IJGIS Theme Issue on Interoperability, Volume 12 Number 4, June 1998, pp.403-425

ISSUES AND PROSPECTS FOR THE NEXT GENERATION OF THE SPATIAL DATA TRANSFER STANDARD (SDTS)

AUTHORS

David Arctur, Ph.D. (author for correspondence)
Laser-Scan, Inc.
45635 Willow Pond Plaza,
Sterling, Virginia, 20164-4457 USA
Arctur@acm.org

David Hair
EROS Data Center, U.S. Geological Survey
Sioux Falls, South Dakota, USA
Hair@usgs.gov

George Timson
Mid-Continent Mapping Center, U.S. Geological Survey
Rolla, Missouri, USA
Timson@usgs.gov

E. Paul Martin
Rocky Mountain Mapping Center, U.S. Geological Survey
Denver, Colorado, USA
EPMartin@usgs.gov

Robin Fegeas
National Mapping Division, U.S. Geological Survey
Reston, Virginia, USA
RFegeas@usgs.gov

KEYWORDS

SDTS, OpenGIS, Object-Oriented, Data Sharing/Exchange, Standards

SHORT TITLE

Issues and Prospects for SDTS

ISSN 1365-8816

IJGIS Theme Issue on Interoperability, Volume 12 Number 4, June 1998, pp.403-425

ISSUES AND PROSPECTS FOR THE NEXT GENERATION OF THE SPATIAL DATA TRANSFER STANDARD (SDTS)

D. Arctur
Laser-Scan Inc.

R. Fegeas, D. Hair, G. Timson, E. Martin
US Geological Survey

ABSTRACT

The Spatial Data Transfer Standard (SDTS) was designed to be capable of representing virtually any data model, rather than being a prescription for a single data model. It has fallen short of this ambitious goal for a number of reasons, which this paper investigates. In addition to issues that might have been anticipated in its design, a number of new issues have arisen since its initial development. These include the need to support explicit feature definitions, incremental update, value-added extensions, and change tracking within large, national databases. It is time to consider the next stage of evolution for SDTS. This paper suggests development of an Object Profile for SDTS that would integrate concepts for a dynamic schema structure, OpenGIS interface, and CORBA IDL.

1 INTRODUCTION

At the time of its approval as a U.S. national standard, many within the GIS community had high hopes that the Spatial Data Transfer Standard (SDTS) (NIST 1994) would prove a workable vehicle for the transfer of geographic data between disparate producers and users employing a wide range of vendor systems. Although the marketplace has been slow to accept and support SDTS, there is much about the standard that justifies these hopes. The SDTS addresses the varying levels of data abstraction necessary for meaningful data exchange from the highest conceptual levels, through logical data structure, to physical encoding. When used to its full potential, SDTS will allow an extremely wide range of data models and user-defined geographic data to be encoded and transferred without information loss. Furthermore, the "umbrella" and modular manner in which the SDTS is specified, and the methodology of defining profiles of SDTS, affords the means for the evolution of SDTS as requirements and technology change.

This paper presents a brief description of SDTS' history and architecture, followed by discussion of several issues limiting its widespread use. These issues span a wide range of factors, including differences in technical agendas among data producers, GIS vendors, and users; shortcomings in the design and documentation of SDTS; and aspects of the standards process itself. The paper then discusses likely effects on SDTS that could result from the dramatic changes taking place in computer and communications technology and usage patterns, such as the proliferation of World Wide Web technology and the emergence of OpenGIS™ (OGC 1997; OpenGIS is a trademark of the Open GIS Consortium). Important new requirements include the need to support incremental updates and value-added (non-standard) extensions to large, national databases. Besides keeping such extensions synchronized with standard updates to the databases, there are additional issues of tracking the lineage of changes such as splits and merges from editing of features and topological primitives, and of managing feature maintenance among distributed sources.

With all these issues in mind, we suggest the beginnings of a solution by drawing on concepts from OpenGIS, as well as the Interface Definition Language (IDL) of the Common

Object Request Broker Architecture (CORBA™) (OMG 1996; CORBA is a trademark of the Object Management Group). We also introduce the notion of *dynamic schema* from development of the new DLG/F feature-based vector data product (Hair et al. 1997) within the U.S. Geological Survey (USGS). These are all brought together in our proposal for development of an Object Profile that would open the next generation of SDTS products.

2 OVERVIEW OF SDTS HISTORY AND DESIGN

Work on a national spatial data transfer standard was begun in the 1980's by the National Committee for Digital Cartographic Data Standards (NCDCDS). This committee consisted of representatives from universities, government agencies, and private industry, and was authorized by the American Congress on Surveying and Mapping (ACSM). In the late 1980's exchange standards work of the Federal Interagency Coordinating Committee on Digital Cartography (the forerunner to the Federal Geographic Data Committee) was merged with the NCDCDS work by a task force drawn from both groups. This resulted in the Spatial Data Transfer Standard (SDTS). In 1992, the National Institutes of Standards and Technology (NIST) approved SDTS as a Federal Information Processing Standard (FIPS) Publication, FIPS PUB 173, and designated the U.S. Geological Survey as the maintenance agency. In the same year, a special issue of the journal, *Cartography and Geographic Information Systems*, presented articles describing various aspects of SDTS, including (Davis et al. 1992, Fegeas et al. 1992, Wortman 1992). Additional guides have been produced more recently, including (FEDSIM 1996a,b,c). Recently, the U.S. Federal Geographic Data Committee (FGDC) has taken the place of NIST in governing the approval process for changes to SDTS. By the time of this publication, the American National Standards Institute (ANSI) committee for geospatial standards will have approved SDTS as an official ANSI standard. Clearly there is national level interest in SDTS, even though it is still maturing.

The remainder of this section summarizes the principal semantic levels, dataset modules and profiles of SDTS. However, it is beyond the scope of this paper to fully describe the SDTS structure. Readers seeking a more comprehensive description are directed to the SDTS web site (USGS 1997).

2.1 Semantic Levels in SDTS

At the conceptual level, the SDTS defines a hierarchy of spatial objects (primitives and aggregates) with varying levels of topology, a methodology for defining feature and attributes, and a model of the representation of features by spatial objects. Many geographic user communities have one or more feature/attribute catalogs, such as for ground transportation, ocean navigation, and electric utilities. The standard lays a foundation for defining such catalogs, and even specifies an initial catalog of topographic and hydrologic terms, while allowing for the substitution of other catalogs.

At the logical level, flexible structures are defined independently of physical encoding, but in sufficient detail to allow automated decoding of complex source data models (with at present some notable ambiguities detailed below). These logical structures can be expanded or tailored to specific requirements. SDTS defines a number of *modules* to hold different logical parts of a dataset. Module types are specified for information global to a dataset (e.g. catalog, spatial reference, data dictionary and other metadata), as well as for spatial objects (vector, raster and composite), non-spatial attributes, graphic representation, object-to-object references, and data quality measures.

SDTS specifies another standard, ISO 8211:1985 (ANSI 1986), for physical data encoding. A complete SDTS dataset is encoded as a set of ISO 8211 formatted files, generally one file per SDTS logical module (Altheide 1992a,b, Lazar 1992b). As mentioned previously, however, this encoding is specified independently of the conceptual and logical levels. This can allow substitution of another physical encoding methodology if necessary (e.g., the more recent

edition of ISO/IEC 8211:1994 (ISO 1994), or the Vector Product Format (VPF) (NIMA 1996)). See (IHO 1996) for an international data standard that uses ISO 8211:1994 for its physical encoding specification, in this case for ocean navigational charting.

2.2 SDTS Profiles

SDTS can support any data model, including raster, planar and linear network, Census' TIGER (Davis et al. 1992, Szemraj 1992), and many others; and can be adapted as requirements demand. However, SDTS allows such a wide range of spatial objects and conceptual data model relationships, that further rules are required to constrain the objects and relationships to a coherent set representing a particular geographic domain's digital data model. This is not a defect but part of the intentional design. The primary means for adapting SDTS to a particular domain is through the specification of an SDTS *profile*. A profile is intended to be a clearly defined, limited subset of the SDTS entities, designed for use with a specific type or model of data (e.g. topologically structured vector data, raster data, point data, network data, etc. (Greenlee 1992, Lazar 1992a,1996)). To limit options to only what is sufficient and necessary, specific profile choices are made for encoding possibilities that were not addressed, left optional, or left with numerous choices within the SDTS. A profile can modify the standard, if such modifications are considered important enough. Profiles also provide the basis for conformance testing of SDTS translators and datasets. While data producers can theoretically build an SDTS dataset without regard to a standard profile, there is no assurance that commercial SDTS translators will understand the semantics of such a dataset in order to make widespread use of it.

The Topological Vector Profile (TVP) (NIST 1994, Part 4) was the first profile to be approved. This profile limits a dataset's use of SDTS spatial objects to those which can represent planar geometry and topology. The TVP also qualifies and extends the standard, such as to indicate a means of specifying cardinality of relationships between spatial objects and attributes. A Raster Profile is soon to enter the FGDC approval process as Part 5 of SDTS, following much work harmonizing the profile with other raster standards efforts (RTRB 1997). A Point Profile (FGCS 1996) was formally approved by FGDC in October 1997 as Part 6 of SDTS, to support high-precision geodetic control point data. A draft profile for Computer Aided Design and Drawing (CADD) data is now well along in the FGDC approval process (TSCGTC 1997). Other vector profiles are on the horizon: a Transportation Network Profile (VNTSC 1996) for representing linear networks is in development; and a profile addressing non-topological vector data has recently begun active development. The U.S. National Imagery and Mapping Agency (NIMA) had begun development of a Vector Relational Profile to harmonize NIMA's Vector Product Format (VPF) standard with SDTS, but this has been interrupted due to reorganizations within NIMA. Several other profiles are under consideration. For more details, see (Hickman 1997; USGS 1997).

SDTS has much on which to build. It can accommodate virtually any data model. The standard provides for many-to-many relationships between features and spatial objects (primitives) and provides for feature-to-feature relationships. It allows the encoding of multiple, scale-dependent graphic representations for features, not just intrinsic geometry. Fine granularity of security classification and data quality information on feature data can be encoded. SDTS provides a means of defining multiple levels of containment hierarchy (e.g., themes and surfaces, see DLG-E (Altheide 1993)). A similar approach with the same module type could be used to define temporal groupings of feature attributes and spatial objects. And many separate datasets can share use of a single directory containing the master data dictionary. Time and technology, however, do not stand still. SDTS needs to evolve further.

3 CURRENT SDTS CONSTRAINTS

The initial promise of SDTS was that it could serve as a general-purpose, GIS vendor-independent intermediary for transferring geographic data between differing GIS vendors'

software. The preceding discussion makes it clear that SDTS continues to be developed with input from a broad community, and that it addresses a number of important issues in the representation and archival of geographic data. However in the years since it was first published, the marketplace has seemed to be slow to accept and use this standard. Government and commercial data producers have been slow to migrate their many different geospatial databases to SDTS format; and GIS vendors have been slow to build and market software tools for translating between their own proprietary formats and SDTS. Certainly there is a simple economic condition at play here: the data producers must first stimulate demand for SDTS by providing large quantities of useful data in that format. The resulting customer demand should, in turn, motivate GIS vendors to develop the conversion utilities needed to bring such data into their proprietary data formats for query and analysis. However, the data producers largely depend on the same commercial GIS software to build their databases, and so must motivate the vendors to support vendor-independent formats in advance of market-based demand. Some delay in building up both the supply and demand for SDTS data might be expected in these conditions. But the issues lie deeper than this.

During the past two years, as the authors brought their respective GIS vendor and U.S. federal data producer perspectives to bear on these issues, certain barriers to the initial promise of SDTS have become apparent. These barriers are the result of a number of factors:

complexity of SDTS;
slowness in the development of practical SDTS profiles;
restriction of each SDTS dataset to a single profile;
lack of a clear definition of geospatial features in SDTS; and
ambiguity in the means of specifying cardinality of relationships in a data model.

Additional requirements have arisen as well, that were not anticipated at the time of SDTS' development. These represent substantial technical challenges, as well as the need for increased international focus within SDTS:

a standard means of representing subtiles within a dataset;
support for permanent, universally unique object identifiers across all datasets;
support for value-added extensions to standard datasets by users;
support for tracking changes and historical lineage of features and spatial primitives;
harmonizing the metadata content requirements in SDTS with emerging international standards; and
harmonizing the SDTS repository organization with vendor-independent OpenGIS software interfaces now emerging.

To complete this list, it should be noted that SDTS was designed in such a way that any conformant dataset would be *self-describing*; that is, its data model could be determined from the dataset contents. However, this continues to prove an elusive goal. An unambiguous means of specifying the complete semantics of a complex geospatial data model has yet to be developed. For example, how can a dataset encode the assertions that distinguish roads from rivers, in such a way that GIS software can understand and enforce such assertions without human intervention? We will discuss each of these concerns in the following sections.

3.1 Complexity of Geospatial Data and Metadata Requirements

A basic question is often asked, "Who really needs SDTS, and why?" The needs driving the design of SDTS are primarily those of large national-level data producers, who are continually seeking better ways to represent the complex interdependencies and other relationships of real-world entities than GIS vendors seem to have implemented. The demands of national civilian and defense mapping agencies for very large databases, high precision, flexibility of data modeling, extensive metadata (FGDC 1994; CEN 1997), and collaborative update capability among multiple users, stretch most GIS software systems beyond their practical limits of use. These producers also seek to document and manage the design and quality assurance at each stage of the geospatial data collection and update processes. Such requirements go well beyond the casual "desktop GIS"

user's needs of today, and even beyond the needs or resources of most government and commercial GIS projects at the local or regional level. Looking back historically, there seems to have been a continual gap between the evolving needs of the large national data producers and the evolving capabilities of GIS software.

SDTS was designed to support complex data models which are not adequately served by commercial vendor-defined CAD and GIS exchange formats. Perhaps it should be no surprise, then, that its own complexity is one of the most common issues cited by data producers and GIS vendors having difficulty understanding SDTS. This also could be said of DIGEST, NIMA's National Imagery Transmission Format (NITF), or International Hydrographic Organization's S57, to name some of today's most complex data transfer standards. Each of these standards requires a significant investment to build and maintain understanding, expertise and software within a data producer or GIS vendor organization. But is this simply a labor and educational challenge, to be fixed by providing better training, clearer documentation and examples?

While granting the need for better documentation and education, the authors feel that SDTS does indeed impose a qualitative increase in complexity over many previous vendor-independent transfer standards, such as Digital Line Graph (DLG), Digital Raster Graph (DRG), Digital Elevation Model (DEM), and Digital Orthophoto Quadrangle (DOQ). These are older standard data formats used by USGS that could all be represented in SDTS profiles. But the USGS is seeking to describe and manage the conceptual data model within each dataset in terms of expressive and useful object-attribute relationships, object-grouping relationships, feature-attribute catalogs, data domains, spatial reference systems, data security models, data quality models, and various other forms of metadata. These aspects of their conceptual models cannot be well expressed in the older data formats, thus necessitating development of new data organizations. Part of the conceptual model may even include audit information such as the nature of any change to the data, together with the timestamp and identification of the person making the change (see Cox 1996, for extensive research in *evidentiary metadata* that is largely independent of the application domain). It should be noted that the FGDC has approved geospatial metadata content standards (FGDC 1994) which differ from the metadata content specified in SDTS. This will require changes in SDTS to comply with (FGDC 1994).

In addition to this level of complexity in the conceptual data model, SDTS places these further requirements on the data producer:

> to describe and manage the logical model in terms of the module naming conventions, as well as module-to-module cross-references for representing composite entities; and
>
> to manage the physical model in terms of proper usage of the ISO 8211 standard for encoding and decoding the physical SDTS files.

All this represents a significant amount of information to collect and encode for each geographic dataset. Until there is a substantial commercial market demand for this level of data model complexity, most GIS software and utilities vendors have little motivation to develop general-purpose support tools to simplify and automate the processes of capturing and managing this much data and metadata. However, the authors feel the need to manage this complexity is inevitable and growing, as the world-wide sources, quantity, quality and potential liabilities of geographic data continue to increase. With national and global warehouses (repositories) and clearinghouses (catalogs of pointers to repositories) of data emerging for use in emergency management, defense and other large-scope initiatives, it will only become more important in the future to manage the complexity of fully describing these distributed and heterogeneous geographic databases. Furthermore, as these enormous resources become increasingly available to the public, the "desktop GIS" user will want access to them. Different levels of tools can help reduce the apparent complexity of this data; most users will not need the ability to produce the data, just to search, display and analyze it. Even for many analyses, it will be important to understand the composition, lineage and quality of heterogeneous sources of data. This degree of

complexity is substantial and not required by all GIS users, but it will only continue to increase and must be managed.

3.2 Slow Development of Profiles

An important reason for the slow acceptance within the marketplace for SDTS has been the lack of appropriate profiles for many applications. The TVP was the only one approved in the first four years of SDTS' release. Furthermore, TVP represents one of the most complex profiles of all, requiring planar topology to be established throughout the dataset. This has restricted the development of vector-based encoders and decoders to the domain of the more sophisticated and costly tier of GIS software (Kelley and Gosinski 1994).

There are a number of reasons behind this slow development. In the early stages of SDTS development, its complexity and limited educational material inhibited the understanding needed by the geographic community at large to engage in building the necessary profiles. Furthermore, a number of vendor-specific exchange formats were in widespread use that satisfied many users' needs. However, at least two profiles had additional reasons for slow development. The Raster Profile (RP) was almost finished when a strong request was made that it support the Basic Image Interchange Format (BIIF) in use by NIMA and NATO. The RP sponsors are also now considering incorporating the GeoTIFF ad hoc standard, as well. Similarly, the Transportation Network Profile (TNP) was interrupted by a number of industry and government requests to incorporate the Geographic Data File (GDF) format, which is a European CEN/TC 287 feature-based geographic data standard widely used in the Intelligent Transportation Systems (ITS) industry. Other forces are also at work with TNP, and there are several possible directions this development may take (see Hickman 1997).

This highlights the risk in advancing profiles too quickly, which can result in inadequate consideration of all the constituent geographic communities' interests. The consensus process among numerous international industry and government organizations adds greatly to the time and complexity of developing profiles. However, this consensus appears to be building. As a result of continued efforts from USGS to migrate its DLG-3, DEM and DOQ products to SDTS, and to encourage broader participation in SDTS development, there appear to be increasing efforts in the geographic community to complete the profiles needed most, such as raster, non-topological vector, and linear network.

3.3 Support for Multiple Profiles in a Single Dataset

There is some disagreement as to the benefit of having numerous, narrowly focused profiles, rather than a single profile that can support multiple levels of topology and models of data within a single SDTS transfer dataset. Geographic datasets are growing rapidly in size and complexity. It is not reasonable to expect a single topological model to serve adequately for all thematic coverages in a complete dataset (Kelley and Gosinski 1994). For example, one theme or coverage might contain a reduced-resolution representation of the database with only a few feature classes and no topological structure. Another coverage might contain transportation or utility system features with linear-network topology. A third coverage might contain political and property boundaries with planar topology. A fourth coverage might contain a matrix of geodetic control points requiring no topology, but needing very high numeric precision. Still another coverage might be a raster backdrop for the vector data. At present, the only approach for transferring all these coverages is to place them in separate SDTS datasets, each with a different profile designed for the level of topology or precision required. The first problem is that not all of these profiles have even been developed yet, and so the linear vector data models must all be coerced into TVP. Another problem is that, according to SDTS, each dataset can only represent a single profile. Thus, information about the spatial reference system and registration must be replicated in separate datasets.

Even if all possible vector profiles were developed, and encoders/decoders were available, there would still be the problem that a significant amount of data may have to be

replicated for feature classes appearing in multiple coverages, resulting in synchronization issues whenever the data is updated. It also means that references between objects of different profiles cannot be made, since object references cannot span datasets, according to SDTS requirements.

There are a number of possible approaches for resolving this issue, including:
design a profile that accommodates multiple levels of topology;
modify SDTS and all profiles to support multiple profiles within a single dataset;
modify SDTS to support module-record references that span datasets; or
treat each SDTS dataset as a single coverage, and create a level of hierarchy above the dataset directory, which would contain and identify all associations between the contained datasets.

Each of these approaches has advantages and disadvantages. One important requirement for future multi-profile designs should be to allow decoders to determine which profiles are incorporated in a dataset, as well as to choose the scope of data to decode based on the decoder's capabilities. Decoders should not be expected to understand all aspects of every multi-profile/multi-model dataset, but they should not crash either. Encoder/decoder conformance guidelines from USGS would also have to be changed to reflect this.

3.4 Precise Definition of Geospatial Features

While the SDTS design provides some support for describing geographical feature entities, the documentation and data model are strongly based around spatial objects. In SDTS, a *feature* is not an explicit construct, but one of many possible entities implied by grouping spatial objects together and linking them with a set of non-spatial attributes, using the SDTS *composite module*.

The composite module has been overlooked or misunderstood by many trying to learn about SDTS, largely due to inadequate documentation. From its structure, documentation and initial intent, it could be primarily considered as a means of combining multiple spatial objects into composite spatial objects. However, this module can also serve as a general-purpose "join table" for associating and grouping records of any SDTS module in a flexible way. It can be used to establish one-to-many, many-to-one, or many-to-many relationships among spatial primitives, composite objects and sets of attributes. As such, this module can be used to represent a wide variety of feature collections, hierarchies and interrelated object structures in addition to geographic features. For example, the enumeration of all spatial objects for each of several thematic layers could be held in a composite module. A composite module also could be used to group temporally related sets of spatial objects, such as hourly increments of shoreline data to reflect tidal variations. However, none of the standard documentation mentions these possibilities.

A result of this oversight is that neither SDTS nor TVP provide any guidelines for the consistent separation of point, line and area features into distinct composite modules. Some GIS software is capable of treating all these feature types consistently (i.e., encode or decode all feature types using a single composite module), while other GIS software must process each feature type differently and thus requires separate composite modules to be used for each. Similarly, the use of composite modules to describe thematic layers and other kinds of feature collections in a consistent manner is left unspecified. Because of these ambiguities, *TVP-compliant datasets can incorporate idiosyncrasies of the encoder's GIS architecture, which may not be readable with another vendors' TVP-compliant decoder*. In this case, it seems more appropriate for the profile than the standard to specify the usage of composite modules, but any such specification must be considerate of other profiles' needs and conventions as well, for instance to avoid module, field, and subfield naming conflicts.

It should be evident by now that the composite module can serve many purposes. This represents important flexibility to support alternative data models. But without a means of identifying the purpose of each relationship in a data model, and without guidelines for implementation of certain widely-used relationships (e.g., point, line and area features), *SDTS falls*

short of its goal of providing a fully self-describing form of dataset. The spatial object modules encode topological relationships among spatial primitives, while the catalog/cross-reference module was designed to represent module-to-module relationships. But the catalog/cross-reference module structure is too simplistic for unambiguously representing all the relationships that might exist between features or feature classes implicit within the composite modules, attribute modules and spatial object modules *in a way that is directly useable by a decoder*. More formally specified entities and their relationships need to be defined and included within each dataset. Profiles could then choose to limit the scope of relationships according to their purposes.

3.5 Specification of Cardinality of Relationships

A related problem with the present design is that a given SDTS encoder's *intent* to use one-to-many, many-to-one, or many-to-many relationships between attributes and spatial objects is inadequately declared within a dataset (see Arctur 1996b for a humorous treatment of this and related issues). It is important for decoders to be able to anticipate the kinds and cardinalities of relationships embodied in a given dataset, without having to read the complete dataset looking for clues. This issue is not addressed in SDTS Part 1, and is left for each profile to specify. The TVP document (NIST 1994, Part 4, Section 5.3) provides a partial solution, specifying a flag "JJ" to be placed in the beginning of the comment subfield of the catalog/cross-reference module if there are one-to-many or many-to-many relationships between attributes and spatial objects. The case of many-to-one relationships is not addressed, presumably based on the assumption that no join table (i.e., a composite module) is needed to represent the association of an individual spatial object with many attribute records. It is inadequate and ambiguous to use a single flag in this way to represent two possible data models out of four. Furthermore, this should not be left to each profile to specify; it seems more prudent for the *standard* to require explicit identification as to which of the four cardinalities of relationships (1:1, 1:N, M:1, M:N) is embodied within a dataset. Each profile may then limit which of these are to be used.

3.6 Subtiling Issues

When SDTS was being developed, a single "map sheet" (such as a USGS DLG-3 quadrangle) or area (such as a county in the Census Bureau's TIGER system) was considered the baseline digital data model to represent. Typically, each such dataset was limited to a single tile in the sense that its boundaries were constrained, with no further areal subdivision. Multiple sheets or areas can be encoded within a single SDTS dataset, but the standard has no predefined mechanisms for organizing tiles or relating objects across tile boundaries. The spatial extent of a tile may be defined either by a hardcopy map from which it was derived, or by the need to have a fully integrated dataset within some administrative, political, or otherwise predefined boundary. This model is sufficient for datasets that are inherently manageable in size as a single tile. However, as SDTS achieves use beyond the scope of these relatively simple data, it will become necessary to support datasets that have tiled subdivisions within them (we will call these *subtiles* to avoid confusion with other meanings of *tile*). The main reasons for using subtiles are to help manage the tasking, storage and processing for data collection, update, and distribution of very large coverages.

For example, the National Imagery and Mapping Agency (NIMA) has several databases of national and worldwide shorelines, political boundaries and other thematic coverages. NIMA has developed a uniform scheme based on latitude and longitude by which it hierarchically subdivides the entire world into rectilinear tiles. These tiles can be square or rectangular, and vary in size from 15'x15' to 30°x30° depending on the scale of the digital data library. The tile naming convention is similar to that used for the Universal Transverse Mercator (UTM) coordinate projections. (See (NIMA 1996) for more information.) Models of non-rectilinear tiles also exist that treat the planet as a polyhedron with triangular, pentagonal or hexagonal facets (tiles) which may be hierarchically decomposed as well (Dutton 1991, 1992; Fekety 1990). However, the real world is structured in terms of objects or fields (phenomena with continuous change of properties).

A better approach thus may be a feature- or field-based approach. In this case, the tiling problem would be a problem of structuring features or fields. Clearly, there are many choices available.

When subtiles are used, they represent bounds on the topological consistency of the spatial objects. So far, all this could be accommodated within SDTS' present design, by judicious use of spatial object and composite modules to delineate the boundaries of subtile regions. However, since there is no consideration of subtiling in the design of SDTS or any of the profiles, any number of different subtiling conventions may emerge from different data producers, each requiring special consideration by SDTS encoder/decoder software.

A related issue is that of establishing and maintaining cross-tile topology. For instance, when a linear feature such as a road or river crosses a subtile boundary, it would need to be split into two topological links at the boundary, connected by a topological node. Similar work is needed when an area feature such as a lake crosses a subtile boundary, resulting in the creation of topological links at the subtile boundary to split the lake feature. This would most likely be done by the GIS software responsible for creating the transfer dataset, rather than by the SDTS encoder itself. To rebuild the complete linear feature again after decoding the subtiled dataset, the target GIS software could either:

> perform costly edge-matching of topological elements along subtile boundaries; or follow references associated with the links on either side of the subtile boundary that could have been stored in the transfer dataset by the encoder.

The kind of references that need to be stored for cross-tile topology are fairly simple: the tile ID of the neighboring tile, and the spatial object ID of the connected topological element in the neighboring tile. But again, a standard scheme needs to be defined as part of SDTS or there will be a proliferation of ad hoc approaches developed by data producers and individual profiles.

3.7 Universally Unique, Permanent Object Identifiers

SDTS introduced the notion of module record identifiers that could be readily maintained as unique across a single dataset. Each record identifier consists of a module identifier (defined to be a four-letter code, by TVP convention), concatenated with the record number of an element within the module. This allows direct feature-to-feature references, as well as relationships among any types of modules. However, such references cannot span multiple SDTS datasets, according to the current standard. At this point in time, it may not seem conceivable that a general system for managing and referencing objects among completely different datasets would be necessary or even feasible. However, it seems to the authors just a matter of time before the interconnectedness of the Internet web makes possible and useful all manner of alliances among databases and objects.

An important first step in achieving this goal is just to support the notion of permanent identifiers within SDTS datasets. In other words, once a dataset-unique identifier has been assigned to an attribute record, spatial object, composite or other module record, *it would never be changed*, except in certain restrictive cases. This requirement is essential to enable incremental updates and value-added attribution to maintain synchronization with the base data, an issue which will be discussed further in section 3.8 below. Permanent record identifiers would also enable versioning and historical lineage tracking of a feature or other object through editing operations, which will be discussed in more detail in section 3.9 below.

Assuming that record and object identifiers can be readily maintained as unique within a single SDTS dataset, some additional mechanism is needed to provide a dataset-level identifier that is unique across all SDTS datasets—and why stop with SDTS datasets? Using the current practice for managing unique Internet domain names and web page URLs as a model, it should be feasible for some organization to maintain a global registry of data producer domains. Each producer would then be responsible for the URLs (or universal resource names, URNs) of their geospatial datasets. If this proves feasible, the capability for managing globally unique identifiers would also provide a means of referencing a single geographic feature or other entity in multiple profiles (datasets), to address an issue raised in section 3.3 above. (An interesting development in

Internet-based storage and data conversion is the Open Geospatial Datastore Interface, OGDI. See http://www.las.com/ogdi/.)

3.8 Incremental Updates and Value-Adding

Presently, an SDTS dataset is expected to be re-created in its entirety, each time any change in the dataset needs to be posted. This is because an SDTS dataset is a passive repository, not a vendor-specific database incrementally maintained by a running software program. Producing the dataset in full just to post updates can impose considerable overhead, both in time and in computer resources. A useful capability which has been incorporated in the design of IHO's S57 version 3 standard, is to support incremental updates to the database. This is partially made possible by S57's requirement for permanent, unique object identifiers (this is discussed further below). NIMA has introduced an approach for incremental VPF updates, but this is handled at a file-replacement level, since VPF does not have unique or permanent object identifiers. The S57 incremental update approach may be well suited to SDTS, as S57 also uses the ISO 8211 physical encoding, which was not designed initially for direct record-level addressing but rather for streaming to/from magnetic tape. This bears further investigation.

A more troublesome issue in the geospatial data community has been the lack of ability to support value-added information in a cost-effective manner and in time-critical applications. There are a variety of needs for value-added information, and many approaches for handling such information. What we mean by value-added information is the extension of the primary database to include additional:

feature classes and their instances;
feature-to-feature relationship classes and their instances;
attributes on feature classes or relationship classes; and
allowable values in the domain ranges of feature and relationship attributes.

Some of the reasons for adding such features, attributes, relationships and values include:

a base of required data exists, but additional information is required for a one-off or transient product, application, etc;
a base of information exists but a long term need arises for additional information that has not been captured in the past; and/or
partnerships have been established based on areas of expertise, to add respective elements to some common and greater set of data.

Where these requirements cannot be supported, the tendency is to acquire a *copy* of a set of data and then add value to that copy. This may result in a new database design effort and frequently provides no way to migrate this back into the primary database without a heavy cost. Yet, if these changes are not integrated back into the primary database, there is a risk of either or both the primary and the copy falling out of synchronization and becoming obsolete. Employing a database design and system design that avoids this would be of great service to the community.

Regardless of whether incremental updates are intended for conveying standard updates or value-added information from an external source, the greatest difficulty is identifying and synchronizing the features from the update transaction with those in the baseline dataset. SDTS does not have a requirement for maintaining permanent, immutable object identifiers in the dataset, so when an update transaction occurs, the updated data may have completely different identifier numbering from the previous version. This reduces synchronization of the updates with the baseline dataset to a potentially difficult conflation process (merging both datasets and then resolving duplicate features, which can be very labor-intensive and error-prone).

Further complicating the picture is the fact that responsibility for the data might be distributed across multiple sites. For example, the USGS could be holding a national database of rivers and streams, while the Environmental Protection Agency (EPA) is actually responsible for all updates to this data. Meanwhile, the U.S. Army Corps of Engineers might wish to use this data, and add feature classes and attributes for its own analyses. It would be useful and important to find

a way to provide all data production partners and users with a synchronized view of the data that was pertinent to their needs.

3.9 Feature Lineage Tracking Through Splits, Merges and Other Edits

Still another consideration we wish to take into account is the maintenance and tracking of feature identity during updates. Maintaining precise lineage information enables value-added vendors, partners, etc. to more easily maintain their data as updates are propagated between agencies and to users who maintain their own data holdings. For example, suppose one user is editing road features while another user is editing waterway features. Let us assume one particular road, Adam Street, has a "crosses-over" relationship with one of the streams, Baker Creek. If the person editing roads splits the Adam Street feature to account for a new intersection, this might change the feature identifier of the portion of Adam Street that crosses over Baker Creek. If, during the same transaction period, the person editing streams makes a correction in the course of Baker Creek, then these edits could potentially fail to resolve their feature-to-feature references if they cannot take into account the lineage of change in Adam Street's identity. A similar issue with identity tracking occurs when two or more features are merged into one. It may not be the responsibility of the transfer dataset encoder or decoder to perform lineage tracking, but the dataset should at least have the semantics to represent historic lineage of features.

4 INTERFACES AND FRAMEWORKS FOR THE FUTURE

We have presented all the issues we intend to take into account for now, as we work toward the next generation of SDTS. We will now present concepts and technology to address these issues.

4.1 Emergence of OpenGIS™

The Open Geospatial Interoperability Specification (OpenGIS) is under development as a collaboration of GIS and utility software vendors, system integrators, data producers, and academia known as the Open GIS Consortium (OGC, see http://www.opengis.org). OpenGIS represents a set of requests, services, and data structures to be supported by geospatial data clients and servers. The intent is that any software vendor's OpenGIS-compliant client application (e.g., a user's map window on the computer screen) will be able to access any geospatial dataset via another software vendor's OpenGIS-compliant data server software. This will be made possible through GIS industry-defined extensions to CORBA (OMG 1996), Microsoft's Object Linking and Embedding/Component Object Model (OLE/COM), and Microsoft's Open Database Connect (ODBC) protocol. These are distributed computing platforms that enable the transfer of requests and services between disparate vendors' client and server software. The OpenGIS Abstract Model (OGC 1997) defines the reference model on which this interface specification is based. OpenGIS extensions to CORBA, OLE/COM and ODBC do not represent changes to the respective base standards, but rather GIS industry-based consensus for usage of those standards. However, emerging relationships between OGC, OMG and ISO appear to be leading to the result that OGC standards of usage will become draft international standards, which can then be directly approved by OMG and ISO. OpenGIS specifications, as well as compliance testing and certification, will be defined and incorporated into GIS software and utilities in stages over the next several years.

SDTS serves in a complementary role to OpenGIS, by providing a standard format for transfer and archive of geospatial data that is independent of any particular GIS software. This is especially important to national-level data producers wishing to store and distribute large repositories of data in a consistent structure that are developed among many agencies and sources. And even after OpenGIS has been implemented, SDTS may be of use for data transfer in cases where OpenGIS-compliant clients or servers are not available. However, OpenGIS-compliant servers could incorporate both encoder and decoder capabilities for SDTS, and might even perform incremental updates to an SDTS dataset. Possibly the greatest strength of SDTS is that it

provides a means of storing datasets in a vendor-neutral format that may be decoded at any time in the future—as long as it is truly self-describing. Possibly the single most important (and most difficult) requirement for a geospatial data archive format is that it encapsulate a complete and unambiguous description of its data model structures and relationships. This would allow it to be decoded and understood years later when the original data specification may no longer be known or available. However, this degree of self-description remains an elusive goal today.

There are a number of differences and gaps between OpenGIS and SDTS, in terms of the structure and semantics of metadata, security, spatial reference systems, spatial objects, attributes, composites, and graphic representations. In some cases, SDTS has defined entities that OpenGIS has not yet addressed. In other cases, OpenGIS has gone beyond the definitions in SDTS. Some work to reconcile these differences will reduce the potential for information loss during transfers, which would benefit both producers and users of geospatial data.

4.2 Emergence of DLG/F

In recent years, USGS has made progress toward defining a feature-based model for geospatial data, DLG/F. This work was driven by the need to overcome limitations in the DLG-3 specification, as well as to begin supporting distributed feature maintenance as it has been described so far (Hair et al. 1997). DLG/F represents an evolutionary step beyond the georelational model that SDTS was initially designed to support, with enhancements such as:

database-unique, immutable object IDs;
basic and compound features, with explicit feature relationships;
dynamic schema (schema is part of each data module);
object versions and lineage tracking through splits and merges; and
transaction information that can be stored in the dataset, to support varying models of long transactions and locking, depending on data server capabilities.

Database-unique identifiers are already possible for basic and compound entities with SDTS, by combining the module name and record identifier fields. But these are not guaranteed to be immutable from one version of a dataset to another, preventing data production partners from developing effective value-adding strategies for the data.

In order to support lineage tracking, the DLG/F production system will retain historical data while accepting and applying updates to a database. As an update is applied, existing data must be kept intact and become the predecessor of the update. These would become versions or generations of the same datum, allowing historical tracking of updates. For some period of time, historical versions must be directly accessible in the operational feature database. As updates would now be required and applied at the feature level, tracking must therefore take place at the feature level instead of the map quadrangle or file level as has been the case historically. The production system would be responsible for feature identifier generation and lineage tracking through splits and merges with other features.

One of the more intriguing aspects of DLG/F is its approach for a *dynamic schema*. One of the motivating factors for this is to enable transfers in which a dataset would contain just the data requested by a user. For instance, if the user requests or updates a subset of a feature's attributes, then the resulting transaction dataset would include only those attributes. The means of achieving this involves transforming the normal table-like organization of data into a more flexible structure in which a feature consists of: (a) one row in a feature table (Figure 1 below), and (b) zero, one, or more rows in an attribute table (Figure 2). This allows the base database schema to remain the same while the actual content may vary dramatically over space, time, etc.

Note that Figure 2 below includes attribute values only for the Stream feature (Obj-id 1). The Stream feature is the "owner" of the first two attributes listed, and the Stream's Elevation attribute (Obj-id 3) is the owner of the last two attribute values (Stage and Source). This is due to DLG/F's design, which allows feature attributes to have attributes. In this example, the Stream's

Elevation value of 300 has been further qualified as the High-water Stage, and based on a Field Source.

Obj-id	Feature-type	Dimension
1	Stream	1-dimensional
8	Lake	2-dimensional
10	Road	1-dimensional

Figure 1. Dynamic Schema: Feature Table

Obj-id	Owner-type	Owner obj-id	Attribute-name	Value data-type	Value
2	feature	1	Name	A	Potomac
3	feature	1	Elevation	N	300
4	attribute	3	Stage	A	High-water
5	attribute	3	Source	A	Field

Figure 2. Dynamic Schema: Attribute Value Table

Figure 3 shows a sample table of relationship descriptions. The Obj-id in this table is each relationship's unique ID. The first row in this table states a "flows-to" relationship between the Stream and the Lake features from Figure 1. Similarly, the second row states a "crosses-over" relationship between the Road and the Stream features in Figure 1.

Obj-id	Obj-1-type	Obj-1-id	Relationship-type	Obj-2-type	Obj-2-id
20	feature	1	Flows-to	feature	8
28	feature	10	Crosses-over	feature	1

Figure 3. Dynamic Schema: Relationship Table

This is a simplistic example not intended to fully define DLG/F. DLG/F uses the dynamic schema approach to define other object types as well (themes, data products, and digital update units or DUUs). Spatial coordinates of point, line and area features are handled in the same way as any other feature attribute. DLG/F does not require topological relationships between feature geometries to be stored, but expects the grouping of feature classes within a theme to represent a unit of topological consistency (the GIS software used for data production would enforce topology as a means of "cleaning" the data). The dynamic schema approach is used to describe a given data product's complete structural specification in terms of these various object types and their relationships. This approach has a significant flexibility and advantage over SDTS: changes to the domain data model can be made without changing the table structures that define them (e.g., feature attributes can be added or removed without changing the schema table structures). It also affords a granularity of definition and access to features, attributes and relationships that is much more expressive than with SDTS. Furthermore, by treating geometry as just another attribute of a feature, this approach allows the definition and storage of multiple geometries for a given feature. One of these might be the accurate locational geometry, while others might be used for scale-dependent cartographic representation.

An obvious aspect of the dynamic schema approach shown in Figures 1-3 is that performance of queries, analysis and display could be seriously reduced, in comparison with an efficient GIS optimized for these operations. Indices to accelerate searching or sorting based on object types and attribute values would be difficult to maintain. However, the purpose of this

schema approach is to support data warehousing, not GIS analysis. It is expected that a user would move data from the DLG/F repository into an efficient GIS for practical use.

In order to address distributed feature maintenance issues mentioned earlier in this paper, USGS is working to manage "long transactions" for updates to the data from remote sites. This would involve checkout and checkin of feature sets from a central database by authorized users. The semantics of handling such transactions is part of the role of DUU objects. What is contained within an update transaction (i.e., the nature of the request and any associated data) is defined by an emerging specification called the Feature Communication Protocol (FCP; see Timson 1996). FCP is, in fact, a precursor to the OpenGIS specification, and will be changed to be consistent with OpenGIS as it matures.

4.3 Enabling Incremental Update and Value-Added Information

The dynamic schema approach in DLG/F provides excellent support for incremental updates and value-added feature classes, attributes, relationships, and other object types. These may be created within the original database or within auxiliary tables. Auxiliary tables may be especially well suited where compound features are being established or where basic features have minimal interaction with data in the primary or original database. An auxiliary database may also be well suited where the value-added information is very transient in nature.

Figures 4 and 5 provide an example of the ease with which instances of a new feature class and its attendant attributes can be added to the database. There are no required physical changes, thus enabling very quick turnaround and complete integration with existing data holdings. These could be added to the primary database or to an auxiliary database if there is not a long-term need to retain these data.

In this example, a feature with Obj-id 100 and a type of "some type" was added to the database along with two attributes that define that feature. The intent here is to illustrate that a feature of any type can be added and the attributes may have any name.

Obj-id	Feature-type	Dimension
1	Stream	1-dimensional
8	Lake	2-dimensional
10	Road	1-dimensional
100	**"some type"**	**1-dimensional**

Figure 4. Value-Added Feature Table

Obj-id	Owner-type	Owner obj-id	Attribute-name	Value data-type	Value
2	feature	1	Name	A	Potomac
3	feature	1	Elevation	N	300
4	attribute	3	Stage	A	High-water
5	attribute	3	Source	A	Field
101	**feature**	**100**	**"some attribute"**	**x**	**"value"**
102	**feature**	**100**	**"some attribute"**	**x**	**"value"**

Figure 5. Value-Added Attribute Value Table

New relationship classes may also be established. These may be created within the original database or within an auxiliary database. Figure 6 provides an example of potentially any relationship being established between two features. By defining some new class of relationship the instances can be stored in the database with no impact on its design.

Obj-id	Obj-1-type	Obj-1-id	Relationship-type	Obj-2-type	Obj-2-id
20	feature	1	Flows-to	feature	8
28	feature	10	Crosses-over	feature	1
30	**feature**	**90**	**“some relationship”**	**feature**	**100**

Figure 6. Value-Added Relationship Table

New attributes may be added to existing feature classes or relationship classes. With the dynamic schema this is very easy to support, and the additional attributes may be stored in either the primary database or in an auxiliary database. Figure 7 provides an example of the ease with which instances of new attributes for existing feature classes and relationship classes can be added to the database. Again, there are no required physical changes, enabling very quick turnaround and complete integration with existing data holdings. These could be added to the primary database or to an auxiliary database if there is not a long-term need to retain these data.

In this example, two new attributes are added having Obj-id 2000 and 2001. The attribute with Obj-id 2000 has been added as an attribute on the stream feature with Obj-id 1. The attribute with Obj-id 2001 has been added as an attribute on the relationship with Obj-id 20.

Obj-id	Owner-type	Owner obj-id	Attribute-name	Value data-type	Value
2	feature	1	Name	A	Potomac
3	feature	1	Elevation	N	300
4	attribute	3	Stage	A	High-water
5	attribute	3	Source	A	Field
101	feature	100	“some attribute”	“some type”	“value”
102	feature	100	“some attribute”	“some type”	“value”
2000	**feature**	**1**	**My attribute**	**A**	**My value**
2001	**relationship**	**20**	**Other attribute**	**A**	**Other value**

Figure 7. Defining New Attribute Values

These techniques make it possible to add either transient or permanent information to a geospatial database with little or no impact in terms of database structures, software design efforts, etc. Also, this reduces the need to make a full “snapshot” of the data in a different database that often results in two diverging copies of the same data.

The concepts that enable these benefits to occur include the dynamic schema, permanent identifiers, and a solid foundation in terms of feature maintenance. These enable a very rapid response to time-critical applications, a tremendous flexibility in terms of a product strategy, and the ability to enable and maintain collaborative value-added partnerships.

4.4 Can An Object Profile Handle All This?

Does SDTS need yet another profile? Can any single profile address all the issues presented so far? Many of these issues could be handled with incremental refinements to the present SDTS. Nevertheless, let us consider what a profile based on object-oriented (OO) technology might include. References for OO concepts and literature are too numerous to cite fully here. However, a good sampling for introductions to various aspects of object-orientation would include: (Bancilhon et al. 1992, Booch 1994, Coplien and Schmidt 1995, Gamma et al. 1995, Goldberg and Robson 1989, Goldberg and Rubin 1995, Jacobson et al. 1992, Rumbaugh et al. 1991, Smith 1991, and Wirfs-Brock et al. 1990). Literature concerned with the application of object-orientation in GIS includes (Arctur 1996a,1997, Arctur et al. 1995, Chung et al. 1995, Cobb et al. 1995,1996, Egenhofer and Frank 1987,1992, Herring 1992, Laurini and Thompson

1992, LSL 1996, OGC 1997, Smith et al. 1987, Tang et al. 1996, Wiegand and Adams 1994, Worboys 1994, and Worboys et al. 1990).

The feature-based models of geospatial data that have been developed for OpenGIS and DLG/F appear reasonably well constructed and complementary. We propose adopting the feature-coverage data model of OpenGIS as the reference model, since DLG/F represents a family of products that could be derived largely from this model. We further propose to enhance the OpenGIS-based reference model as needed to support any aspects or requirements of DLG/F that are not otherwise included (e.g., to support Product and Digital Update Unit object types). The "well known structures" (WKS) of the OpenGIS reference model, augmented as needed for DLG/F, should provide the atomic data types needed to represent spatial attributes of features. The dynamic schema approach would be used to define the data product specification that is embodied within a given dataset, as well as the general form for the remaining dataset content.

Both OpenGIS and DLG/F incorporate the concept of unique object identifiers. However, neither OGC nor USGS have adopted a strategy for defining object identifiers that could be unique across multiple datasets. We propose the method in section 3.7 above, for consideration as a potential means of managing universal object identifiers. Certainly some guidelines must be established, such as the overall size of ID that might be created. But it would be to the entire GIS community's benefit if a simple scheme such as this enabled object-to-object references that could unambiguously span any dataset boundaries.

One aspect of geospatial data models that has not been addressed yet in this paper is that of *behavior* of geospatial entities. In OO systems, objects encapsulate both their structure and behavior (procedural methods or functions). Non-OO GIS software should not be required to support object behavior, but OO-GIS software also should not be prevented from handling such support. An example of geographic object behavior that would be useful to transfer might be Encapsulated PostScript, OpenGL or some other means of expressing cartographic representation in a computer platform-portable manner. Because of the proliferation of different languages which might be used for object behavior, we propose to utilize the approach in CORBA IDL (OMG 1996) to describe *method signatures*, by expressing IDL's method signature parameters in dynamic schema terms. These would be implemented in SDTS as a separate module, and could be made an option of the Object Profile.

The OpenGIS and DLG/F data models each specify a certain set of object types. Other data models will no doubt create different object types; thus the set of SDTS modules needed to represent all data models' specifications should not be expected to stay fixed. For the most benefit, the standard needs to be *self-extensible*, and the dynamic schema approach seems well suited to implementing SDTS in such a way that it could support its own extension in a computer-decodable manner. Once the most appropriate method for doing this is determined, it should be possible to define any of the data models for which other SDTS profiles have been developed, including raster, point-coverages, transportation/utility system networks, and CADD. It should also be possible to define a hierarchical structure for SDTS datasets that enables a combination of these data models to coexist and reference any of the objects within the entire dataset. This multi-model dataset design must, of course, permit future decoders to easily determine which data models are present, as well as to decode only those data models that are meaningful to the decoder.

5 SUMMARY

SDTS has an important role, which is complementary to the emerging OpenGIS, in providing a standard format for transfer and archive of very large datasets that is independent of any particular GIS software. However, SDTS has a number of shortcomings. We have presented many issues in this paper, including the need for support of:

explicit and unambiguous definition of basic and compound features, as well as feature relationships;
complete and unambiguous description of a data product's specification within a dataset;
multiple levels of topology, as well as allowing both vector and raster within a single dataset;
extensive metadata support, including harmonization of SDTS with FGDC metadata content standards; and
harmonization of SDTS spatial objects with OpenGIS "well-known structures."

In addition, requirements analysis from the development of DLG/F has added to this list:
subtiling within a dataset;
database-unique, immutable object identifiers;
incremental update and value-adding capability;
lineage tracking through splits, merges and other edits;
dynamic schema, where the schema is part of each data module; and
addition of CORBA IDL-like object behavior signatures expressed in dynamic schema form.

Some of these issues might have been anticipated in the design of SDTS, while others have evolved with technology improvements in the GIS industry. But we have learned much from the current standard and profiles, and it is now time to incorporate this learning into a strategy for the next generation of SDTS. Some considerations for an Object Profile have been presented, which start with harmonizing the frameworks for DLG/F, OpenGIS, and CORBA IDL. The dynamic schema approach in DLG/F provides a promising method for describing any geospatial data model, as well as to enable SDTS to become self-extensible. Even if we succeed in addressing all the above issues, we have still not found the means to express the full semantics of a geospatial data model unambiguously within the dataset, but we must take a step at a time.

This is a proposal of work that is needed. There is much to be done, but our approach is to develop an incremental strategy that would yield enough early and continuing results to be worthwhile. This will require time and coordination with various standards bodies. Please contact the authors if you wish to participate in this endeavor.

6 REFERENCES

Altheide, P., 1992a, An Implementation Strategy for SDTS Encoding. Cartography and Geographic Information Systems 19:5, p. 306-310.

_____, 1992b, Design of a Spatial Data Transfer Processor. Cartography and Geographic Information Systems 19:5, p. 311-314.

_____, 1993, The SDTS Mapping of the DLG-E Model. Documentation on the Internet, ftp://sdts.er.usgs.gov/pub/sdts/datasets/tvp/dlge/DLG_E_Mapping_SDTS.wp.

American National Standards Institute (ANSI), 1986, Specification for a Data Descriptive File for Information Interchange, ANSI/ISO 8211:1985. FIPS PUB 123, New York.

Arctur, D., 1996a, Design of an Extensible, Object-Oriented GIS Framework with Reactive Capability. Doctoral Thesis (Gainesville, FL: University of Florida).

_____, 1996b, Spatial Data Transfer Standard (SDTS): A GIS Vendor's Perspective. Document on the Internet, http://www.lsl.co.uk/~arctur/portfolio/sdts.html.

_____, 1997, Introduction to Object-Oriented GIS. Notes for URISA'97 Conference Workshop (Washington DC: Urban and Regional Information Systems Association).

Arctur, D., Anwar, E., Alexander, J., Chakravarthy, S., Chung, M., Cobb, M., and Shaw, K., 1995, Comparison and Benchmarks for Import of Vector Product Format (VPF) Geographic Data from Object-Oriented and Relational Database Files. In Proceedings of the Fourth International Symposium on Advances in Spatial Databases, SSD'95, Portland, Maine (New York: Springer-Verlag), pp.368-384.

Bancilhon, F., Delobel, C., and Kanellakis, P., 1992, Building an Object-Oriented Database System: The Story of O2 (San Mateo, CA: Morgan Kaufmann).

Booch, G., 1994, Object-Oriented Analysis and Design, with Applications (New York: Benjamin/Cummings).

Coplien, J., and Schmidt, D., eds., 1995, Pattern Languages of Program Design (New York: Addison-Wesley).

Chung, M., Cobb, M., Shaw, K., and Arctur, D., 1995, An Object-Oriented Approach for Handling Spatial Topology in VPF Products. In Proceedings of GIS/LIS'95 Conference, Nashville, Tennessee, p. 163-174.

Cobb, M., Chung, M., Shaw, K., and Arctur, D., 1995, A Self-Adjusting Indexing Structure for Spatial Data. In Proceedings of GIS/LIS'95 Conference, Nashville, Tennessee, p. 182-192.

_____, 1996, Object-Oriented Database Design and Implementation Issues for Object Vector Product Format (OVPF). Technical Report No. NRL/FR/7441—95-9641 (Washington DC: Naval Research Laboratory).

Davis, B., George, J., and Marx, R., 1992, TIGER/SDTS: Standardizing an Innovation. Cartography and Geographic Information Systems 19:5, p. 321-327.

Egenhofer, M., and Frank, A., 1987, Object-Oriented Databases: Database Requirements for GIS. In Proceedings of International Geographic Information Systems (IGIS) Symposium: The Research Agenda, Arlington, Virginia, p. 189-211.

_____, 1992, Object-Oriented Modeling for GIS. Journal of the Urban and Regional Information Systems Association 4:2, p. 3-19.

Federal Geodetic Control Subcommittee (FGCS), 1996, Draft Spatial Data Transfer Standard, Point Profile. Document on the Internet, http://www.ngs.noaa.gov/FGCS/sdtsp.html.

Federal Systems Integration and Management Center (FEDSIM), 1996a, Spatial Data Transfer Standard: Senior Management Overview (Reston, VA: U.S. Geological Survey).

_____, 1996b, Spatial Data Transfer Standard: Guide for Technical Managers. (Reston, VA: U.S. Geological Survey).

_____, 1996c, Spatial Data Transfer Standard: Handbook for Technical Staff. (Reston, VA: U.S. Geological Survey).

Fegeas, R., Cascio, J., and Lazar, R., 1992, An Overview of FIPS 173, The Spatial Data Transfer Standard. Cartography and Geographic Information Systems 19:5, p. 278-293.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. 1995, Design Patterns: Elements of Reusable Object-Oriented Software (New York: Addison-Wesley).

Goldberg, A., and Robson, D., 1989, Smalltalk-80: The Language (New York: Addison-Wesley).

Goldberg, A., and Rubin, K., 1995, Succeeding with Objects: Decision Frameworks for Project Management (New York: Addison-Wesley).

Greenlee, D., 1992, Developing a Raster Profile for the Spatial Data Transfer Standard. Cartography and Geographic Information Systems 19:5, p. 300-302.

Hair, D., Timson, G., and Martin, P., 1997, Feature Maintenance, Version 2.6. Internal Research Report (Reston, VA: U.S. Geological Survey).

Herring, J., 1992, TIGRIS: A Data Model for an Object-Oriented Geographic Information System. Computers and Geosciences 18:4, p. 443-452.

International Hydrographic Organization (IHO), 1996, IHO Transfer Standard for Digital Hydrographic Data, Special Publication No. 57, Edition 3.0. Document ordering information on the Internet, http://www.iho.shom.fr/.

International Standards Organization (ISO), JTC 1 / SC 21, 1994, Information Technology - Specification For A Data Descriptive File For Information Interchange, ISO/IEC 8211:1994. Document ordering information on the Internet, http://www.iso.ch/cate/d20305.html#0.

Jacobson, I., Christerson, M., Jonsson, P., and Overgaard, G., 1992, Object-Oriented Software Engineering (New York: Addison-Wesley).

Kelley, C., and Gosinski, T., 1994, Spatial Data Transfer Standard: Do You Fit The Profile? GIS World, August 1994, p. 48-50.

Laser-Scan, Ltd. (LSL), 1996, Introduction to Gothic Concepts. Technical Report (Sterling, VA, and Cambridge, UK).

Laurini, R., and Thompson, D., 1992, Fundamentals of Spatial Information Systems (New York: Academic Press).

Lazar, R., 1992a, The SDTS Topological Vector Standard. Cartography and Geographic Information Systems 19:5, p. 296-299.

_____, 1992b, SDTS Support Software: The FIPS 123 Function Library. Cartography and Geographic Information Systems 19:5, p. 303-305.

_____, 1996, Understanding SDTS Topological Vector Profile Implementations. GeoInfo Systems, June 1996, p. 42-45.

National Institute of Standards and Technology (NIST), 1994, Federal Information Processing Standard Publication 173-1, Spatial Data Transfer Standard (Washington DC: U.S. Department of Commerce).

National Mapping and Imagery Agency (NIMA), 1996, Interface Standard for Vector Product Format, MIL-STD-2407 (Washington DC: U.S. Department of Defense).

Object Management Group (OMG), 1996, The Common Object Request Broker: Architecture and Specification, Version 2.0. Technical Document PTC/96-08-04. Document on the Internet, http://www.omg.org.

Open GIS Consortium (OGC), 1997, Open GIS Abstract Model. Document on the Internet, http://www.opengis.org/public/abstract.html.

Raster Technical Review Board (RTRB), 1997, Spatial Data Transfer Standard, Part 5: Raster Profile with Basic Image Interchange Format Extension. Draft Document (Reston VA: Federal Geographic Data Committee).

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorenson, W., 1991, Object-Oriented Modeling and Design (Englewood Cliffs, NJ: Prentice-Hall).

Smith, D., 1991, Concepts of Object-Oriented Programming (New York: McGraw-Hill).

Smith, T., Peuquet, D., Menon, S., and Agarwal, P., 1987, KBGIS-II: A Knowledge-Based Geographical Information System. International Journal of Geographical Information Systems 1:2, p. 149-172.

Szemraj, J., 1992, TIGER/SDTS Topology. Cartography and Geographic Information Systems 19:5, p. 328-331.

Tang, A., Adams, T., and Usery, E., 1996, A Spatial Data Model Design for Feature-Based Geographical Information Systems. International Journal of Geographical Information Systems 10:5, p. 643-659.

Timson, G., 1996, Introduction to Feature Communication Protocol, Version 1.0. Internal research report (Reston, VA: U.S. Geological Survey).

U.S. Geological Survey (USGS), 1997, SDTS Home Page. Document on the Internet, http://mcmcweb.er.usgs.gov/sdts.

Volpe National Transportation Systems Center (VNTSC), 1996. Draft Spatial Data Transfer Standard, Transportation Network Profile, Version 1.1. Document on the Internet, http://www.bts.gov/gis/reference/tnp_11.html.

Wiegand, N., and Adams, T., 1994, Using Object-Oriented Database Management for Feature-Based Geographic Information Systems. Journal of the Urban and Regional Information Systems Association 6:1, p. 21-36.

Wirfs-Brock, R., Wilkerson, B., and Wierner, L., 1990, Designing Object-Oriented Software. (Englewood Cliffs, NJ: Prentice-Hall).

Worboys, M., Hearnshow, H., and Maguire, D., 1990, Object-Oriented Data Modelling for Spatial Databases. International Journal of Geographical Information Systems 4:4, p. 369-383.

Worboys, M., 1994, Object-Oriented Approaches to Geo-Referenced Information. International Journal of Geographical Information Systems 8:4, p. 385-399.

Wortman, K., 1992, The Spatial Data Transfer Standard (FIPS 173): A Management Perspective. Cartography and Geographic Information Systems 19:5, p. 294-295.