GIS Core Database Revision:

Revised Core Database General Architecture Design

March 9th, 2000

DRAFT

Robert Maki
GIS Database Coordinator
Minnesota DNR
Management Information
Services Bureau

Revised Core Database General Architecture Design

1.0 Introduction

This system design document describes a comprehensive database architecture for the Minnesota Department of Natural Resources (DNR). The Revised Core Architecture includes the administrative components of data maintenance, access, cataloging, and description. Technically, business application development and database design efforts exist apart from the architecture. The database architecture exists to standardize information storage and access, streamline end-user access to information, reduce the costs associated with new database and application development, and set the stage for information systems integration at the database level.

Although the effort is being driven principally by changes in DNR Geographic Information System (GIS) requirements, it is expected that some components will support the full domain of corporate information technology resources within the organization. In fact, the integration of GIS and traditional databases, resulting in new information resources, is a principle objective of the effort.

This document is preceded by two other documents that describe various aspects of the system and its implementation. These include a general white paper on the subject, "GIS Core Database Architecture Revision", and a project plan, "Revised Core Database Architecture Project Plan." These are included in the attached bibliography.

This document is oriented around the primary data storage structures and supporting processes that comprise the proposed system. Section 3 of the document provides an overview of these. Most of the descriptions rely on Data Flow Diagrams (DFD's) to describe the various subsystems, and are supported by textual descriptions that describe preferred approaches, detailed design considerations, and standards requirements. The remainder of the document describes key data stores and processes in separate sections, as follows:

Section Number	*Section Title*	*Section Description*
Section 2:	Design Principles	Assumptions underlying the effort
Section 3:	Principle System Components	An overview discussion of the various components which comprise the Revised Core Database environment
Section 4:	Core Database Components	A discussion on the range of data resources and storage constraints of the integrated Core Database environment
Section 5:	Core Access Site Definition	Characteristics and description of an Application Data Store, including both central and remote incarnations

Section Number	*Section Title*	*Section Description*
Section 6:	Data Dictionary	A discussion of the general parameters of an extended data dictionary to support the system, and serve as the basis for database integration in new efforts, and legacy conversions
Section 7:	Mirroring Subsystem	A description of the subsystem that copies data out to remote Core Access Sites
Section 8:	Product Generation Subsystem	A description of the processes that are used to generate Application Data from Core Data
Section 9:	Access Support Subsystem	A description of the processes that facilitate access to Core Access Site data and link remote sites to the Data Dictionary
Section 10:	Wrap-up and Next Steps	Additional considerations plus last thoughts and comments on the project

2.0 Design Principles

The Revised Core Architecture builds on our experiences with the original GIS Core Database by extending the existing design concepts to new types of source data, new technologies, and greater flexibility in end user applications. The approach outlined here is intended to improve access to the technology.

Technology is in a constant state of change–a condition which threatens to continually destabilize information systems. The Revised Core Architecture should serve to insulate users from structural changes in technology (data migration, new database environments, hardware/software revisions) to the greatest extent possible.

All efforts of this type rest on a set of underlying assumptions. The following list was developed from our current experiences with database architecture development and operation within DNR:

1. The DNR is a distributed application environment where staff in remote locations require localized information products to overcome technological limits in network bandwidth for acceptable performance. Although this is probably not the case for traditional, non-graphic intensive applications, it is certainly true for GIS applications. This condition will persist over at least a five year time horizon, even as sub-regional network connectivity becomes more common. Optimally, remote sites will have access to local, as well as centralized data resources.
2. DNR staff desire information products that are tailored to their requirements and which can function effectively in responsive desktop applications. These products should be rich in long-name descriptions, rather than obtuse attribute codes.
3. DNR corporate data resources should and will experience a continuing process of consolidation and integration. Users should be insulated from major structural changes in Core data.
4. Managed data resources within the organization exist in a diversity of administrative environments (hardware, software) and will continue to do so over the next five years, although on-going standardization in this area is desirable and will be an investment priority through this period.
5. End-user applications will be centered on the personal computer, either as desktop application clients or as intranet browser clients. Users desire a data access environment where they need not know the particulars of where and how data are stored on the system.
6. The department has a significant number of general access users (typically, ARCVIEW-based) that need ad hoc access to a wide variety of data and productivity tools.
7. Data resources should be documented to the fullest practical extent. DNR users rely on documentation to determine fitness of use for their applications.
8. All efforts in database infrastructure development should be build around our current hardware and software computing resources, avoiding major capital investments when possible. We should seek to leverage our existing investments.

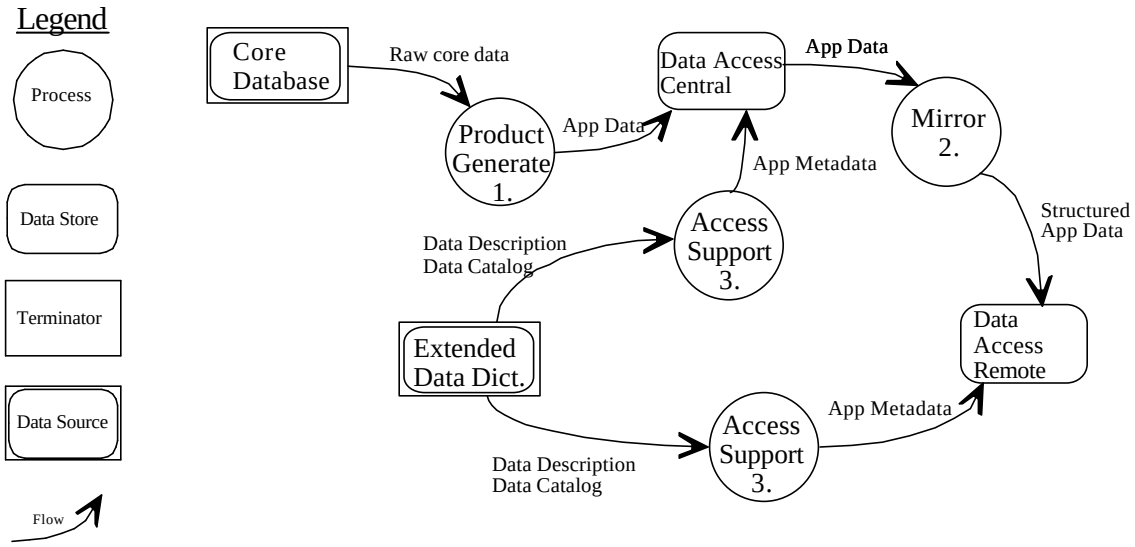
3.0 Principle System Components

The Revised Core Database Infrastructure has seven primary database and subsystem components. These are listed in Table 1, and presented in a data flow diagram (DFD) in Figure 1. Each component is discussed in separate sections below.

Table 1: Primary Architecture Component Descriptions

Component	*Description*
Core Database	The suite of managed data stores, optimized for administrative processes, including maintenance and integration. These will include the range of formats in which strategic data resources are stored.
Data Access Site-Central	The principal location where application data products are stored. This will include the range of formats in which data resources are accessed. It serves as both a point of access for central office applications, and the staging area for data distribution to remote data access sites.
Data Access Site-Remote	The suite of locations that provide access to application data products at remote sites, including regional offices and (potentially) sub-regional sites. These will include the range of formats in which strategic data resources are accessed. Data are mirrored from the central to the remote application data sites.
Extended Data Dictionary	A centralized database serving as the repository for data element definition, narrative metadata, and core and application product registeries.
Mirror	A set of automated processes which keep Remote Application Data Sites in sync with the Central version.
Product Generate	A set of automated processes which generate application data products that reside on the central Application Data Site.
Access Support	A set of automated processes which generate system metafiles that support standardized access and application interface to the various Data Access Sites.

Figure 1: Principal System Components Data Flow Diagram



Note: Access Support Processes (3.) are identical but separate instances

4.0 Core Database

The Core Database is the collective environment where digital enterprise data resources are stored and maintained. The Core Database is optimized for database administration processes and aspires to high levels of content integration. The Core Database is comprised of multiple sources, including: traditional enterprise databases in AS/400 and Oracle environments, GIS databases in ARC/INFO data formats, and (potentially) Novell-based MS-Access databases. Source types are listed in Table 2. Some data standards will be globally applicable, while others will be specific to a particular environment. The enterprise has a long-term objective to consolidate data resources into a common Oracle-based environment, although this will take years to accomplish.

Table 2: Core Database Environment Types

Data Type	*Description*
ARC/INFO filesystem-based	Includes Shapefiles and/or ARC/INFO 7.X coverage formats at some level
Oracle-ARC/INFO DBMS	ARC/INFO 8.X Geodatabase
Oracle DBMS	Traditional relational database administrative environment
AS/400 UDB2	Traditional relational database administrative environment
MS-Access	MS-Access support is uncertain. It would be desirable to capitalize on some of the data resident in this environment, but a number technical issues present themselves, particularly interfacing with these databases through automated processes.

Core Database Elements are subsidiary databases. It would be desirable for this body of data to conform to a particular suite of database, table, and field naming standards, but in reality, this is not realistic. Large volumes of legacy data will be entering the system, bringing with it a variety of conventions that must be supported for the foreseeable future.

Some Core Database element types will conform to file system level structural standards. Clearly ARC/INFO coverages and ARC Shapefiles will require structured organization at this level. These structural standards may be borrowed directly from the current Core Database architecture.

The Core Database will be administered by multiple persons, including representatives from potentially every business unit within the DNR. Some maintenance processes will be highly structured and distributed (e.g. customer database), while others will be centralized and much less structured. Some maintenance processes will conducted via web-based technologies. Structural database standards could assist in the development of standardized maintenance applications.

The Core Database should include some provision for data versioning. Categorizing changes as to type is useful when considering the requirements for version tracking. Error correction as a category of change probably does not need to be tracked, while temporal changes probably should be logged in some fashion. The latter case is one of the principle features of a so-called data warehouse, where legacy information is maintained and made accessible. If the source data are stored in a Oracle environment, then some provision for versioning at that level could presumably be made, which could find its way into the derived products. Data managed on fileservers will have to have some other mechanism for tracking change, and perhaps some business-specific applications for presenting data changes within desktop applications.

Although most data need not include a lineage in its maintenance environment, others will require it. Business data with legal implications, such as land records and the Protected Waters Inventory (PWI) may require roll-back capabilities to illustrate the state of the data at some particular point in time. Certain types of land imagery are available to users (e.g. LandSat TM data) with multiple dates for any given area. The temporal aspects of the data are extremely valuable for land planners, ecologists, and policy makers.

Core Database Elements will be linked using both client-server TCP/IP and mounted file system protocols.

5.0 Access Site-Central and Remote

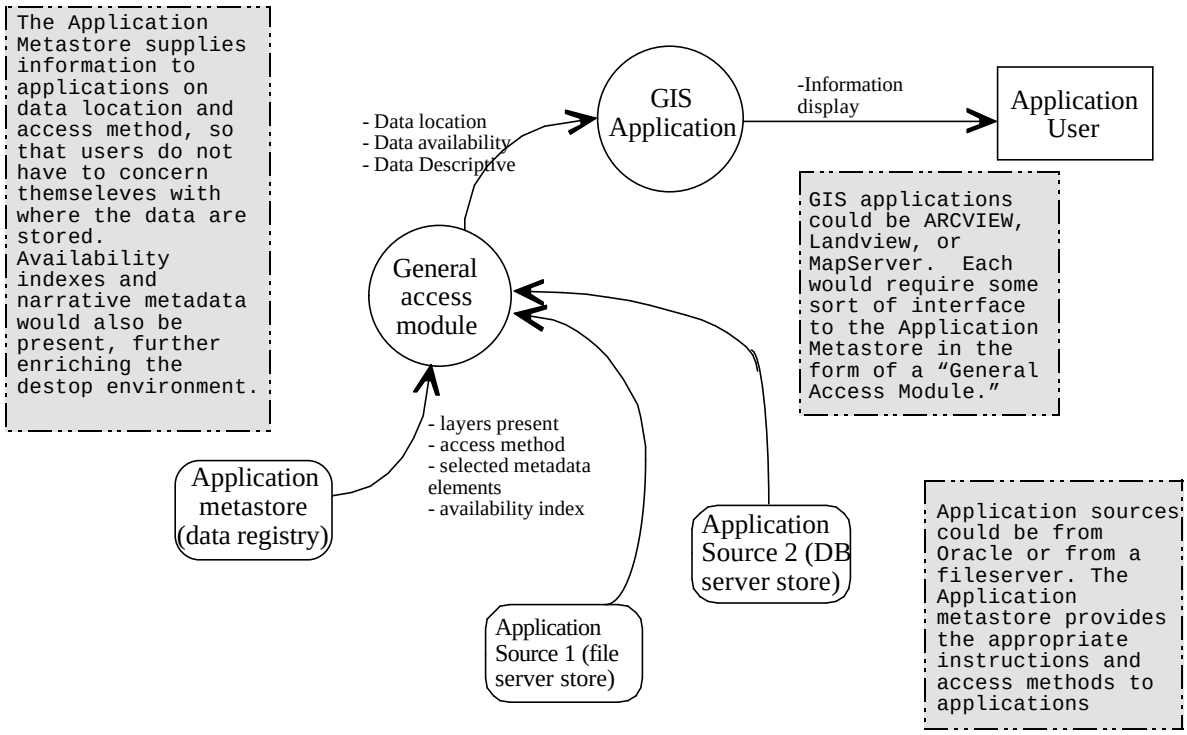
The central office Access Site is where derived ("application") products are placed upon creation. Access Sites are centered around a suite of system level metadata which provide a registry of data available on a site. Some data will exist on a local file server within a standardized directory structure using standardized file names. Some data will reside on database servers, and will be accessible through client-server applications. Data may exist as actual derived data sets, or as "logical or "virtual layers" that are the result of a query process against an application server. Data may be truly "local", or reside remotely while appearing local. The various storage schemes are listed in Table 3.

Table 3: Access Site Data Types

Data Type	*Local Storage Option*	*remote Storage Option*	*Description*
File server	yes	no	Fileserver-based access to ARC/INFO data
Database server-derived	yes	yes	Derived data resident within a DBMS environment
Database server-logical	yes	yes	Logical data view within a DBMS environment or in the case of SDE for Coverages, traditional ARC/INFO data presented by a database technology

All data types will exist within the local registry (which is provided via a file server), along with their specific access method. The registry would likely be fileserver-based, for reasons that are discussed in Section 9. In some future world when network capacity increases, the entire Access Site concept may collapse into a single enterprise-wide access site that is referenced from all locations. The concept of an Access Site with an associated data registry and an application interface is show in Figure 2.

Figure 2: Data Access Site-Application Interface



Remote access sites are identical to the Central Access Site, except that they are more distant from the Data Dictionary subsystem which serves as the master registry of data sources and participating sites. The data content of the remote sites will vary somewhat, since resource management, land ownership, and land use patterns, and associated data requirements vary considerably between regions.

6.0 Extended Data Dictionary

The Data Dictionary subsystem consists of a wide array of metadata resources, ranging from classic entity-level data definition, to detailed attribute-level descriptions of data products, to product catalogs and narrative metadata descriptions (e.g. supporting standard GIS data reporting). The subsystem also would support the generation of narrative user products (e.g. web-based DNR GIS data resource pages) and data discovery tools (search engines) as the array of available data resources increases in complexity and volume. A listing of required data elements is provided in Table 4. This list is not exhaustive. Other elements may necessarily be added to support other DNR business functions.

Table 4: Required Data Dictionary Elements

Element	*Comments*
Data Element Definition	Physical Entity Definition (Possibly the same as Table Registry)
Attribute Field Registry	A master registry of attribute fields that can be cross-referenced against Entity Definition and Table Registry entries
Attribute Field Domains	Domains of each attribute field. Conceptually (and perhaps physically) embedded in Field Registry.
Table Registry	A master registry of both physical and virtual attribute tables, cross-referenced to Attribute field domains. Spatial feature classes are registered as tables.
Core "Layer" Registry	A master list of Core Data Products. There are difficulties associated with defining Core "layers". The term may not be meaningful for some suites of highly integrated data. On the other hand, some spatial core products (.e.g. image products), cannot be integrated and therefore do exist as discrete "layers". These may be registered as application data without a Core Data source. This category of data may not exist within the traditional DB domain.
Application Data Registry	A master list of derived products. This is less ambiguous than the Core Data Registry concept. Application products are by definition discrete. This registry needs to be affiliated with the local registries for each Data Access site.
Core Relationships	We could use some mechanisms for describing relationships amongst some core elements, particularly relational tables. At this point we wander into some of the native capabilities present in the RDBMS environment. Still, some relational data will exist outside of (for example) Oracle, and cannot be effectively described using native Oracle.
Application Data Parent-Child Relationships	We need to know where Application Data products come from (i.e. their lineage from the Core Database).

Element	*Comments*
Narrative Metadata	Descriptive metadata is an important part of the GIS documentation environment. This would include the FGDC and MGMG data elements
Product Generation Process Registry	Registry of the process name, type, and specification used to generate derived products

The Data Dictionary subsystem has two crucial points of integration within the Revised Core environment. The first of these is with the Application Support subsystem, where the Data Dictionary describes Application Data Product availability for each site. Certain descriptive metadata elements need to be accessible at the Application Data Site level, perhaps through a daily download-update process from the Central Office to the remote sites.

The second point of integration is with the Product Generation processes. Products and associated methods are identified in the Data Dictionary, and may help drive the set of processes which generate the Application Data Products. It may not be necessary to integrate these two subsystems. Product Generate could function independently.

At the time of this writing, the Data Dictionary is envisioned as an Oracle 8 database, perhaps conforming to the ISO Data Dictionary standard being studied in the MIS Bureau. The Data Dictionary would be maintained through a client application with a graphic interface. On-line metadata products will almost certainly exist as XML documents with associated style sheets.

7.0 Mirror

The Mirror subsystem pushes data from the Central Office Data Access Site to the Remote Data Access Sites. Only fileserver-based products will be affected by this process (Oracle database products are assumed to available via remote access). Mirror will operate wholly from filesystem-based change detection processes, much like the techniques that are used in the current Core Database environment. The difference here will be the physical standards that shape the Revised Core Access Sites. The processes will almost certainly be written in PERL and scheduled for nightly execution. They may either be resident as a Central Office process which pushes data out to the regions, or as a regional process which pulls data from the Central Office. Security considerations may guide the decision as to which configuration is implemented.

8.0 Product Generate

The Product Generate subsystem creates Application Data Products resident on the Central Office Data Access Site from the Core Database. This subsystem will be one of the most difficult to fully realize. Three different data dimensions are at work in the subsystem: 1) source data type(s), 2) target data types, and 3) update schedule.

Product generation processes will likely be written in Structured Query Language (SQL) for Oracle database queries, and ARC Macro Language (AML) for processing ARC/INFO-based products. PERL should be used when possible to optimize text-based processing, especially those processes which feature some form of data import/export through the filesystem.

Some products will have simple data sources, simple target data types, and simple scheduling scenarios (e.g. “Weekly”, or “Static”). This type of product generation process will be simple to support. An example might be “PLS Sections” derived from the “Control-Point Generated PLS” Core layer executed on a weekly schedule using AML. Other product generation processes could be considerably more complex. Consider, for example, “DNR Conservation Officer Contact by Enforcement Area” derived from a combination of GIS base data in ARC/INFO format, and an Oracle-based employee database (a change in either of which might trigger the generation of a new product) using a mixture of AML and SQL through Database Integrator processes. This would require change detection in multiple environments. It may be prudent to abandon change detection triggers from the Product Generate subsystem and move wholly to scheduled updates, except when dealing with both sources and derived products that exist wholly within the Oracle environment, which is better equipped to handle change detection-based processes.

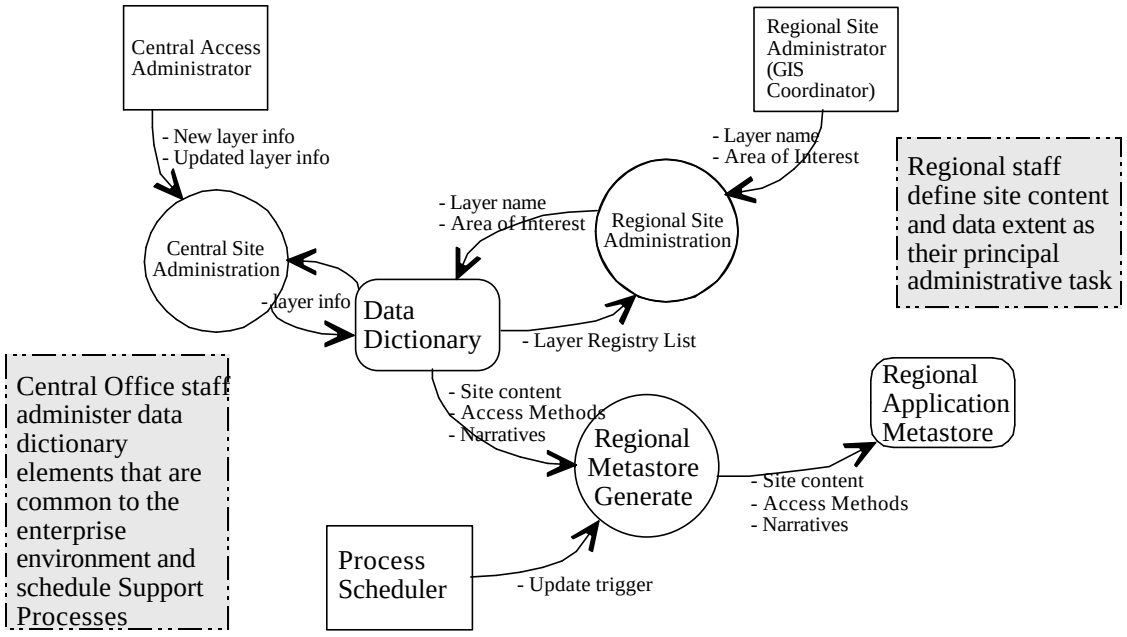
9.0 Access Support

The Access Support subsystem exists to populate and maintain the Application Metastore (data store) introduced in Section 5 of this document. There are a variety of options available for accomplishing this. They include:

1. Keep the Application Metastore (for each access site) in the Data Dictionary. In this scenario, all applications that use the metastore would query the Data Dictionary at startup to establish its access environment. The Access Support subsystem would exist wholly within Oracle.
2. Compose each site-based Application Metastore in the Data Dictionary, and then write out a system file (or series of files) that serve as the Metastore for each site. These might be updated daily, or triggered on demand. Under this scenario, maintenance of all sites would be administered through an Oracle application.
3. Maintain the Application Metastore as an encapsulated environment at the remote site, with system processes that read local data registries, browse data sites, and generate availability information based on what it sees. Under this scenario, maintenance would be centered at the remote site. This would be an extension of the current approach, which includes processes that peruse the filesystem looking for registered data, and record data availability information which are then used by applications (e.g. Core Access Tools, Roadmap).

Of these, Option 2 has some distinct advantages. One of the objectives of the Revised Core Architecture is to reduce the burden of Access Site administration on the Regional GIS Coordinators. Option 2 provides for centralized administration of all sites within Oracle. At the same time, there are advantages to providing local storage (at the remote sites) of the Application Metastore, including 1) stable access that is unaffected by variations in network throughput, 2) allow for application development that can implemented in standalone environments, as well networked ones, and 3) the potential for a three-tier architecture-based site administrative applications, facilitating maintenance of the administrative environment. Option 1 has the disadvantage of not working for non-networked locations, although it probably would work for regional ones (over the wide-area network). Option 3 is unnecessarily distributed, and probably has a higher overhead associated with its maintenance than the other options. Option 2 is depicted graphically in Figure 3.

Figure 3: Application Metastore Generation and Associated System Administration



10.0 Wrap-up and Next Steps

This general design document provides the framework for proceeding with the detailed design, prototyping, development, and integration of the various infrastructure components. Subsequent investigation may serve to invalidate some of the conclusions and directions suggested here, and take individual components into somewhat different implementation scenarios. Still, this document stands as the initial framework for the effort.

It is expected that technical teams will be assembled around the various subsystems with staff contributed from various DNR units in addition to MIS Bureau staff. MIS Bureau staff will have principle responsibility for system implementation, integration, and operation.

Bibliography

Department of Natural Resources Management Information Services Bureau, “GIS Core Database Architecture Revision”, 1999.

Department of Natural Resources Management Information Services Bureau, “Revised Core Database Architecture Project Plan”, 2000.