTELE

TIGER-O->DLG
TIGER-O->DLG-O->DCASS
TIGER-O->DLG-S->ARC/INFO
TIGER-O->DLG-S->ARC/INFO-S->ARC IMP/EP
TIGER-O->DLG-S->ARC/INFO-S->DLG OPTIONAL
TIGER-O->DLG-S->ARC/INFO-S->GIMMS
TIGER-O->DLG-S->ARC/INFO-S->PIOS
TIGER-O->DLG-S->ARC/INFO-S->SVF
TIGER-O->DLG-S->GES
TIGER-O->DLG-S->GES-M->MAP
TIGER-O->DLG-S->INTERGRAPH
TIGER-O->DLG-S->INTERGRAPH-S->EXTELE
TIGER-O->DLG-S->VDI(RIM)
TIGER-O->DLG-S->VDI(RIM)-P->FGEF
TIGER-O->DLG-S->VDI(RIM)-S->AGIS/GRAM
TIGER-O->DLG-S->VDI(RIM)-S->LAS
TIGER-O->DLG-S->VDI(RIM)-S->MOSS
TIGER-O->DLG-S->VDI(RIM)-S->MOSS-S->COS
TIGER-O->DLG-S->VDI(RIM)-S->MOSS-S->MAPS
TIGER-O->DLG-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP

UCLGES UPDATE-S->UCLGES
UCLGES UPDTE-S->UCLGES-S->DLG

UCLGES UPDATE-S->UCLGES-S->DLG-O->DCASS
 UCLGES UPDATE-S->UCLGES-S->DLG-O->TIGER
 UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO
 UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->ARC IMP/EXP
 UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
 UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->GIMMS
 UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->PIOS
 UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->SVF
 UCLGES UPDATE-S->UCLGES-S->DLG-S->GES
 UCLGES UPDATE-S->UCLGES-S->DLG-S->GES-M->MAP
 UCLGES UPDATE-S->UCLGES-S->DLG-S->INTERGRAPH
 UCLGES UPDATE-S->UCLGES-S->DLG-S->INTERGRAPH-S->EXTELE
 UCLGES UPDATE-S->UCLGES-S->DL-S->VDI(RIM)
 UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-P->FGEF
 UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->AGIS/GRAM
 UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->LAS
 UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS
 UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS-S->COS
 UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS-S->MAPS
 UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP

UCLGES-S->DLG
 UCLGES-S->DLG-O->DCASS
 UCLGES-S->DLG-O->TIGER
 UCLGES-S->DLG-S->ARC/INFO
 UCLGES-S->DLG-S->ARC/INFO-S->ARC IMP/EXP
 UCLGES-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
 UCLGES-S->DLG-S->ARC/INFO-S->GIMMS
 UCLGES-S->DLG-S->ARC/INFO-S->PIOS
 UCLGES-S->DL-S->ARC/INFO-S->SVF
 UCLGES-S->DLG-S->GES
 UCLGES-S->DLG-S->GES-M->MAP
 UCLGES-S->DLG-S->INTERGRAPH
 UCLGES-S->DLG-S->INTERGRAPH-S->EXTELE
 UCLGES-S->DLG-S->VDI(RIM)
 UCLGES-S->DLG-S->VDI(RIM)-P->FGEF
 UCLGES-S->DLG-S->VDI(RIM)-S->AGIS/GRAM
 UCLGES-S->DLG-S->VDI(RIM)-S->LAS
 UCLGES-S->DLG-S->VDI(RIM)-S->MOSS
 UCLGES-S->DLG-S->VDI(RIM)-S->MOSS-S->COS
 UCLGES-S->DLG-S->VDI(RIM)-S->MOSS-S->MAPS
 UCLGES-S->DLG-S->VDI(RIM)-S->MOSS-S->MOSS IMP/EXP

VDI(RIM)-M->MAP
 VDI(RIM)-P->FGEF
 VDI(RIM)-S->AGIS/GRAM
 VDI(RIM)-S->ARC/INFO
 VDI(RIM)-S->ARC/INFO-S->ARC IMP/EXP

VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 VDI(RIM)-S->ARC/INFO-S->GIMMS
 VDI(RIM)-S->ARC/INFO-S->PIOS
 VDI(RIM)-S->ARC/INFO-S->SVF
 VDI(RIM)-S->LAS
 VDI(RIM)-S->MOSS
 VDI(RIM)-S->MOSS-S->COS
 VDI(RIM)-S->MOSS-S->MAPS
 VDI(RIM)-S->MOSS-S->MOSS IMP/EXP
 VDI(RIM)-T->INTERGRAPH
 VDI(RIM)-T->INTERGRAPH-S->DLG
 VDI(RIM)-T->INTERGRAPH-S->DLG-O->DCASS
 VDI(RIM)-T->INTERGRAPH-S->DLG-O->TIGER
 VDI(RIM)-T->INTERGRAPH-S->EXTELE

F.2 Data Flows by Output Data Structure

(first priority only)

AGIS/GRAM

PATH

ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-S->AGIS/GRAM
 ALMRS-P->VDI(RIM)-S->AGIS/GRAM
 AMS-T->VDI(RIM)-S->AGIS/GRAM
 ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
 ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)-S->AGIS/GRAM
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->AGIS/GRAM
 DIME-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
 DLG OPTIONAL-S->GES-S->VDI(RIM)-S->AGIS/GRAM
 DLG-S->VDI(RIM)-S->AGIS/GRAM
 FGEF-P->VDI(RIM)-S->AGIS/GRAM
 GES-S->VDI(RIM)-S->AGIS/GRAM
 GIMMS-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
 GIRAS-O->DLG-S->VDI(RIM)-S->AGIS/GRAM
 INTERGRAPH-S->DLG-S->VDI(RIM)-S->AGIS/GRAM
 LAS-S->VDI(RIM)-S->AGIS/GRAM
 LUNR-S->VDI(RIM)-S->AGIS/GRAM
 MAP-D->AGIS/GRAM
 PIOS-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
 SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-S->AGIS/GRAM
 SVF-S->ARC/INFO-S->VDI(RIM)-S->AGIS/GRAM
 TIGER-O->DLG-S->VDI(RIM)-S->AGIS/GRAM
 UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->AGIS/GRAM

UCLGES-S->DLG-S->VDI(RIM)-S->AGIS/GRAM
VDI(RIM)-S->AGIS/GRAM

AMS

PATH

MAP-D->AMS

ARC/INFO

PATH

ADS-O->DLG OPTIONAL-S->ARC/INFO
ALMRS-P->VDI(RIM)-S->ARC/INFO
AMS-T->VDI(RIM)-S->ARC/INFO
ARC IMP/EXP-S->ARC/INFO
CONTRACTOR FMT-O->GIRAS-O->DLG-S->ARC/INFO
CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO
DIME-S->ARC/INFO
DLG OPTIONAL-S->ARC/INFO
DLG-S->ARC/INFO
FGEF-P->VDI(RIM)-S->ARC/INFO
GES-S->VDI(RIM)-S->ARC/INFO
GIMMS-S->ARC/INFO
GIRAS-O->DLG-S->ARC/INFO
INTERGRAPH-S->DLG-S->ARC/INFO
LAS-S->VDI(RIM)-S->ARC/INFO
LUNR-S->VDI(RIM)-S->ARC/INFO
AP-D->ARC/INFO
PIOS-S->ARC/INFO
SIF-S->INTERGRAPH-S->DLG-S->ARC/INFO
SVF-S->ARC/INFO
TIGER-O->DLG-S->ARC/INFO
UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO
UCLGES-S->DLG-S->ARC/INFO
VDI(RIM)-S->ARC/INFO

DLG

PATH

ALMRS-P->VDI(RIM)-T->INTERGRAPH-S->DLG
AMS-O->DLG
ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
CONTRACTOR FMT-O->GIRAS-O->DLG
CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG
DIME-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG

FGEF-P->VDI(RIM)-T->INTERGRAPH-S->DLG
 GES-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 GIMMS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 GIRAS-O->DLG
 INTERGRAPH-S->DLG
 LAS-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 LUNR-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 PIOS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 SIF-S->INTERGRAPH-S->DLG
 SVF-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH-S->DLG
 TIGER-O->DLG
 UCLGES UPDATE-S->UCLGES-S->DLG
 UCLGES-S->DLG
 VDI(RIM)-T->INTERGRAPH-S->DLG

DLG OPTIONAL

PATH

ADS-O->DLG OPTIONAL
 ALMRS-P->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 AMS-T->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 ARC IMP/EXP-S->ARC/INFO-S->DLG OPTIONAL
 ARC/INFO-S->DLG OPTIONAL
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->ARC/INFO-S->DLG OPTIONAL
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
 DIME-S->ARC/INFO-S->DLG OPTIONAL
 DLG-S->ARC/INFO-S->DLG OPTIONAL
 FGEF-P->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 GES-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 GIMMS-S->ARC/INFO-S->DLG OPTIONAL
 GIRAS-O->DLG-S->ARC/INFO-S->DLG OPTIONAL
 INTRGRAPH-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
 LAS-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 LUNR-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL
 PIOS-S->ARC/INFO-S->DLG OPTIONAL
 SIF-S->INTERGRAPH-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
 SVF-S->ARC/INFO-S->DLG OPTIONAL
 TIGER-O->DLG-S->ARC/INFO-S->DLG OPTIONAL
 UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
 UCLGES-S->DLG-S->ARC/INFO-S->DLG OPTIONAL
 VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL

GES

PATH

ADS-O->DLG OPTIONAL-S->GES

ALMRS-P->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 AMS-T->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 ARC IMP/EXP-S->ARC/INFO-S->DG OPTIONAL-S->GES
 ARC/INFO-S->DLG OPTIONAL-S->GES
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->GES
 DIME-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 DLG OPTIONAL-S->GES
 DLG-S->GES
 FGEF-P->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 GIMMS-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 GIRAS-O->DLG-S->GES
 INTERGRAPH-S->DLG-S->GES
 LAS-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 LUNR-S->VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 MAP-D->CRAIGES
 MAP-D->GES
 PIOS-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 SIF-S->INTERGRAPH-S->DLG-S->GES
 SVF-S->ARC/INFO-S->DLG OPTIONAL-S->GES
 TIGER-O->DLG-S->GES
 UCLGES UPDATE-S->UCLGES
 VDI(RIM)-S->ARC/INFO-S->DLG OPTIONAL-S->GES

GIMMS

PATH

ADS-O->DG OPTIONAL-S->ARC/INFO-S->GIMMS
 ALMRS-P->VDI(RIM)-S->ARC/INFO-S->GIMMS
 AMS-T->VDI(RIM)-S->ARC/INFO-S->GIMMS
 ARC IMP/EXP-S->ARC/INFO-S->GIMMS
 ARC/INFO-S->GIMMS
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->ARC/INFO-S->GIMMS
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->GIMMS
 DIME-S->ARC/INFO-S->GIMMS
 DLG OPTIONAL-S->ARC/INFO-S->GIMMS
 DLG-S->ARC/INFO-S->GIMMS
 FGEF-P->VDI(RIM)-S->ARC/INFO-S->GIMMS
 GES-S->VDI(RIM)-S->ARC/INFO-S->GIMMS
 GIRAS-O->DLG-S->ARC/INFO-S->GIMMS
 INTERGRAPH-S->DLG-S->ARC/INFO-S->GIMMS
 LAS-S->VDI(RIM)-S->ARC/INFO-S->GIMMS
 LNR-S->VDI(RIM)-S->ARC/INFO-S->GIMMS
 PIOS-S->ARC/INFO-S->GIMMS
 SIF-S->INTERGRAPH-S->DLG-S->ARC/INFO-S->GIMMS
 SVF-S->ARC/INFO-S->GIMMS
 TIGER-O->DLG-S->ARC/INFO-S->GIMMS
 UCLGES UPDATE-S->UCLGES-S->DLG-S->ARC/INFO-S->GIMMS

UCLGES-S->DLG-S->ARC/INFO-S->GIMMS
VDI(RIM)-S->ARC/INFO-S->GIMMS

INTERGRAPH

PATH

ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH
ALMRS-P->VDI(RIM)-T->INTERGRAPH
AMS-T->VDI(RIM)-T->INTERGRAPH
ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
CONTRACTOR FMT-O->GIRAS-O->DLG-S->INTERGRAPH
CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->INTERGRAPH
DIME-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
DLG OPTIONAL-S->GES-S->VDI(RIM)-T->INTERGRAPH
DLG-S->INTERGRAPH
FGEF-P->VDI(RIM)-T->INTERGRAPH
GES-S->VDI(RIM)-T->INTERGRAPH
GIMMS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
GIRAS-O->DLG-S->INTERGRAPH
LAS-S->VDI(RIM)-T->INTERGRAPH
LUNR-S->VDI(RIM)-T->INTERGRAPH
MAP-D->INTERGRAPH
PIOS-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
SIF-S->INTERGRAPH
SVF-S->ARC/INFO-S->VDI(RIM)-T->INTERGRAPH
TIGER-O->DLG-S->INTERGRAPH
UCLGES UPDATE-S->UCLGES-S->DLG-S->INTERGRAPH
UCLGES-S->DLG-S->INTERGRAPH
VDI(RIM)-T->INTERGRAPH

LAS

PATH

ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)-S->LAS
ALMRS-P->VDI(RIM)-S->LAS
AMS-T->VDI(RIM)-S->LAS
ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)-S->LAS
ARC/INFO-S->VDI(RIM)-S->LAS
CONTRACTOR FT-O->GIRAS-O->DLG-S->VDI(RIM)-S->LAS
CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->LAS
DIME-S->ARC/INFO-S->VDI(RIM)-S->LAS
DLG OPTIONAL-S->GES-S->VDI(RIM)-S->LAS
DLG-S->VDI(RIM)-S->LAS
FGEF-P->VDI(RIM)-S->LAS
GES-S->VDI(RIM)-S->LAS

GIMMS-S->ARC/INFO-S->VDI(RIM)-S->LAS
 GIRAS-O->DLG-S->VDI(RIM)-S->LAS
 INTERGRAPH-S->DLG-S->VDI(RIM)-S->LAS
 LUNR-S->VDI(RIM)-S->LAS
 PIOS-S->ARC/INFO-S->VDI(RIM)-S->LAS
 SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-S->LAS
 SVF-S->ARC/INFO-S->VDI(RIM)-S->LAS
 TIGER-O->DLG-S->VDI(RIM)-S->LAS
 UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->LAS
 UCLGES-S->DLG-S->VDI(RIM)-S->LAS
 VDI(RIM)-S->LAS

MAP

PATH

ADS-M->MAP
 AGIS/GRAM-S->MAP
 AHDS-U->MOSS-M->MAP
 ALMRS-P->VDI(RIM)-M->MAP
 AMS-M->MAP
 ARC IMP/EXP-S->ARC/INFO-M->MAP
 ARC/INFO-M->MAP
 CONTRACTOR FMT-O->GIRAS-O->DLG-S->GES-M->MAP
 CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->GES-M->MAP
 DCASS-M->MAP
 DIME-S->ARC/INFO-M->MAP
 DLG OPTIONAL-S->GES-M->MAP
 DLG-S->GES-M->MAP
 EXTELE-M->MAP
 FGEF-P->VDI(RIM)-M->MAP
 GES-M->MAP
 GIMMS-S->ARC/INFO-M->MAP
 GIRAS-O->DLG-S->GES-M->MAP
 INTERGRAPH-S->EXTELE-M->MAP
 LAS-S->VDI(RIM)-M->MAP
 LUNR-S->VDI(RIM)-M->MAP
 MOSS IMP/EXP-S->MOSS-M->MAP
 MOSS-M->MAP
 PIOS-S->ARC/INFO-M->MAP
 SIF-S->INTERGRAPH-S->EXTELE-->MAP
 SVF-S->ARC/INFO-M->MAP
 TIGER-O->DLG-S->GES-M->MAP
 UCLGES UPDATE-S->UCLGES-S->DLG-S->GES-M->MAP
 UCLGES-S->DLG-S->GES-M->MAP
 VDI(RIM)-M->MAP

MOSS

PATH

ADS-O->MOSS IMP/EXP
ADS-O->MOSS IMP/EXP-S->MOSS
AHDS-U->MOSS
ALMRS-P->VDI(RIM)-S->MOSS
AMS-S->MOSS
ARC IMP/EXP-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
ARC/INFO-S->DLG OPTIONAL-S->MOSS
CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)-S->MOSS
CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS
DIME-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
DLG OPTIONAL-S->MOSS
DLG-S->VDI(RIM)-S->MOSS
FGEF-P->VDI(RIM)-S->MOSS
GES-S->VDI(RIM)-S->MOSS
GIMMS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
GIRAS-O->DLG-S->VDI(RIM)-S->MOSS
INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS
LAS-S->VDI(RIM)-S->MOSS
LUNR-S->VDI(RIM)-S->MOSS
MAPS-S->MOSS
MOSS IMP/EXP-S->MOSS
PIOS-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)-S->MOSS
SVF-S->ARC/INFO-S->DLG OPTIONAL-S->MOSS
TIGER-O->DLG-S->VDI(RIM)-S->MOSS
UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)-S->MOSS
UCLGES-S->DLG-S->VDI(RIM)-S->MOSS
VDI(RIM)-S->MOSS

VDI(RIM)

PATH

ADS-O->DLG OPTIONAL-S->GES-S->VDI(RIM)
ALMRS-P->VDI(RIM)
AMS-T->VDI(RIM)
ARC IMP/EXP-S->ARC/INFO-S->VDI(RIM)
ARC/INFO-S->VDI(RIM)
CONTRACTOR FMT-O->GIRAS-O->DLG-S->VDI(RIM)
CRAIGES-S->UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)
DIME-S->ARC/INFO-S->VDI(RIM)
DLG OPTIONAL-S->GES-S->VDI(RIM)
DLG-S->VDI(RIM)
FGEF-P->VDI(RIM)
GES-S->VDI(RIM)

GIMMS-S->ARC/INFO-S->VDI(RIM)
GIRASO->DLG-S->VDI(RIM)
INTERGRAPH-S->DLG-S->VDI(RIM)
LAS-S->VDI(RIM)
LUNR-S->VDI(RIM)
PIOS-S->ARC/INFO-S->VDI(RIM)
SIF-S->INTERGRAPH-S->DLG-S->VDI(RIM)
SVF-S->ARC/INFO-S->VDI(RIM)
TIGER-O->DLG-S->VDI(RIM)
UCLGES UPDATE-S->UCLGES-S->DLG-S->VDI(RIM)
UCLGES-S->DLG-S->VDI(RIM)