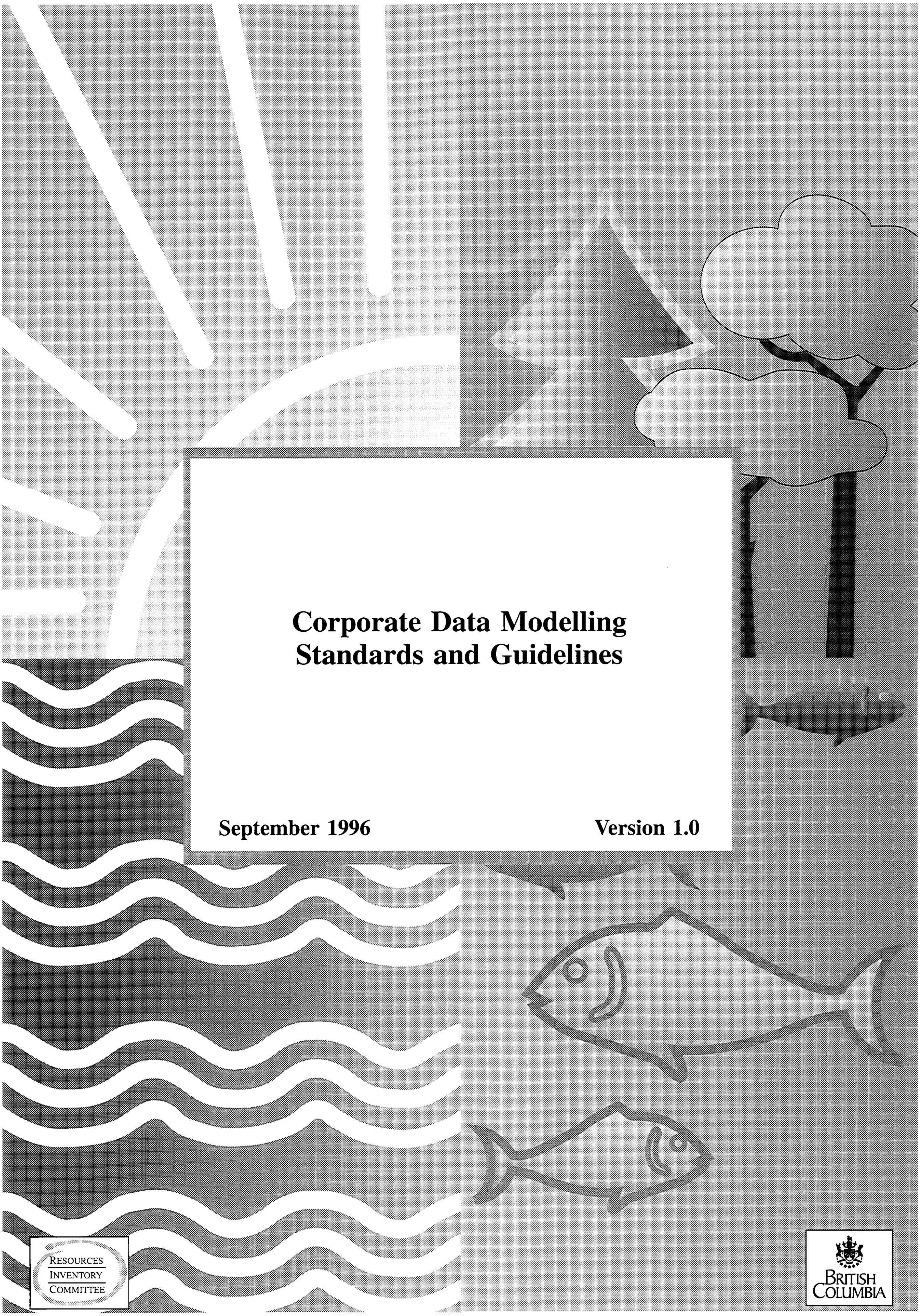

Corporate Data Modelling Standards and Guidelines

September 1996

Version 1.0

RESOURCES
INVENTORY
COMMITTEE

BRITISH
COLUMBIA

BRITISH COLUMBIA

RESOURCES INVENTORY COMMITTEE

Resources Inventory Committee

STATEMENT LIMITING THE LIABILITY OF THE PROVINCE

The entire risk of loss, personal injury, or damages of any kind, as a result of using the information and procedures contained in these materials, including the print and/or workshop materials, is with the user. Although the Province of British Columbia has used considerable efforts in preparing the materials for use in the province of British Columbia, the Province of British Columbia does not warrant the accuracy, completeness or effectiveness of the information and procedures contained in the materials. In no event will the Province of British Columbia be liable for damages of any kind arising out of the use of the materials.

Government Publications Centre
Phone: (250) 387-3309 or
Toll-free: 1-800-663-6105
Fax: (250) 387-0388
www.publications.gov.bc.ca

Corporate Data Modelling Standards and Guidelines

Prepared by
Ministry of Environment, Lands and Parks
LandData BC, Geographic Data BC
for the Land Use Task Force
Resources Inventory Committee

September 1996

Version 1.0

Published by the
Resources Inventory Committee

Canadian Cataloguing in Publication Data

Main entry under title:
Corporate data modelling standards and guidelines [computer file]

Available through the Internet.
Issued also in printed format on demand.
Includes bibliographical references: p.
ISBN 0-7726-3529-3

1. Land use - British Columbia - Databases - Mathematical models. 2. Information storage and retrieval systems - Land use - British Columbia - Mathematical models. I. LandData BC.
II. Resources Inventory Committee (Canada). Land Use Task Force.

HD319.B7C672 1998 333.73'13'011309711 C98-960100-5

Additional Copies of this publication can be purchased from:

Government Publications Centre
Phone: (250) 387-3309 or
Toll-free: 1-800-663-6105
Fax: (250) 387-0388
www.publications.gov.bc.ca

Digital Copies are available on the Internet at:

http://www.for.gov.bc.ca/ric

Acknowledgments

Funding of the Resources Inventory Committee work, including the preparation of this document, is provided by the Corporate Resource Inventory Initiative (CRII) and by Forest Renewal BC (FRBC). Preliminary work of the Resources Inventory Committee was funded by the Canada-British Columbia Partnership Agreement of Forest Resource Development FRDA II.

The Resources Inventory Committee consists of representatives from various ministries and agencies of the Canadian and the British Columbia governments as well as from First Nations peoples. RIC objectives are to develop a common set of standards and procedures for the provincial resources inventories, as recommended by the Forest Resources Commission in its report “The Future of our Forests”.

Land Use Task Force

This document is based in part on an unpublished draft prepared for the Resources Inventory Committee (RIC) by Glenn Pope of MacDonald, Dettwiler and Associates Ltd. (MDA), with assistance from Ozzie Sawicki of MDA and Richard Lay of Context Consulting. Guidance and support in the development of this document has been provided by the RIC Data Model Committee (DMC), under the chairmanship of Evert Kenk. The document also benefited from assistance and spatiotemporal information provided by Mark Sondheim and Peter Friesen of the Strategic Developments Unit.

Table of Contents

List of Figures

1. Introduction

The purpose of this document is to provide guidance on modelling techniques and standards specific to LandData BC corporate data modelling. There are several BC Environment Data Administration Practices which will be referenced, so this document can be thought of as an extension of these practices as they relate to the CDM. In addition, some modelling techniques necessary in corporate data modelling (e.g., spatiotemporal modelling), and some tools pointers, are covered.

There are a number of good textbooks on the subject of data modelling, which will cover most of the data modelling techniques required to do corporate data modelling. For Oracle Designer/2000 users, a good text to reference is ORACLE *CASE*Method Entity Relationship Modelling* [R-3] by Richard Barker, published by Addison Wesley. A useful text for LogicWorks ERwin users is the *Methods Guide* [R-4], which is part of the documentation set delivered with the tool. Usage of the LandData BC Repository Management Tool is documented in a User Manual, which covers all aspects of data entry and reporting.

1.1 Audience

These guidelines are intended for information engineering professionals, such as:

- Corporate Data Modellers,
- Data Administrators, and
- Database Developers.

1.2 Prerequisites

The material covered assumes a working knowledge of conceptual and logical data modelling using ER (Entity Relationship) diagramming.

Readers should be familiar with the *LandData BC Corporate Data Model Framework* (CDMF) document [R-1]. It describes the approaches used for doing corporate data modelling, as well as the repository structures and tools utilized by Corporate Data Modellers. There are multiple references to document [R-1] in the sections which follow.

1.3 Document Outline

The document contains three major sections:

- Introduction: Provides an introduction to this document, including document prerequisites and CDM Framework standards conformance.
- Corporate Data Modelling Procedures: Contains a detailed description of the procedures used for producing the CDM (Corporate Data Model), as well as ensuring quality of the resulting models.
- Data Modelling Guidelines: Contains a detailed description of approaches and standards used in ER diagramming and spatiotemporal modelling.

1.4 CDM Standards Conformance

The *Corporate Data Model Framework* document [R-1] provides a framework for the CDM and Corporate Data Modelling. Conformance to CDM standards requires conforming to the architectures, guidelines and standards outlined in both [R-1] and the *Corporate Data Modelling Standards and Guidelines* (this document).

1.5 Document Status

This manual is in the developmental stage for the first year. Most sections in this document have not been tested. A review will be held a year from now to revise the status as appropriate with the then stage of development and usage of this document.

2. Corporate Data Modelling Procedures

2.1 Model Transformation Process

An overview of the process is provided in [R-1] *CDM Vision*. Figure 1.2 [R-1], *Corporate Data Modelling Process* provides a view of the different types of data models (i.e., source, conforming, sub-federated, and federated) and the key activities which result in transforming one type to another. Basically, the process involves capture of source meta-data and a consolidation, through transformations on the meta-data, into a Federated Corporate Data Model. The model transformation process is now described in more detail.

2.1.1 Source Model Capture

Source models may be a data model, a schema, or a model based on a standard, or a combination of these. They are developed during local modelling, i.e., the modelling is performed during local database development at the ministry level. In order to facilitate data sharing (described more fully in [R-1] *Data Sharing*) and development of the CDM, it would be cost effective to utilize local models as input to the CDM. In order to make the process of source model capture as effective as possible, it is desirable that the source models follow the same rules that govern the CDM. In turn, this will facilitate the utilization of information in the CDM during the development of other local models. It will also simplify the process of transforming the source models into conforming or federated models. The high level information flows involved are as follows:

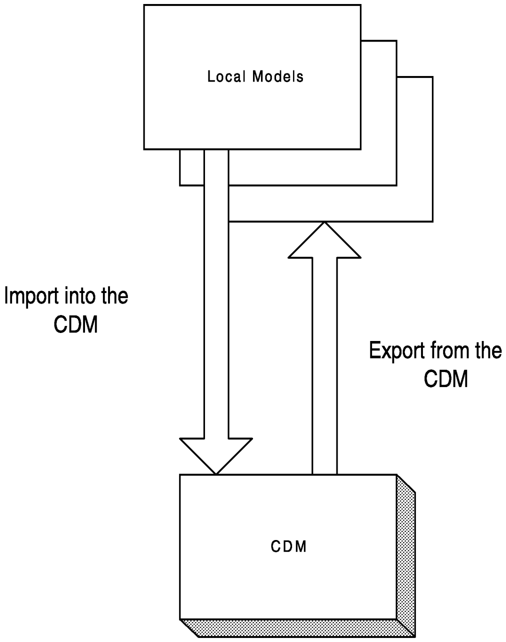

Figure 2.1 CDM Import/Export Data Flows

In order to facilitate source model capture, it is necessary that they be in the generic model format described in [R-1] section *Generic Model Structure*. A minimum requirement is that a schema be provided in an ERD (Entity Relationship Diagram) format accompanied by descriptive information, either in ERwin or Designer/2000.

The source information is imported into the CDM repository; at this point, the resulting model is considered non-conforming, unless the source it came from is considered to be conforming to CDM standards. The process of dumping source data from CASE (Computer-Aided Systems Engineering) tools into files, and then loading those into the LandData BC Repository is called 'Extract' in [R-1] Figure 1.2. Figure 2.2 is an alternate view of the process, describing the importation tasks and showing who is responsible:

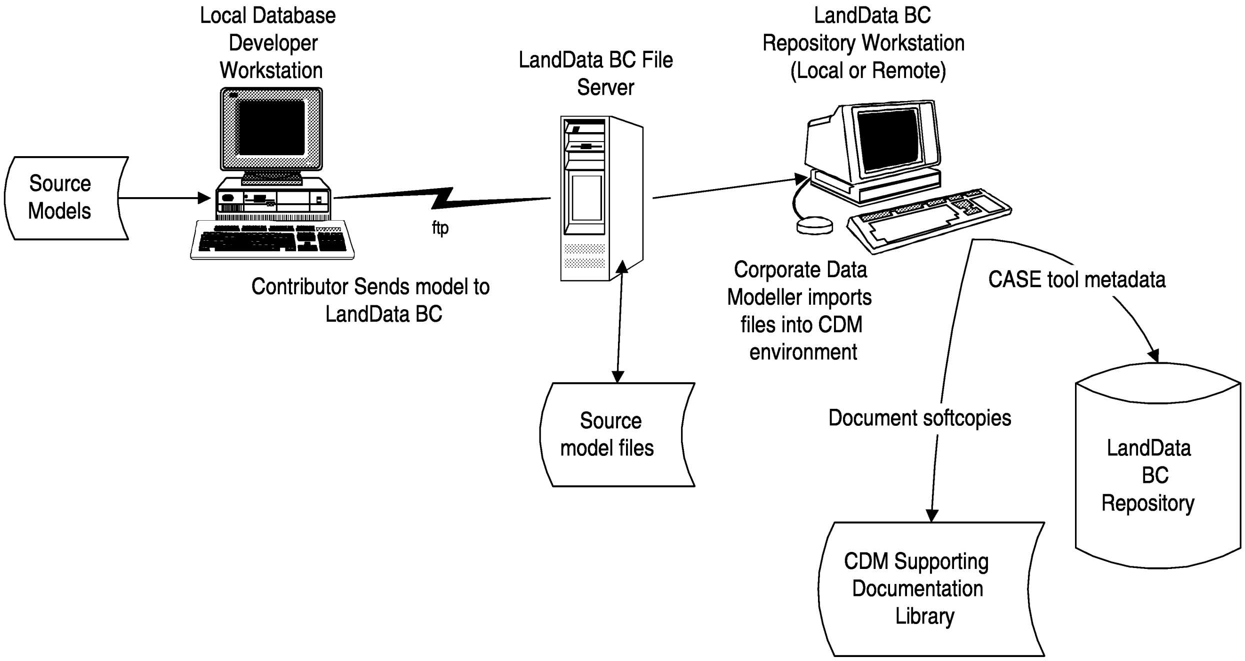
Local Database Developer Workstation
LandData BC File Server
LandData BC Repository Workstation (Local or Remote)
Source Models
ftp
Contributor Sends model to LandData BC
Corporate Data Modeller imports files into CDM environment
CASE tool metadata
Source model files
Document softcopies
LandData BC Repository
CDM Supporting Documentation Library

Figure 2.2 Source Model Importation

If a standard, such as a RIC Inventory Standard, does not have a schema supplied with it, one must be developed before importation can proceed. Typically, this will involve implementation or extension of a database for the standard by a Database Developer, which will then make the schema available. This process is called 'Forward Engineer' in the [R-1] Figure 1.2.

The importation includes documentation, which must be provided in softcopy. The standard documentation tools used in the CDM are Microsoft Word, the Microsoft Excel spreadsheet and the Visio drawing tool. [R-1] Section 3, *Corporate Data Modelling Tools* provides further information on these tools. Any softcopy provided in other formats must be converted to formats supported by these tools prior to capture, in order to provide a common tools access interface to Corporate Data Modellers.

2.1.2 Model Standardization

The next step is to transform the source data models into standardized data models and spatiotemporal frameworks, and to compare them to other models at this stage to arrive at a comparative level of descriptive detail.

The data model transformations are central to the consolidation into a federated CDM, and supporting documentation must be modified by the Corporate Data Modeller to stay consistent with the transformed data models. As part of this activity, traceability records of what components are being transformed into what new components, are created and filed.

This step involves dealing with detailed meta-data. Depending on the background of the Corporate Data Modeller, he or she may not have sufficient information to deal with all aspects of the models. They may therefore call for assistance from local Data Administrators, Database Developers, and domain experts.

The following detailed processes are required:

- Any physical schemata provided are reverse engineered into an LDM (Logical Data Model) using ERwin, or, if more effective, Designer/2000.
- Conceptual data models are forward engineered to an LDM provided this is feasible, i.e.,
  - there is sufficient meta-data available to do so, or
  - there is a need to forward engineer to an LDM because this information is required by Government users and there are domain experts who can assist with the transformation.
- The resulting LDM components are analyzed for syntactic conformance to other models at this level of the federated CDM. Differences are analyzed and resolved.
- Next, the LDM components are analyzed for semantic conformity, i.e., how do the following compare?
  - relationships, including all constraints such as cardinality and mandatory/optional constraints,
  - attributes, including structure and naming,
  - data types, and
  - domains, including extensions (instances and enumerations) of data for specific attributes,
  - descriptions (Are there attributes with the same or similar names, but different descriptions? Are there attributes with the same name but different descriptions?)
  - business rules (Business rules are constraints on data over and above entity relationship constraints.)
- Differences are analyzed and resolved, resulting in changes to the source models or possibly to the CDM. Changes include:
  - revising entity and attribute names;
  - amalgamation/generalization or partitioning of entities;
  - creation, elimination or re-alignment of entity subtypes;
  - changes in attribute definitions, i.e., partitioning into more elementary attributes, merging, or other more complex attribute transformations; and
  - transformation of datatypes to standard logical datatypes.

Any resulting LDMs must at a minimum be in Third Normal Form (3NF).

Changes to the CDM should be additive only, since retrofitting modifications (e.g., changing an object name or a relationship) may no longer leave previously integrated models in a conformant state. The resultant data model must conform to the standards in BC Environment Data Administration Practice #2, *NAMING CONVENTIONS* attached in Appendix A.

If a decision is made to leave a model at the conceptual level, then all that is required is to ensure that the entity names and relationships conform to the corresponding CDM entities and relationships (if those exist). The resultant data model must conform to the standards in BC Environment Data Administration Practice #2, *NAMING CONVENTIONS* attached in Appendix A.

- Standardize spatiotemporal data based on the approach outlined in the Corporate Data Model Framework [R-1] in section *Spatiotemporal Modelling*, with emphasis on the allowable SAIFLite components described in section *SAIFLite Components*, and the modelling approach outlined in section *Corporate Data Modelling Using SAIFLite*.
- Add data distribution information. This involves adding any new location information (i.e., information not already in the CDM) to the meta-data at the attribute level.
- N.B.: a given source model (schema) may span more than one subject category, and may therefore be modelled in more than one Designer/2000 Application System. If this is the case, care must be taken to ensure that no redundancies are introduced or at least are minimized.

At this point, the models are considered to be conforming to CDM standards. Refer to Section 1.4, *CDM Standards Conformance* for a definition of standards conformance. Figure 2.3 on the next page is a graphic depiction of high level model standardization data flows:

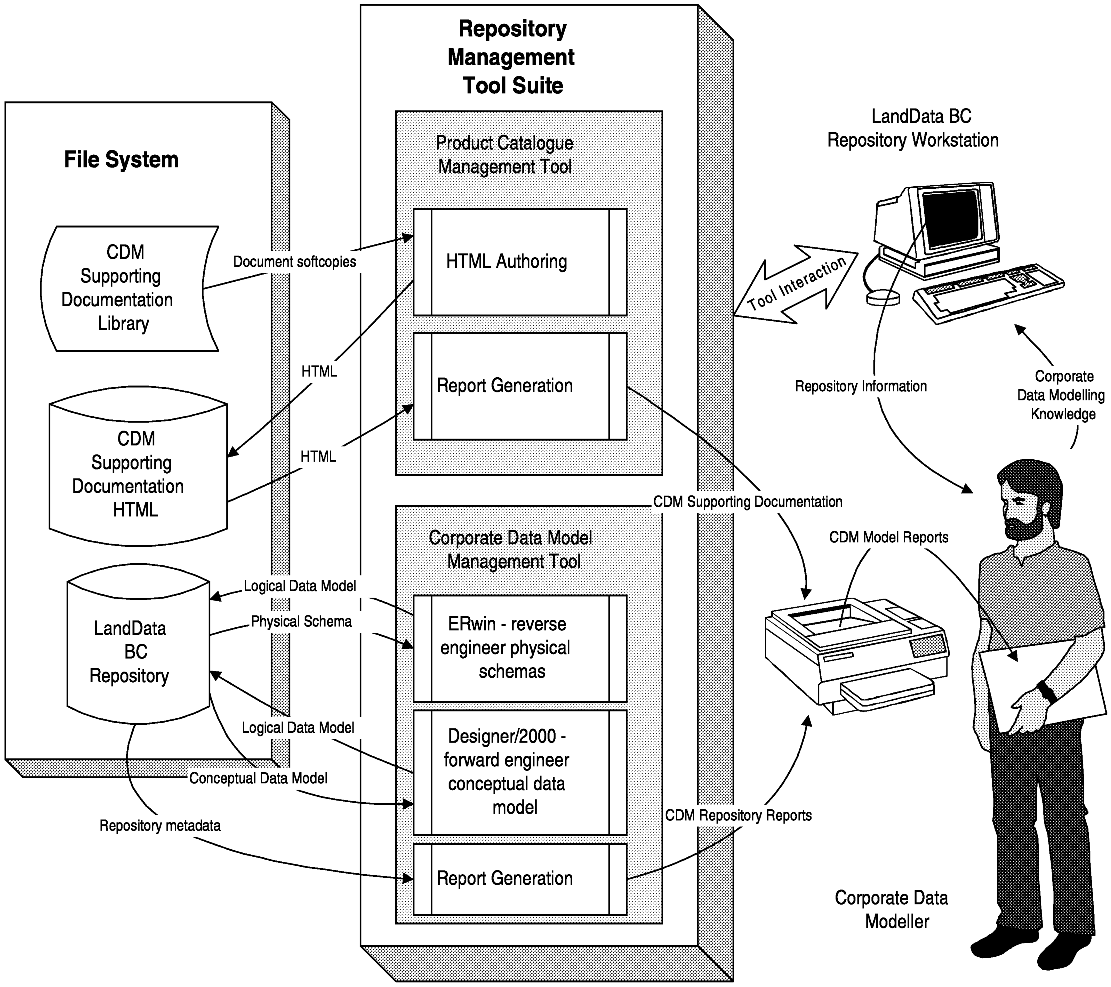

Figure 2.3 Model Standardization Data Flows

2.1.3 Creating a Sub-Federated Model

A sub-federated model is a consolidation of conforming models typically at a Subject-Area level ([R-1] in section *Classification Hierarchy* discusses subject categories ‘Supersubject-Area’, ‘Subject-Area’ and ‘Datagroup’). Typically, a Corporate Data Modeller will be dealing with one or more models classified at a ‘Subject-Area’ or ‘Datagroup’ level. The models are compared for shared and distinguishing features, either amongst themselves or to an existing sub-federated model which is a consolidation of models in the appropriate ‘Subject-Area’, i.e., the one that includes the ‘Datagroup’s being consolidated. The process is a difference analysis which is carried out mechanically by performing SQL operations on the meta-data. The differences, or distinguishing features (entities, relationships, subtypes, attributes, etc.), are analyzed and added to (integrated with) the sub-federated model. The process shown in [R-1] Figure 1.2 is ‘Integrate Models for a Category’.

2.1.4 Federated Model Integration

The final step in the consolidation process is the integration of the sub-federated model with the federated model. The process is the same as the sub-federated model creation/integration step. That is, the sub-federated model is compared to the federated model, and selected distinguishing features or differences are added to the federated model, i.e., the models are integrated with the proviso that the federated model remains a roadmap to the sub-federations. That is, context dependent data details, either attributes, subtypes, or entity relationships, are only shown at the sub-federation levels. The federated model need not be fully attributed. As a rule, attributes (in particular keys) that can be kept in the federated model should be ones that are common to sub-federations, i.e., entities in the federated model are like supertypes. Those same attributes should not be stored at the sub-federation level. Therefore the total attribute set for an entity in a given context is the union of sub-federation attributes and federated model attributes for that entity.

2.1.5 CDM Quality Assurance

Quality assurance is implemented through a strategy of model confirmation, which is a process of:

- model review by source contributors: the models at various stages of transformation are validated with model contributors using LandData BC product access facilities; that is, the models are made available to clients, so they are aware of what is going on and can provide input; and
- acceptance and signoff by the LandData BC Data Administrator.

The confirmation and acceptance of a model depend on the level of model being confirmed. Confirmation occurs after:

- the transformation from a conforming to a sub-federated model, and
- the integration of the sub-federated model with the federated model.

The models must be conformant with CDM standards as described in Section 1.4, *CDM Standards Conformance*.

Sub-Federated Model Signoff

There are two main steps, both requiring signoff by the respective parties shown below. The Corporate Data Modeller is responsible for ensuring that these reviews are carried out and the signoffs are obtained, before proceeding with the federated model integration. The data model components, which typically consist of ERDs and LDMs, are to be reviewed and signed off in accordance with the BC Environment Data Administration Practice #1, *LOGICAL DATA MODELS*, attached in Appendix A. In accordance with this Practice, this may require other parties to signoff (in addition to the ones shown below).

Review by Contributor–

The main purpose of this review is to obtain agreement from the contributor (i.e., the Data Provider) that the transformed model is representative of the source model(s) provided. Two key aspects to be reviewed are:

- proper interpretation of the source model(s), and
- fidelity of any resulting schema(s).

Review by LandData BC Data Administrator–

The LandData BC Data Administrator will perform a quality assurance evaluation of the sub-federated model. This will ensure that the model meets the standards and guidelines in this document, the CDM Framework. In particular, key aspects to be reviewed are:

- conformance to CDM standards, e.g., syntactical rules, and

- semantic conformance, i.e., the “meaning” or “usage” of names, objects and relationships is correct and as it was intended by their authors/contributors.

Federated Model Signoff

The LandData BC Data Administrator will ensure that the integration is syntactically and semantically conformant. A final review with the contributor will be held to ensure that the models have been properly integrated.

2.2 Repository Reporting

The model confirmation strategy outlined above requires reviews by contributors of their meta-data in the CDM. Reports which provide the required views of the data are available. Reports by ‘Datagroup’ or ‘Subject-Area’ are also available. Where ERDs are required, the models at the conforming, sub-federated or federated levels are available. All of these, as well as supporting documentation, are available either from:

- LandData BC as products, or
- the LandData BC Schema & CDM Management Tool Reporting function.

2.3 Model Documentation Standard

Supporting documentation (see [R-1] section *Generic Model Structure* for a definition of supporting documentation) is organized using the following template:

Title, e.g. “CDM - Vegetation Subject Area Analysis”

List of Acronyms and Abbreviations

1. Introduction
   - Goal of document
   - Brief summary of discipline process, including pictures
   - Tools used
2. Scope
   - Source models, databases and standards used for input and brief description & initial assessment of each
   - Stakeholders
   - Models & documents not used
3. Process
   - Overall approach
   - Exceptions to the Corporate Data Model Framework
   - Issues
4. Models
   - Models used for input
   - Models produced

   Each of the above models will contain the following:
   - paragraph on scope
   - quality, e.g., uncertainty about entity or attribute names
   - model sources and contributors (Each of the contributors of the source models should be documented. In the future any CDM entity should have an identified custodian/Data Manager who is the key person to consult when changes to the model are anticipated for that entity.)

    - model specific issues (Any model will have a number of issues that have arisen during their development which should be documented. This section will include these issues.)
    - relationship of model to discipline process
    - discussion of transformation and interpretation issues (A summary of transformations that have been applied to source and intermediate models to establish the entities in the CDM should be included so that a reader, e.g., a model contributor, can trace source entities and determine whether they have been fairly represented.)
5. Integration issues and resolutions
    - rationale on significant integration issues
    - terminology issues
    - datatype integration
    - attribute mapping
    - entity resolution
6. Status and current issues outstanding
7. Next steps if model is incomplete
8. Reference documentation and standards
9. Authorized waivers for any deviations from CDM standards. See Section 1.4, *CDM Standards Conformance* for a definition of standards conformance.

Appendices

A. Glossary of terms used in the document

B. Cross-reference of Standards and Models to Entity and Database to Entity

C. CDM - ERD

D. CDM Definitions and CASE Tool Reports

2.4 Conformance to Data Modelling Standards

All conforming models shall meet the standards described in the BC Environment Data Administration Practices Standards:

#1 LOGICAL DATA MODELS

#2 NAMING CONVENTIONS

These standards are attached in Appendix A.

Any deviations from standard:

- must be approved by the LandData BC Data Administrator and
- an authorized waiver must be included in the supporting documentation.

3. Data Modelling Guidelines

In addition to information in the BC Environment Data Administration Practices referenced above, the data modelling textbooks listed in Section 1 *Introduction* suggest numerous good data modelling practices which should be utilized by Corporate Data Modellers. In particular, the choosing of object names, labelling relationships, and positioning of entities during Entity Relationship (ER) modelling are well documented. The following are added information recommended for use by Corporate Data Modellers.

3.1 ER Modelling

This sections provided additional ER modelling standards and guidelines.

3.1.1 Dividing and Combining Entities

Entities that have more than one relationship should be divided in two when they contain no entity instances which will commonly be traversed through from one relation to another. The resulting entities will not have a relationship between them.

3.1.2 Entity Subtyping

The data models for land-related data are semantically rich and require a significant amount of entity subtyping. When instances of an entity can be classified such that certain groups of attributes have values for only some classifications, and are null for others, then the entity is a candidate for subtyping. The classifications become subtypes, and they possess the attributes which have values in them for instances of each subtype. Attributes that are never null for any instance of the entity are assigned to a supertype.

Entity Subtyping Rules

Subtyping must be based on the following rules:

- the criterion for creating subtypes for an entity is based on attributes, which are distinct for every subtype; in addition, no attributes found in subtypes are allowed in the supertype (i.e., what remains of the original entity after subtyping);
- every subtype must possess at least one non-key attribute;
- the modelling of instances shall be avoided;
- all attributes from the supertype are inherited, i.e., the combinations of attribute values for instances of a supertype and its associated subtype are valid and meaningful;
- relationships between instances of two or more subtypes at the same level may be exclusive (only one allowed) or inclusive (one or more allowed); the model shall show which one it is;
- subtypes are not allowed to be orphaned, i.e., every subtype instance must have an associated supertype instance (mandatory relationship); and
- all relationships between a supertype and its subtypes are one-to-one, i.e., a given instance of a super type may be associated with at most one instance of each of its subtypes.

Multiple Inheritance

In multiple inheritance, a subtype has two or more supertypes. For example, in the SAIF (Spatial Archive and Interchange Format) model shown in Figure 3.4. the entity 'PartitionedCoverage' has supertypes 'SpatialDataSet' and 'Coverage'. The Object Modelling Technique (OMT) modelling paradigm used in this diagram can support multiple inheritance, however Designer/2000 cannot because it supports only single inheritance. The approach used for implementing multiple inheritance in Designer/2000 is to apply one of several possible workarounds. These are documented in [R-2]. The workaround usually chosen involves picking one supertype for the 'is-kind-of' (i.e., regular super/subtype) relationship and making the other supertype(s) associated components of the subtype, i.e., there is a simple one-to-one associative relationship between them. By convention, the relationship on the subtype side is called 'is-kind-of' to indicate that it is a subtype relationship. A refinement to this approach is to extend the subtype to include the attributes from the associated "supertype" and giving it a composite name. This forces inheritance of attributes. None of these solutions are perfect, and any additional programming code required for emulating subtype behaviour in a physical implementation of the relationship usually has to be manually maintained.

3.1.3 Aggregation

This term is used in an Object Oriented (OO) modelling context. Aggregation is a strong association between two or more entities, and is found in situations involving 'is-part-of' or 'contains' relationships. Thus a 'car door' 'is-part-of' a 'car', and a 'door handle' 'is-part-of' a 'car door'. The position of the door handle is determined by the position of the door, and the position of the door is determined by the position of the car., i.e., they are tightly coupled. The OMT approach represents aggregation using a 'diamond' symbol. Designer/2000 does not have this symbol so the approach for modelling aggregation in Designer/2000 is to use a workaround. This involves using standard relationships with standard relationship labels reflecting aggregation, e.g., if we have an 'assembly' and 'parts', then the 'assembly' 'is-composed-of' / 'is-aggregate-of' / 'contains' 'part's; a 'part' in turn 'is-part-of' / 'is-member-of' / 'is-contained-in' 'assembly'. Any programming code required for ensuring 'aggregate' behaviour in a physical implementation has to be manually maintained.

3.1.4 Classifying Entities

Many entities in resource inventory modelling exist to classify, often from a scientific viewpoint, the instances of another entity. Typically the classification consists of a hierarchy. Where the levels of the hierarchy are named and consistent, and there will be relationships (from another entity) to instances of the hierarchy at that level, each level should have a separate entity. Otherwise, the hierarchy should be modelled within one entity in a recursive fashion. An example of this is the modelling of the SURVEY_ENTITY in the Consolidated Wildlife Inventory Model[1]. For purposes of illustrating this approach, the now classical example of a model for an 'organization' will be used. The two example ERDs shown below demonstrate two modelling approaches (simplified for purposes of this illustration), with the recursive model being the preferred approach:

[1] *Consolidated Wildlife Inventory Data Model Project*, Habitat Inventory Section, Wildlife Branch, Ministry of Environment, Lands and Parks (MELP), September 1995 [R-7]

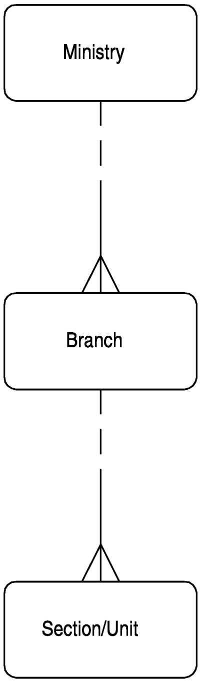

Figure 3.1 Levelled Organization Hierarchy

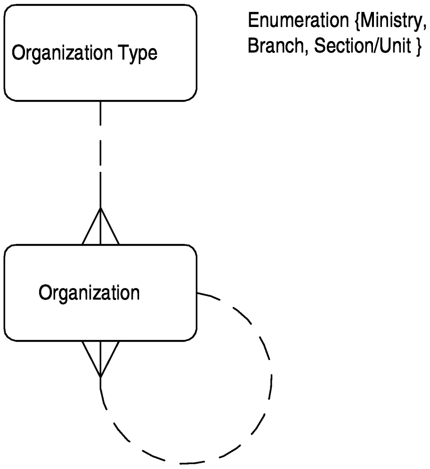

Figure 3.2 Recursive Organization Hierarchy

3.1.5 Single Attribute Entities

Single attribute entities should be subsumed into the entity they relate to as an attribute unless there are a number of relationships to the single attribute entity.

3.1.6 Enumerated Domains

Enumerated Domains are attributes where there are fixed, and named, instance values of an entity. Enumerated domains often form part of a classification system. Sometimes different disciplines will view the same entity from a slightly different viewpoint and will have a classifying enumerated domain with the same name and very similar definition, but different enumerated values. These entities cannot be integrated because it is not possible to query across the two meanings. The only way to integrate these entities is to have the authors of the enumerated values rationalize the values so that queries will be meaningful. This situation should be captured in the model as a Model Issue. The integrated domains are stored in a domain library in the repository.

3.1.7 Data Handling Artifacts

Models may contain entities that are data handling artifacts; that is to say, entities created by information handlers to track their information handling business. Examples include the entity 'Field_Form' or 'Resultant'. While instances of these entities do not occur "naturally" in the environment and are not part of the discipline domain modelled *per se*, the sharing and interchange of land-related data requires that these be part of the models. These entities should be included in the model.

3.1.8 Nominal Entities

Some disciplines will refer to a class of objects with a term which would be an entity if the modelling were done at a single discipline level. These objects may be modelled with another name within the CDM, or may be modelled in such a way that the original objects can be captured stored and delivered within the CDM structure without having a specific entity of that name. When this occurs, there must be a cross-reference and an explanation so that those searching for the Nominal Entity can recognize that it will be properly represented in the CDM.

3.2 Spatiotemporal Modelling

Spatiotemporal modelling in the CDM involves the creation of LDMs which model data normally found in GIS (Geographic Information System) databases.

3.2.1 Spatiotemporal Entities

Spatiotemporal entities have a spatial or locational nature, as well as a temporal component or context. That is to say, they will have attributes that express their location and/or time, or will have spatial relationships with other entities. Part, or all, of the data for these entities will typically be stored and manipulated in a GIS. They are modelled in Oracle Designer/2000 in the same way as regular (i.e., attribute data) entities, with some adaptations for handling the slightly more complex nature of the meta-data involved.

3.2.2 Spatiotemporal Modelling Approach

The GIS schemas housing spatiotemporal data are fairly complex, and it is of benefit to utilize existing standards and templates to facilitate modelling this kind of data. Such an approach is offered by SAIFLite, which is well suited to corporate data modelling. SAIFLite is a functional standard implementation of the SAIF base standard. It provides strong guidelines as to just how SAIF may be employed[2]. It restricts SAIF classes and how those classes may be used for modelling.

The primary focus in the CDM is on modelling geographic objects, which are principally features, and leaving out the detail of physical shape construction. SAIFLite provides the necessary and sufficient standards infrastructure for this approach. Shape information is modelled using a restricted set of geometric object classes, and time information is modelled using a restricted set of time object classes. By extracting feature (i.e., business) information from GIS databases, which is feasible, and leaving the physical aspects of GIS schemas outside the CDM, it is possible to develop LDMs for spatiotemporal data. An example of such a model is provided in Section 3.2.7, *Consolidated Model*.

3.2.3 SAIF Classes

Figure 3.4 *SAIF Standard & User Defined Schemas*[3] shows a high level view of classes in SAIF. These classes are also of interest from a CDM perspective. The diagram is an OMT[4] representation of key high level classes in SAIF.

At this level of diagramming,

- "classes" (the rectangular boxes) are synonymous with "entities",
- the triangular constructs mean "subclass" or "entity subtype", and
- a line with a diamond shape at one end expresses a 'contains' relationship.

The following is a more precise expanation of the notation used, taken from [R-2]:

[2] Adapted from *Introduction, SAIFLite,* Release 1.1, March 1996 [R-6]

[3] Based on *Spatial Archive and Interchange Format: Formal Definiton,* Release 3.2, January 1995, Surveys and Resource Mapping Branch, B.C. Ministry of Environment, Lands and Parks [R-5].

[4] Rumbaugh et al, 1991 [R-2]

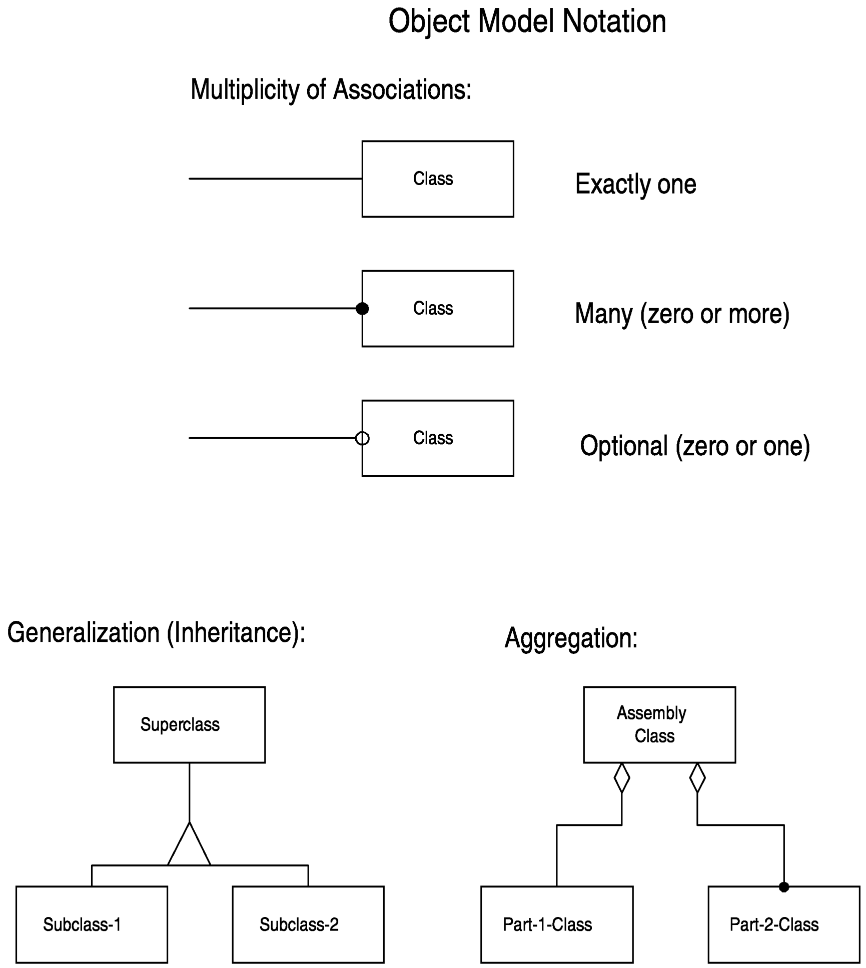

Figure 3.3 OMT Object Model Notation

The three most important classes are: ‘GeographicObject’[5], ‘SpatialObject’ and ‘TemporalObject’. There are other high level classes in SAIF, but these are not shown because they are not allowed by SAIFLite and are not used in the CDM. The area above the broken line shows the standard classes provided by SAIF. These are modelled as entities in the CDM (except for ‘AbstractClass’). The area below the broken line shows user defined classes representing the specifics of models extracted from GIS databases.

[5] Capitalization of class and attribute names follows OO conventions where SAIF and SAIFLite components are referenced.

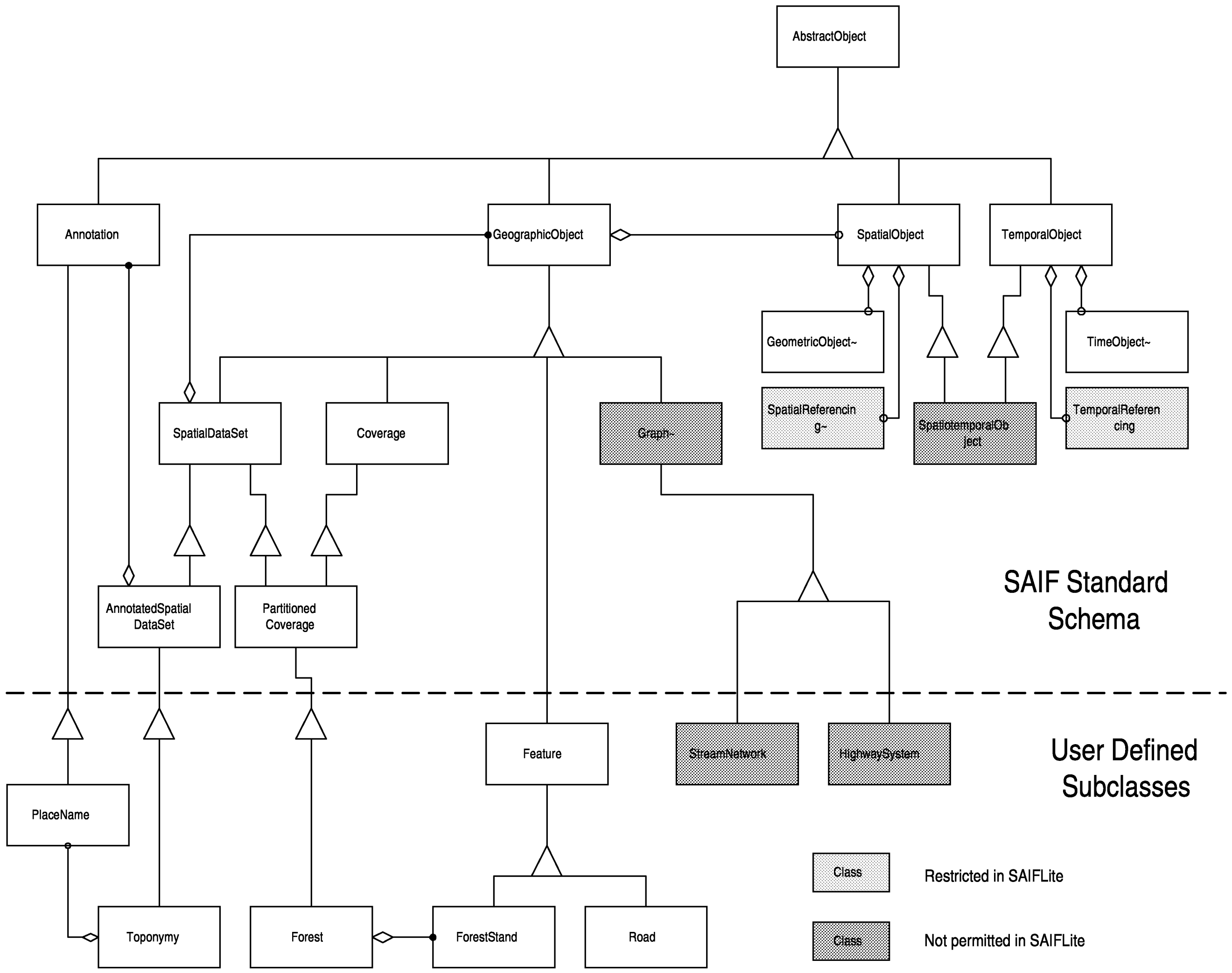

Figure 3.4 SAIF Standard and User Defined Schemas

3.2.4 SAIFLite Components

All of the classes except for shaded ones shown in Figure 3.4 are allowed in SAIFLite. The shaded classes are either restricted under SAIFLite (e.g., 'SpatialReferencing' and 'TemporalReferencing'), or are not permitted (e.g., 'Graph' and any user defined subclasses of 'Graph', and 'SpatiotemporalObject'). SAIFLite is an implementation of a SAIF functional standard or template of allowable classes, relationships, and enumerations.

A key element of SAIFLite is the specification of the 'GeometricObject' class. This class has a number of subclasses (not shown in Figure 3.4) that express geometric information. The allowable subclasses under SAIFLite are:

- GeometricAggregate: contains a set of 'GeometricObject's restricted to 'Point', 'AlignedPoint', 'Arc', 'ArcDirected', 'OrientedArc', and 'BoundedArea'; it is modelled as an attribute, i.e., 'GeometricAggregate'[6], with a set of allowable values, e.g., {'Arc', 'ArcDirected'};
- Point: used for the position of single points, including nodes, origins of grids, placement of text, etc.; it is modelled as an attribute value, i.e., 'Point';
- AlignedPoint: used for representing alignment of a spatial entity represented as a point; this subclass has several attributes and may need to be modelled as an entity;
- DEMpoint: used by the surface subclass Digital Elevation Model (DEM); it is modelled as an attribute value;
- Arc: is the locus of points connecting a list of points; it is modelled as a value;
- ArcDirected: is used where some kind of movement or directions exists along or across the arc; this subclass has an attribute ('flowDirection') and may need to be modelled as an entity or replaced by several enumerated values, e.g., 'ArcDirectedForward', 'ArcDirectedBackward', 'ArcDirectedAcross', etc.;
- OrientedArc: is a type of vector line which specifies both an arc traversal direction, either forward or backward, which conveys the direction in which the arc should be traversed; this may be used to correlate with external information such as street addresses increasing as the arc representing the street is traversed, or the interior of an area being designated as being on the right side of the arc as a polygon boundary is traversed; the traversal direction has no direct physical meaning; this subclass has an attribute ('transversalDirection') and may need to be modelled as an entity or replaced by several enumerated values;
- Breakline: is used by the surface subclass 'MeasuredSurface'; this subclass has an attribute ('breaklineType') and may need to be modelled as an entity or replaced by several enumerated values;
- Contour: is used for specifying the z values (elevation) for all points on an arc; it is modelled as an attribute value;
- BoundedArea: is used for expressing concepts similar to that of 'Polygon' or 'PolygonWithHoles', without the rigorous rules applied to boundary construction for the latter; for purposes of corporate data modelling, it is modelled as an attribute value;

[6] Descriptions are adapted from *Spatial Archive and Interchange Format: Formal Definiton,* Release 3.2, January 1995, Surveys and Resource Mapping Branch, B.C. Ministry of Environment, Lands and Parks. [R-5]

- MeasuredSurface: is used for representing surfaces defined by a series of irregularly spaced points; it is a complex class, and for purposes of corporate data modelling, it is modelled as an attribute value.

Some of these subclasses have attributes (as indicated in the list above) or additional subclasses, or may contain subclasses. There may be some restrictions on these relationships. From a corporate data modelling perspective most of these attributes and lower level subclasses are of a very spatial nature and will not be utilized. As a consequence, the 'GeometricObject' subclasses themselves can be treated as attributes when modelling feature geometries. Note that 'SpatialObject' contains 'GeometricObject's, and that 'GeographicObject' contains 'SpatialObject's. As a consequence, the allowable geometric object types available for describing 'GeographicObjects' (which is a supertype of 'Feature') is limited to the above list.

3.2.5 Corporate Data Modelling Using SAIFLite

The approach for corporate data modelling is to model user defined subclasses (or entity subtypes) of the kind shown under the 'Feature' subclass of the 'GeographicObject' class in Figure 3.4, e.g., features such as 'Road's, 'Forest's and 'ForestStand's. Since 'Graph' is not part of SAIFLite, it is sufficient for corporate data modelling purposes that graph information such as stream networks and highway systems is modelled using simple classes or perhaps 'PartitionedCoverage'. For example, a 'StreamNetwork' (can be a subclass of 'PartitionedCoverage') contains 'Stream's (is a subclass of 'Feature'). Along with each subclass, a 'geometricObjectType' attribute based on the SAIFLite 'GeometricObject' subclass list above will be captured, giving the end user an idea of what kind of shape to expect , for example, 'Point', 'Arc', 'BoundedArea', etc. If there is a requirement to provide some detail for some of these 'geometricObjectTypes', they may be modelled as entities with attributes, but no subtypes.

The user definable subclasses of the 'GeographicObject' class are represented by entities with as many simple attributes as possible. For example, attributes for entity 'ROAD' in Designer/2000 notation could be 'SURFACE_TYPE'[7], 'NUMBER_OF_LANES', 'DIVIDED' (a Boolean 'yes' or 'no'), 'UNDER_CONSTRUCTION' (also a Boolean), and 'GEOMETRIC_OBJ_TYPE_CD' (geometric object type code). See Figure 3.5 for a Designer/2000 ERD of 'ROAD' and its relationships to 'GeographicObject' and 'GeometricObject'.

The 'GEOMETRIC_OBJ_TYPE_CD' attribute has a restricted set of values for each entity. For example, a 'RIVER_STREAM' may be of geometric object type 'OrientedArc' or 'ArcDirected'. These values are specified as 'Allowable Values' in a domain, which is associated with attribute 'GEOMETRIC_OBJ_TYPE_CD' in Designer/2000.

Allowable object position coordinate information that can be modelled is two or three dimensional coordinate points as per SAIFLite (classes 'Coord2D' or 'Coord3D'). No aggregates are allowed. A spatial reference attribute must also be specified.

Allowable time information that can be modelled consists simply of date and time or a timestamp, with no other aggregates allowed.

[7] The use of upper case letters and underscores is a limitation of Designer/2000.

3.2.6 Spatiotemporal Entity Relationships

The ERD in Figure 3.5 shows the relationship between a regular entity (‘ROAD’) and the SAIF ‘GeometricObject’ class and subclasses. In the ERD, the relationship between allowable subclass ‘Arc’ and entity ‘ROAD’ is implemented by associating a domain with allowable value ‘Arc’ with attribute ‘GEOMETRIC_OBJ_TYPE_CD’. This means that instances of ‘ROAD’ will be restricted to a ‘GEOMETRIC_OBJ_TYPE_CD’ of ‘Arc’. Note that ‘FOREST’, which is a composite of ‘FOREST_STAND’, is also assigned a ‘GEOMETRIC_OBJ_TYPE_CD’ value of ‘BoundedArea’, which is the composite geometry that results from its containment of one or more ‘FOREST_STANDS’, which are ‘BoundedArea’s.

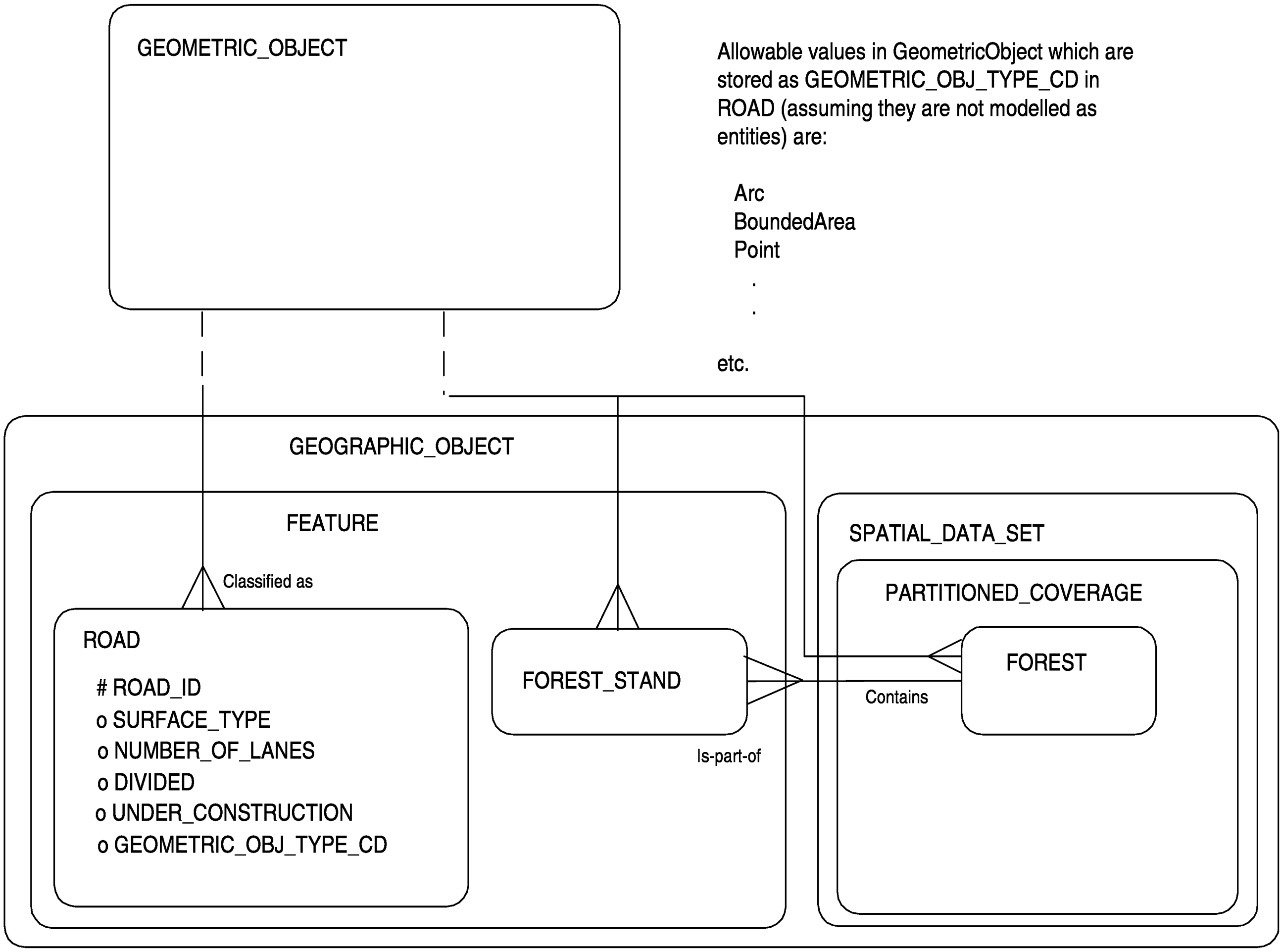

Figure 3.5 SAIFLite Geometric Classification in Designer/2000

3.2.7 Corporate Data Modelling Example

The example focuses on a single entity, ‘ROAD’, and some of its relationships for purposes of demonstrating how spatiotemporal modelling is done in a corporate data modelling context. The databases which were used for this example contain references to ‘ROAD’ or ‘ROAD’ subtypes. They are:

- TRIM,
- Baseline Digital Atlas (BDA) 1:250,000 (referred to as ‘BDA250’ further on),

- Baseline Digital Atlas 1:2,000,000 (referred to as ‘BDA2MIL’ further on), and
- Crown Land Registry (CLR)/Cadastral Data Management System (CDMS).

Entity Relationships

‘Road’ is a subtype of ‘Feature’ in all models. ‘Feature’ is associated with a temporal entity called ‘UpdateHistory’ in the BDA250 and BDA2MIL schemas, with plans on providing it in TRIM. The following diagram is a Conceptual Data Model ERD showing the basic components being analyzed:

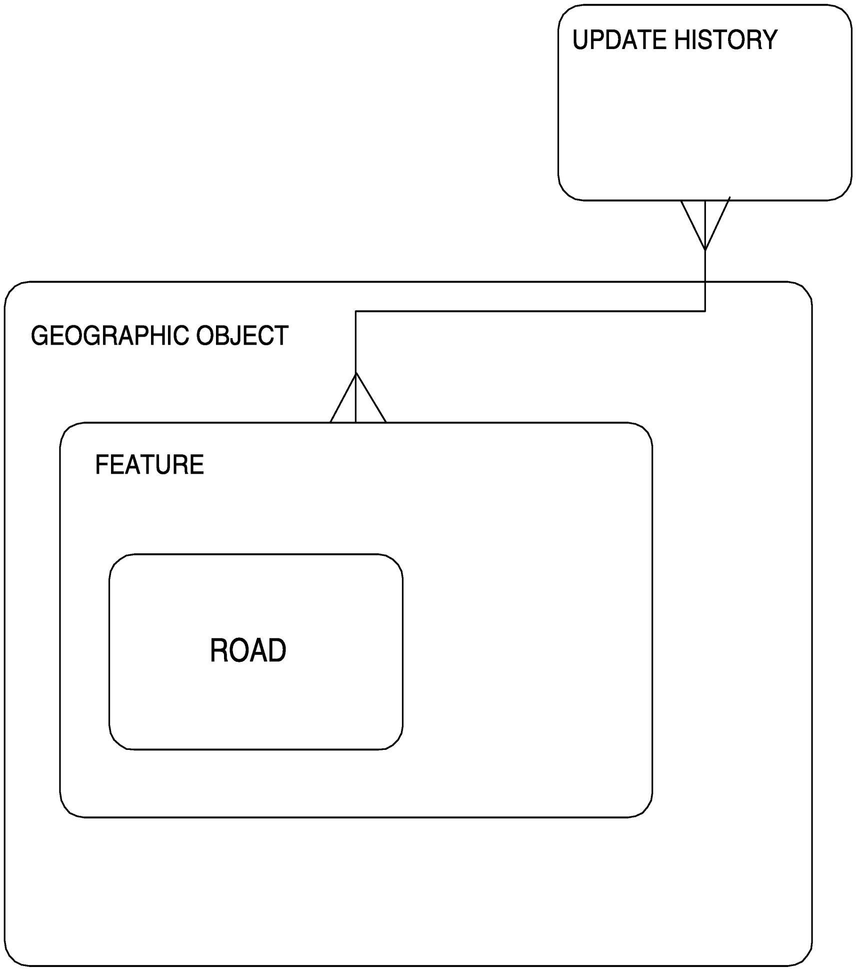

Figure 3.6 Conceptual Data Model for “Road”

Attributes

The following attributes were derived from the SAIF models[8]:

UpdateHistory:[9]

[dateOfSourceData] Date10

date of source data capture (aerial photography, satellite imagery, DGPS11,etc.)

[admissionDate] Date

date of admission to file archive or database (data now available for external access).

[admissionStatus] Boolean

distinguishes between a feature added to the dataset as a result of new compilation where no representation of the feature existed in the dataset previously (F) and a feature(s) added to the dataset to replace an existing feature (T)

[obsoleteFeature] Boolean

distinguishes between features which are part of the most current view of the data (F) and features that not part of the most current view of the data (T)

[dateOfObsolescence] Date

date when the feature was removed from the most current view of the data

[admissionRevisionKey] String

revision version - may in the future be used as a key to other information

[admissionSpecificationRelease] String

TRIM specification version used at the time of revision data capture

[retirementDate] Date

date of retirement from file archive or database (data no longer available for external access).

[8] At time of writing, most models are still under development or temporary.

[9] Currently proposed for TRIM, assumed to be used by other ‘Feature’ classes for purposes of this example. Entities in this example are bolded.

[10] The item in square brackets is an attribute, the text following it is a data type, e.g., ‘Date’.

[11] Differential GPS (Global Positioning System)

[retirementStatus] Boolean

T - object is replaced by a new object. F - object is not replaced by a new object

[retirementRevisionKey] String

revision version - may in the future be used as a key to other information.

[retirementSpecificationRelease] String

TRIM specification version used at the time of data retirement

Feature::BDA250[12]

none

Feature::BDA2MIL

none

CDMSFeature::CLR

[area]	Real
[perimeter]	Real
[component_id]	Integer
[plan]	String
[cdid]	String
[pin]	String
[descriptor]	String
[file]	String
[ir_code]	String
[ir_name]	String
[ir_number]	String
[isa_number]	String
[isa_name]	String
[combined_factor]	String
[component_type]	String
[number]	Integer
[id]	Integer

[12] The name following the double quotation is the schema name

Feature::TRIM

[originalFileType] values: dem contours nonPositional planimetric toponymy

[displayType] values: primary primaryLine primaryCurve constructionLine constructionCurve

defaults:

displayType: primaryLine

originalFileType: planimetric

Road::BDA250

[surfaceType] values: loose paved winter cartTrack dryWeatherLoose pavedNonStandard

[numberOfLanes] values: one two twoOrMore threeOrMore

[divided] Boolean

[underConstruction] Boolean

[geometricType] values: Arc

defaults: surfaceType:paved

numberOfLanes: two

divided:false

underConstruction:false

Road::BDA2MIL

[type] values: local main trunk winter

[geometricType] values Arc

ForestRoad::CLR

[type] values: serviceRoad serviceProtectionTrail roadRightOfWayPermit

[geometricType] values Arc

PetroleumDevelopmentRoad::CLR

[geometricType] values Arc

Road::TRIM

[surfaceType] values: loose paved rough

[numberOfLanes] Integer8

[travelDirection] values: oneWay twoWay

[divided] Boolean

[underConstruction] Boolean

[geometricType] values Arc

defaults: surfaceType:paved

numberOfLanes:2

travelDirection:twoWay

divided:false

underConstruction:false

numberOfLanes: 1 | 2 | 3 | 4 | 6

position geometry restricted to Coord3D

Observations

- ‘Feature’ entity: all models contain a high level entity called ‘Feature’, which is qualified for the CLR as ‘CDMSFeature’. The ‘Feature’s for BDA250 and BDA2MIL are identical, but the ones for TRIM and CLR are quite different (i.e., they contain numerous different attributes).
- ‘Road’ entity: all models contain references to ‘Road’, some with qualifications.
- Some domains overlap, e.g., ‘Road[surfaceType]’ for BDA250 and TRIM.
- Some attributes have the same names but take on different datatypes and values, e.g., ‘Road[numberOfLanes]’ for BDA250 and TRIM.
- Some attributes have the same names and take on the same meaning, e.g., ‘Road[divided]’ for BDA250 and TRIM.

Modelling Approach

The goal of the modelling exercise is to build an integrated LDM. The Classification Hierarchy Subject-Area[13] is ‘TRANSPORTATION UTILITIES’, and the relevant Datagroup below it is ‘ROAD (TRANSPORTATION UTILITIES)’. By modelling at the Datagroup level, end users will be able to search for ‘ROAD’ information on a corporate wide level.

- Since the attribute sets for the different ‘Feature’s contain significant differences, they will be modelled as subtypes called ‘Feature BDA250’, ‘FeatureBDA2MIL’, ‘CDMSFeature’ and ‘FeatureTRIM’. Their common supertype is ‘Feature’, whose primary purpose is to link to ‘UpdateHistory’.

[13] Refer to [R-1] for information on the Classification Hierarchy

- Since the attribute sets for the different ‘Road’s also contain significant differences, each ‘Featurexxxx’ subtype will have a ‘Roadxxxx’ subtype, each with a supertype of ‘Road’ (multiple inheritance with ‘Featurexxxx’s). In this way, it will be easy to locate ‘Road’ related entities in the model. Common attributes will be kept in ‘Road’.
- Value lists for ‘surfaceType’, ‘type’ (of road), and ‘numberOfLanes’ can be modelled in at least two different ways:
- in Designer/2000 as domains with allowable values; i.e., each ‘Roadxxxx’ will have a ‘surfaceType’, ‘roadType’, and ‘numberOfLanes’ attribute, each with a specified domain; or
- as code tables in the LDM, populated with each code value tagged with a schema name, e.g., a ‘surfaceType’ code table would look as follows:

loose	BDA250
loose	TRIM
paved	BDA250
paved	TRIM
rough	TRIM
winter	BDA250
	.
	.
	.

In this way, both unique and overlapping values can be stored and easily retrieved. By storing this data in tables rather than the ORACLE Repository, it makes end user web access much easier. Schemas which have ‘Roadxxxx’s that do not have any corresponding attributes are simply coded as ‘none’. For example,
none BDA2MIL.

Consolidated Model

The following ERD shows the consolidated spatiotemporal LDM:

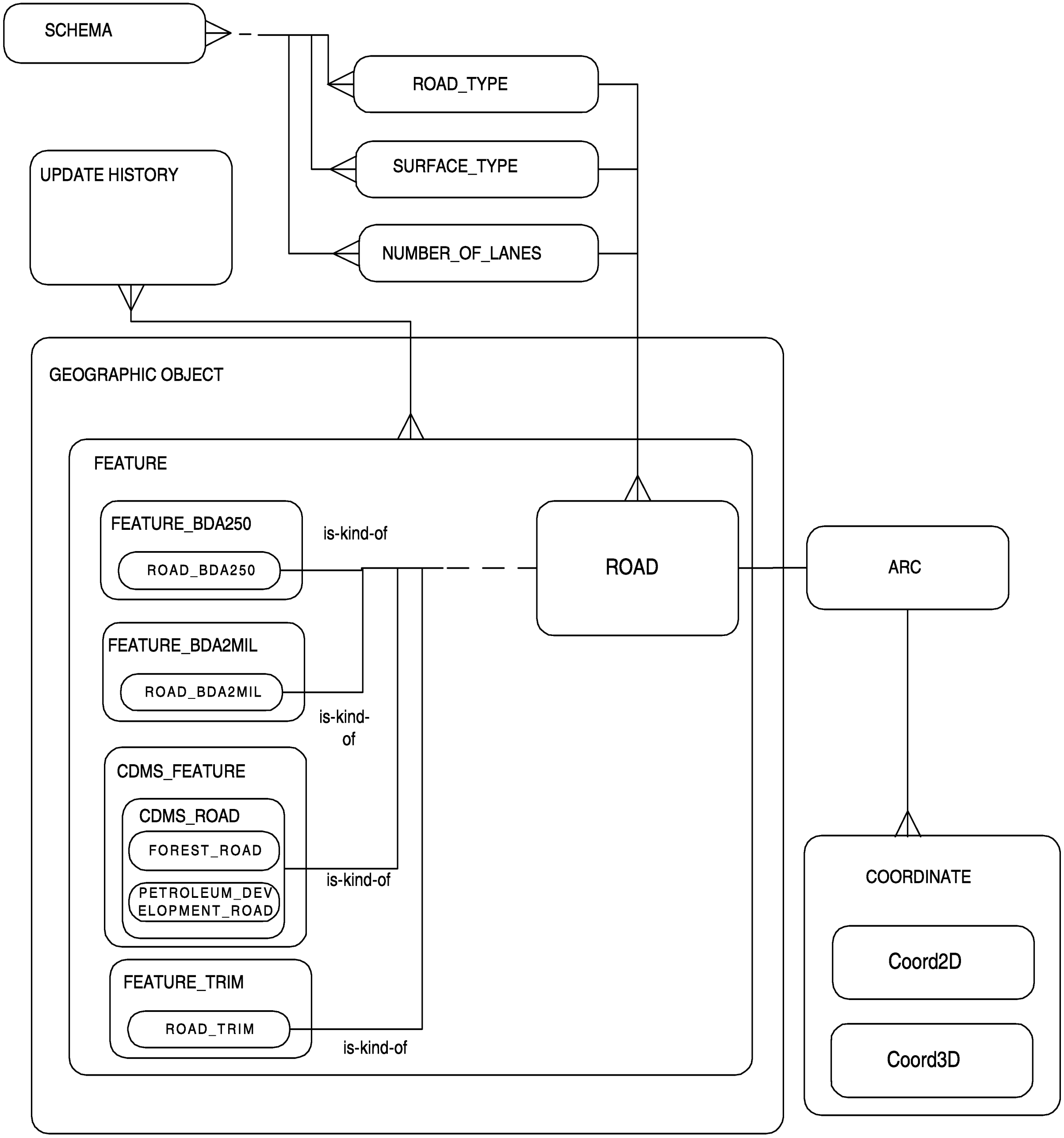

Figure 3.7 Consolidated Spatiotemporal LDM

Integration With CDM

The above model is incomplete (there are some many-to-many relationships that need to be resolved), and for sake of brevity, attribute lists are not shown. At this point, the model would be compared to an existing sub-federated LDM under ‘TRANSPORTATION UTILITITIES’, and integrated with it as described in Section 2.1.3, *Creating a Sub-Federated Model*. In the above example, the allowable ‘GeometricObject’ value ‘Arc’ is modelled as an entity, that is, it will be modelled with attributes. This is in contrast to the approach shown in Figure 3.5, where it is modelled as an attribute. Both approaches are acceptable and depend on the level of detail required or available.

Appendix A

B.C. Environment
Data Administration Practices #1–
Logical Data Models

The material in this appendix has been drawn from a number of sources. It is derived in part from Ministry of Forests Data Administration Practices. Discussions with the Data Administrator at B.C. Ministry of Forests and their Data Modelling Guide formed an initial source of material for this material. It has been significantly augmented by input from members of Systems Services Branch and key systems professionals in B.C. Environment.

The information contained in this appendix is available on the Web. See the Ministry of Environment, Lands and Parks (MELP) home page under System Support.

Data Administrator
Systems Services Branch
B.C. Environment
Version 1.2 - June 1994

Introduction

In May 1994 the need for data models has increased, driven by the ease in which information can be delivered to sophisticated users with Oracle and BC Environments LANs.

This document outlines the rationale, and policy, procedures, and standards for the use of Logical Data Models in the Ministry of Environment.

The document will primarily be of interest to those involved in developing and maintaining systems for BC Environment, for those who must access information and understand the meaning and structure of the data and for the decision-makers who resolve issues arising from systems development and implementation.

Logical Data Models are diagrams and a structured description, to a recognized industry standard, of the data, its meaning, and inter-relationships, for the data pertinent to the solution of a specific business problem. Logical Design Models will be referred to as LDM subsequently in this document.

The production and review of the data models provides a strong medium for anticipating and understanding problems which will arise in the use of data and the systems which interact with it.

Within the information systems industry modeling is a well recognized and universally performed process. There is a common agreement of its necessity in the systems development process.

Organizations which produce and review logical data models in a structured fashion benefit by identifying and resolving some of the issues which will require costly solutions after the implementation of the system.

This document proposes the rationale, policy, procedures, standards, responsibilities, and implementation for Logical Data Models in BC Environment.

Intent

The intent of the introduction of data models in BC Environment is ensure that the understanding of a proposed system by its developers is comprehensive, and to provide a medium to communicate that understanding to stakeholders beyond the immediate scope of the project who may be impacted by the system. Particularly, it provides a platform for ensuring data sharing and consistency within the organization. The process of producing the models may expose issues which must be resolved in order to avoid problems in using the data.

Policy

Logical Data Models will be produced and reviewed during the development or maintenance of all systems which cost over $7,000. The procedures, standards and responsibilities are specified in this Logical Data Model Document. The Data Models will not be considered complete until signed off by the Project Manager, the Data Administrator (and the Program Data Administrator where applicable) and the Application Manager.

The **LDM Process** outlines how models will be produced and reviewed during the development of systems. The **LDM** Standards outline the specific requirements tested for in the Model Reviews. The **LDM Responsibilities** outline the persons or organizational roles responsible for producing endorsed models.

LDM Process

Hold Project Data Planning Session

At the initiation of a project, the Project Manager will hold a data planning session in conjunction with Data Administration to establish the broad data requirements for the project. The number and size of models will be determined and the appropriate standards will be identified. Sources for some models may also be identified. The data planning session should be held early in the project, well before formal requirements are resolved. Data Administration will be responsive and cooperative in the participation in the planning session.

Develop Data Structure and Definitions

During the systems development process the Project Manager will ensure that Entity-Relationship Diagrams and a Data Dictionary are produced, to standard, for any data pertinent to the scope of the project. In some cases, where models exist, they may be available from the Data Administrator, and need only be modified and augmented. A CASE diagramming tool may be used to develop the models. The models should be completed before any detailed design or coding is started. The standard for diagrams and dictionary are in the section LDM Standards below.

Hold Logical Model Reviews

When an appropriate collection of data is prepared, the Project Manager will schedule a Logical Data Model Review two weeks in advance and will provide the models to the Data Administrator one week in advance. The Review will address how the models meet the standards and will surface, and document, with risks, any issues which may impact the project or the rest of the organization. In the absence of concerns, the model may be approved at the review. The Model will be approved when deviations from standards have been remedied or minimized. Issues and concerns arising from the Review will be documented, including risk and impact and, and significant problems will be raised to the projects steering committee, or other appropriate resolvers.

Resolve Significant Issues

This is the hardest step. Typical issues may be that the data already exists in an awkward system, or that all the needs of potential users of the data cannot be economically met. The identified issues should be fairly represented and explicitly raised to the appropriate decision making level.

LDM Process In Plain Terms

- Hold a meeting very early in any systems project with key stakeholders, including Data Administration and data users, to specifically anticipate issues concerned with the data itself.
- Explicitly define all of the data which will be used and show how it interrelates. Show it to key stakeholders, and discuss issues which arise. Use the data models to portray the structure.
- Ensure that issues are appropriately disclosed to decision makers.

LDM Standards

The intent of the Logical Data Model is to document explicitly, a comprehensive understanding of the data to be stored and delivered in an application. Documentation provided for the review should be to a level of detail so that little other information is needed to read and understand the structure of the data as a whole. The presented model may refer to readily available documents, or append any ancillary information as necessary.

The Logical Data Model (diagrams and dictionary) will be produced on a recognized CASE tool. Currently recognized CASE tools include: ORACLE CASE, ERWin, DEFT, EXCELERATOR, IEF, IEW. Other forms of presentation should be confirmed with the Data Administrator in advance of starting documentation.

Entity and attribute names must conform to BC Environment's Naming Standards.

The intent of Logical Data Models will be kept in mind in the interpretation of these standards.

The intent of systems development is to produce a working application and the intent of these standards is not to impact that process, but to enhance it, to the benefit of the organization as a whole. Where the efforts to produce the Logical Data Model appears excessive, and it can be shown that the risks the models attempt to expose are understood and resolved elsewhere, the standards may be explicitly relaxed by the Data Administrator.

Where significant maintenance is to be performed on a system which has no models, data models must be provided for the parts of the system which are significantly altered. Where maintenance is performed on systems with existing models the models must be modified to reflect these standards.

Use of the Data Models

The models will be used by those who access the data using modern query tools. For data warehouse applications the tools include Oracle Browser and Oracle Glue.

The models will also be used to develop and assist in the maintenance of the application.

Entity Relationship Diagram

This is a graphical or pictorial representation of the entity types and their relationships, defined by an analysis of what information the business records and what business rules are used.

- It is at the Logical level. It does not address physical implementation needs.
- Each entity type is named and represented by its own rectangle.
- Relationships between entity types are shown as a line connecting the two entity types.
- Each relationship is given two names for the information it imparts between the entity types.
- The cardinality and optionality of the relationship is shown at each end.
- The diagram should only show those attributes which make up the key for the entity.

Standard Notation

Where a standard CASE tool is used the notation will be constrained by what is available in that tool. Where there are choices, the Bachman notation is preferred. In general, entities are shown as rounded rectangles, relationships are not curved and cardinality is shown by ‘crows feet’ and optionality by circles or crossing lines.

Large Data Models

When the number of entities in a data model cannot be clearly shown on one page, the models should be split into ***Entity Display Groups (EDG)***. An Entity Display Group is a group of entities which have a business coherence, and are relatively decoupled from other entities. Every entity will be in one and only one EDG.

The models delivered must include a page for each EDG showing the relationship of the entities within that single EDG. On this page, the included entities will be shown within a rectangular boundary. All entities which have a relationship to those in the facet will be shown on the page outside the rectangle. All relationships between any pair of entities on the page must be shown.

To complete the model a Entity Display Group Map must be provided to represent all of the Entity Display Groups on one page to act as a table of contents for the following models.

The Map will have one labeled box for each Entity Display Group and will have a single line joins any of the EDGs which share a relationship.

An example of an Entity Display Group is included in the lst section **LDM Samples and Support**.

Entity Naming Standards

Names for the entities must be meaningful and must not conflict with other entity names. Where abbreviations are used the full words must be obvious.

Entity Definition Standards

1. Instances

   An Entity must have instances.

2. Nouns

   The first sentence of the definition should give a noun or group of nouns, with modifying adjectives or phrases which summarize the meaning of the entity.

   E.G.: An Employee is any person, alive or dead, who works or has worked, for the company.

3. Examples

   Typical examples should be included.

   The first example should be one which is firmly in the collection, which will solidly reinforce the readers understanding. Other examples should reveal the bounds of the entity. An example which might not be considered as an instance will help to establish the boundary. This requires anticipating instances where there may be confusion for the reader. Also of value is an example of an instance which is not a valid member of the group. This shows the boundary from the other side. Often wording which explains why examples are, or are not, included are necessary. The definition should include clear explanation of why these examples are, or are not, entities.

   E.G.: As soon as a person becomes an candidate for a position in the company they become an Employee whose status is un-hired. This allows personal information to be entered only once. Applicants who are not considered for a position are not Employees.

4. Intent

   Where possible the intent of the entity should be shown. This allows the reader to understand the rationale for the entity and fit their own understanding of the business into the one represented by this entity.

   E.G.: The intent of the Employee entity is to capture any information about a person the company deals with in an employee employer relation. This includes personal information, tax and statute information and skills and capabilities. Information relating to contracted individuals is kept with the client entity.

5. Differentiation

   Where there may be confusion between two similar entities, either because their names are similar or because their descriptions or relationships are similar, there should be explicit text which explains the difference.

   E.G.: The Worker entity may be confused with the Employee entity. The Worker entity is used to collect information about clients who are supported in the work they do.

Attribute Definitions

Each attribute must be clearly named and defined. If the attribute is from a domain which will not change, the values should be listed or described. At the physical level these domains will be implemented as code tables but it is not necessary to show these for the logical model.

Logical Model Checklist[14]

At the Logical Model Review, the following is considered:

- Identification of cross-functional entity types that may require other branch representatives to attend. Cross functional entity types describe data used within a number of functions in the Ministry.
- Technical quality or the Entity-Relationship Diagram
    - Does the cardinality (one-to-one, one-to-many, many-to-many) and optionality (sometimes, always) of each relationship reflect the true business information requirement.?
    - Were relationship names chosen properly? Are they meaningful, do they describe the business?
    - If multiple relationship paths between the same entity types are really required, what are the reasons?
    - Are many-to-many truly many-to-many? (most are not)
    - Are mandatory one to one relationships really required or should the two entity types be collapsed into one?
- Technical quality of the data dictionary.
    - Do entity types and attributes clearly indicate the meaning of what they store?
    - Do entity type definitions accurately describe the business usage of what the entity type contains?
    - Are entity type identifiers properly defined?
    - Do entity types have their custodians identified?
    - Do attributes fit the entity type? (proper normalization)
    - Do the names follow naming standards for full names and abbreviations?
    - Do attributes containing codes include a representative sample of code values?

Data Dictionary

A Data Dictionary is considered as an integral part of the required deliverable. It provides a definition of the entity, attributes, and relationships included in the diagrams.

The definitions in the dictionary should have the following characteristics:

Precision	- they resolve ambiguities and qualify imprecise terms.
Completeness	- all terms are defined.
Clarity	- plain English, few if any buzzwords.
Brevity	- brief and to the point.
Compatibility	- with other definitions.

[14] Items in the checklist and the Data Dictionary definition characteristics on the next page are taken from *Systems Development Guide S7: Data Modelling Guide*, Information Systems Branch, Ministry of Forests; Contact: Data Administration section.

LDM Deliverables

1. A scheduled data-planning meeting, including agenda. (Project Manager)
2. Scheduled Reviews of suitable size. (Project Manager)
3. A Logical Data Model consisting of entity-relation diagrams and data dictionary. (Project Manager)
4. The Review's feedback document. (Data Administrator)

Data Model Sign-off: The logical model will be considered complete when unconditionally signed off by the Project Manager, the Data Administrator, the Application Manager and the models author.

LDM Responsibilities

Project Steering Committee and User Management

- Identify Project Manager and charge him/her with following the LDM process.

Project Manager

- Follow LDM process. Perform Quality Assurance.

Project Participants

- Prepare Logical Data Models.

Data Administrator

- Provide consultation. Interpret standards. Expedite development. Produce feedback.

Steering Committee and Managers

- Resolve issues.

RFP Requirement

The following section should be included in all RFPs for analysis or development of systems for BC Environment.

Logical Data Models

BC Environment has standards in place for producing and reviewing Logical Data Models as part of the systems development process. The production of these models, and their review are considered a requirement of any analysis or development project. The standards call for an initial data planning meeting at project kickoff, the production of entity-relationship diagrams and a data dictionary for all pertinent data , and reviews of the models. Up to 8 entities and their attributes (i.e., one Entity Display Group) can be reviewed in one meeting. The model is not considered complete until signed off.

Further details and the document LOGICAL DATA MODELS available from the Data Administrator, BC Environment.

Ongoing Updates to Standards

With the standard in place, proposals for their improvements will be periodically assessed and a new version will be produced. This process relies strongly on users of the standards to identify frustrations and corrections. The Data Administrator will compile proposed changes and at least every year will convene key convention users to implement the changes.

Where proposal are un-contentious and an on-line method of delivering the conventions is available, the changes will be made immediately.

LDM Samples and Support

The Data Administrator will provide examples of Entity Relationship Diagrams and a Data Dictionary on request. References and further material on modeling are also available. For all projects, the Data Administrator will work with the developers to ensure good quality models are produced.

Entity Display Group - Example

The following example was produced to illustrate the standard for Entity Display Groups and Summary Key Maps. An existing model was chosen and divided into six Entity Display Groups. Because it was prepared in order to expedite the understanding of the standards, the logical quality of the diagrams may be suspect in some places. For models delivered for review the modeler's familiarity with the material will make the delivery of a logically supportable and coherent breakdown relatively straightforward. The Key Map is not a formal Entity Relationship diagram, and cardinality and relationship names should not be shown. The names used on the Key Map should be identical to the title of each Entity Display Group. Every Entity in the overall model should appear once and only once within the delineated areas on all the EDGs. Outside the delineated areas the Entities will appear when they participate in a relationship which joins two EDGs.

In the example an existing, one page model is also included as an illustration of the scope of the whole model. The delivery of a one page model is not a requirement, although modelers occasionally find them valuable working tools. The included model has been carefully prepared and was relatively easy to subdivide. SITE, the central entity, has seven external relationships, EVENT 4, PARTICIPANT 6, CONTAMINANT 2, PERSON 2, and SUBDIVISION 1. If an overall model is a challenge to subdivide it is often an indication that the model requires further modeling efforts.

Three example pages are included:

1. Existing Model with the Entity Display Group subdivision shown.
2. An Entity Group Key Map showing the 6 EDGs
3. A single EDG.

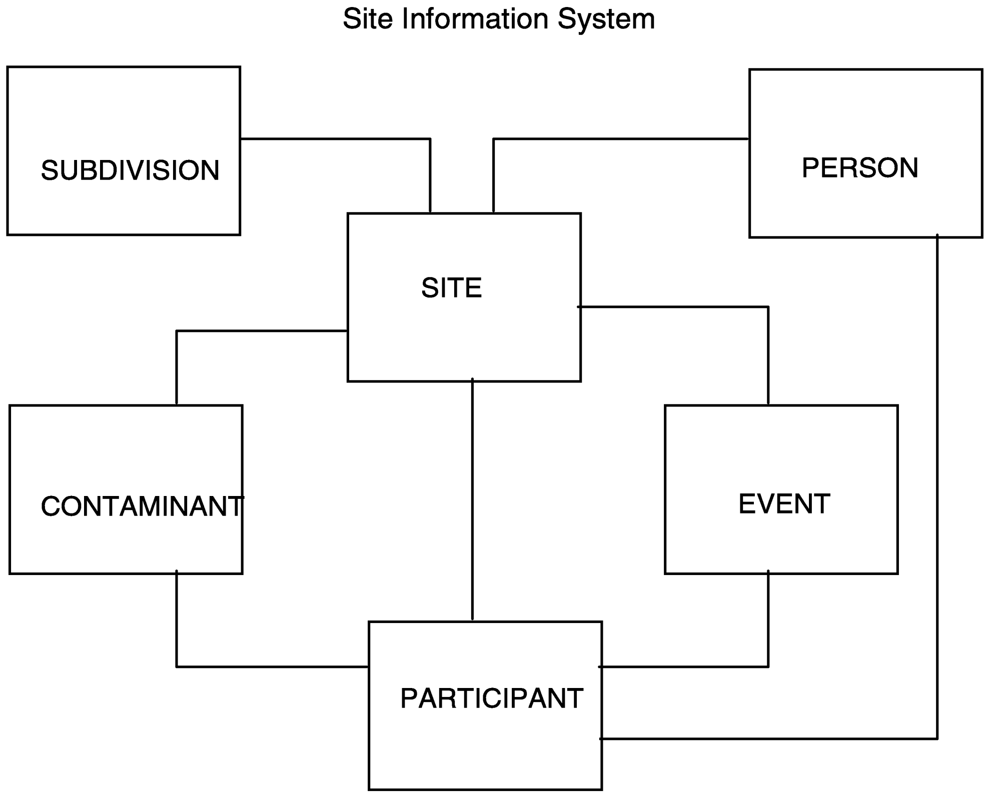

Figure A.1 Entity Group Key Map (Sample)

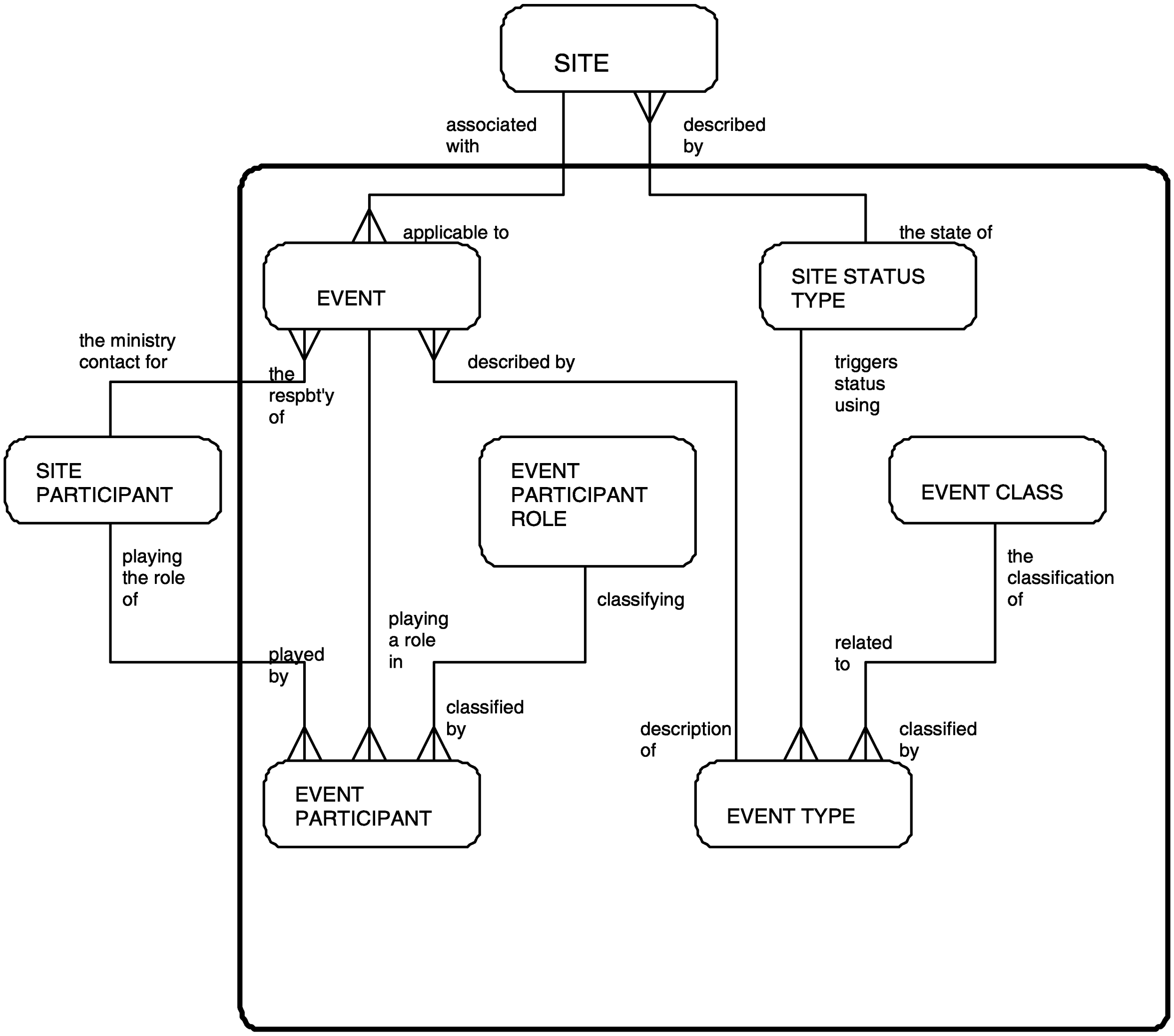

Figure A.2 Entity Display Group (Sample)

B.C. Environment
Data Administration Practices #2–
Naming Conventions

Naming Standards from a number of other organizations have been used in the development of this document. Standards from the Ministry of Forests and the Ministry of Health have been particularly helpful. Thanks go to the respective contributors to this documents.

Data Administrator
Systems Services Branch
BC Environment
June 1993

Introduction

In May 1992 the Role of Data Administrator was filled in BC Environment. In the absence of Data Management policy in the Ministry, it was recognised that there are some common procedures which would have immediate benefit for BC Environment. Logical Data Models and Naming Standards were two identified focuses. The third was Data Management Factors such as currency and security which must be addressed in implementing data systems.

This document outlines the rationale, and proposed procedures, and standards for application components in the Ministry of Environment. As a proposed standard it is open for comment and suggestions. Its intent is to present these standards for review and endorsement.

This document is addressed to decision makers and data practitioners.

Naming Standards are written procedures and tables used to develop a consistent name for the objects used for the development and implementation of systems in BC Environment.

When naming standards are used, any person working with these objects can make intelligent assumptions on the nature of the object, and can also be assured that the likelihood of synonyms and duplicate names is reduced. Developers efforts can be focussed on developing systems and not on developing another way of naming things.

Intent

The intent of the introduction of naming standards in BC Environment is ensure that

effort in development is addressed to the system at hand, and that objects are consistently recognised and referred to. In addition migrations of data from one system to another are facilitated and the integration of systems will be easier.

Particularly it provides a platform for ensuring data sharing and consistency within the organization. It improves communications for both developers and users.

Policy

The BC Environment standards for naming data elements will be used during development of all systems.

The **Data Administrator is responsible** for maintaining these standards. Periodic updates will be made. Comments will be compiled for the update process. Stakeholders will participate in update of these standards.

Systems developers and project managers are responsible for ensuring the conventions are followed, and that improvements are suggested to the Data Administrator.

Naming Standards

The following process should be used for deriving a full name for an object to be used in BC Environment. Once the object is named, the abbreviation process should be used to develop shorter names where necessary.

Deriving the FULL NAME for an object:

The general form of the Full Name is:

MODIFIERS : ENTITY_NAME : MODIFIERS : CLASS_NAME

EG. Casual Employee Hire Date

Where necessary underscores (or if unavailable hyphens) should join the words. The full name should be in upper case.

Therefore CASUAL_EMPLOYEE_HIRE_DATE

A Class is a collection of descriptors for an entity, usually with a standard representation (data-type). Examples are NAME, DATE, AMOUNT, KEY etc..Some classes are directly supported by a datatype in programming languages, others must be specified for a particular environment. A list of the current standard classes in BC environment is included in this document.

Some data elements are common to many applications. These are enumerated in the STANDARD ELEMENTS. The standard elements should be checked and used when developing elements.

Classes

Classes (sometimes called domains) are standard representations for collections of objects. Often data elements for many different applications will be built from the same classes.

For example BIRTH_DATE, HIRE_DATE, and CONTRACT_DATE all have the same class, but apply to different entities. One or more standard attributes will be developed for the following classes(some are apparent from the standard elements). The standard class name or its abbreviation should be used in the development of element names.

Standard Class Name	Abbreviation
ADDRESS	ADRS
ADDRESSLINE	ADRSLN
AMOUNT (A numeric value of money)	A,AMT,AMNT
CODE	C,CD,CDE
COUNT	CNT
DATE	DT
YEAR	YR
MONTH	MNTH
DAY	DY
DESCRIPTION	DSCRPTN
FLAG	FLG
KEY	KY
LOCATION	LCN
NAME	NM
NUMBER	NMBR
SERVICE	SRVC
SEX	SX
STATUS	STTS
TEXT	TXT
UNIT	UNT

Abbreviations

These standards provide a means to determine standard abbreviations for use in defining application components.

Standard

No abbreviations may duplicate those appearing on the approved list. An abbreviation of 4 characters cannot be a word. Abbreviations are determined for words/terms that do not appear on the approved list as follows:

1. Determine length of abbreviation.
2. Put first and last letter of word in first and last position of abbreviation.
3. Remove vowels.
4. Remove one of any double consonants.
5. Fill remaining spaces of abbreviation with consonants of the word in the order in which they appear until the required number of letters is obtained.

Example

1. To abbreviate ‘ABBREVIATION’ to 5 characters.
2. Put ‘A’ in location 1 and ‘N’ in location 5.
3. Remove ‘E’, ’’I’, ‘A’, ’’I’, and ‘O’.
4. Remove one ‘B’.

5. This leaves ‘BRVT’ for 3 spaces.
6. Select appropriate characters and fill spaces 2 to 4.

   Result is ABRVN

Standard Acronyms and Abbreviations

ADMINISTRATION	ADMN
ALTERNATE	ALT
APPLICATION	APPL
AUTHORITY	AUTH
BRANCH	BR
BUSINESS	BUS
CANADIAN	CDN
CLASSIFICATION	CLASSN
CLIENT	CLNT
COMMENT	CMNT
COMMITTEE	CTTE
COMPANY	CMPNY
CONTROL	CTL
CORPORATION	CORP
CREDIT	CR
DESTINATION	DEST
DEPARTMENT	DEPT
DISTRICT	DIST
DIVISION	DIVN
EFFECTIVE	EFF
ERROR	EST
EXECUTIVE	EXEC
FEDERAL	FED
HECTARES	HA
IDENTIFICATION	ID
INDEX	INDX
INITIALS	INTLS
INDICATOR	IND
LENGTH	LNTH
LICENCE	LIC
LOAD	LD
LOCATION	LOC
MANAGEMENT	MGMT
MARK	MRK
METHOD	MTHD
MINISTRY	MNSTRY
NAME	NM
ORGANIZATION	ORG
PAYMENT	PYMNT
PERCENT	PCT
PERMIT	PRMT
PIECE	PCE
POSITION	POSN
PRIMARY	PRI
PRODUCT	PRO
PROJECT	PROJ
RECEIVED	RECD
REGION	REG
REGISTRATION	REGN
REQUIRED	REQD
RETURN	RTN
REVENUE	RVNU
SCHEDULE	SCHEDL
SEARCH	SRCH
SECONDARY	SEC
SECTION	SCTN
SEQUENCE	SEQ
SERVICE	SRVC
SQUARE FEET	SQFT
SQUARE INCHES	SQIN
SQUARE METERS	SQM
STATEMENT	STMT
STATUS	STS
STATUTORY	STAT
STATISTICS	STATS
TEXT	TXT
TRANSACTION	TXN
TYPE	TYP
USERID	UID
VALUE	VAL
YEAR	YR
YEAR TO DATE	YTD
BCSC	British Columbia System Corporation
SIN	Social Insurance Number

Standard Elements

The following elements are Ministry standards and where appropriate their attributes (length, type) and definitions should be used.

The intent of these is to ensure that when similar data is to be merged of compared, the process is relatively simple. This avoids problems of matching fields of different lengths.

Some applications will need to use these elements in ways that have not initially been foreseen, and may find they do not meet their needs.

Data Administration wishes to maintain consistency and will add new corporate elements where necessary. Please contact the Data Administrator in this case. These standard elements should be used even when the contained data may never interact with other parts of the organization. Deviations should be confirmed with Data Administration.

CLIENT_ID INTEGER 4 ZZZZZZ9

System assigned key. Unique to the client.

The ID will be unique within a particular database. Any client function must provide searches to ensure uniqueness within the database. This is set as integer so that a new number can be generated by adding one. Primarily used internally but may be used externally.

CLIENT_NAME CHARACTER 150

Format 'FAMILYNAME , GIVEN NAMES ' for an individual, otherwise the name of an organization. Set at 150 to allow long organizational names. Searches will be done on this to ensure uniqueness. For individuals only the family name is searched.

CLIENT_TYPE_CODE CHARACTER 1

Indicator of individual or organization. 'I' Individual, 'O' Organization.

ADDRESS_ID INTEGER 4 ZZZZZZ9

System assigned key to ensure address uniqueness.

ADDRESS_TYPE_CODE CHARACTER 2

A code to indicate the type of address. Currently values are for mailing, billing or legal.

ADDRESS_LINE_1 CHARACTER 50

Four address lines hold the unstructured address, excluding city, province, and postal code which are separate elements. Standard data elements for structured address will be developed as the need arises.

ADDRESS_LINE_2

ADDRESS_LINE_3

ADDRESS_LINE_4 all CHARACTER 50 See above.

CITY CHARACTER 30

PROVINCE_STATE_CODE CHARACTER 10

COUNTRY	CHARACTER	30	
POSTAL_ZIP_CODE	CHARACTER	10	X9X 9X9 (if CDN.)
AREA_CODE	CHARACTER	3	
PHONE_NO	CHARACTER	7	
PHONE_EXT	CHARACTER	4	

If different types of phone numbers are stored then the element names should be prefixed with the abbreviation of the type. EG BUS_PHONE_NO, FAX_AREA_CODE etc.

RFP Requirement

The following section should be included in all RFPs for analysis and development of systems for BC Environment.

BC Environment has standards in place for the **naming of objects** in its systems environment. The standards include requirements for developing full names for objects, standard abbreviations, an abbreviation method, and a list of standard elements with their attributes. Any development work done for BC Environment is to follow these naming conventions. Included objects to be named are entities, attributes and data elements.

Further details, and the document "Naming Conventions", can be obtained from the Data Administrator, BC Environment.

Convention Updates

With the Naming Conventions in place, proposals for their improvements will be periodically assessed and a new version will be produced. This process relies strongly on users of the standards to identify frustrations and corrections. The Data Administrator will compile proposed changes and at least every year will convene key convention users to implement the changes.

Where proposal are un-contentious and an on-line method of delivering the conventions is available, the changes will be made immediately.

Glossary–
Acronyms and Abbreviations

3NF	Third Normal Form
BDA	Baseline Digital Atlas
CASE	Computer-Aided Systems Engineering
CDMF	Corporate Data Model Framework
CDM	Corporate Data Model
CLR	Crown Land Registry
DGPS	Differential GPS
DMC	Data Model Committee
EDG	Entity Display Groups
ER	Entity Relationship
ERD	Entity Relationship Diagram
GIS	Geographic Information System
GPS	Global Positioning System
LDM	Logical Data Model
MDA	MacDonald, Dettwiler and Associates
MELP	Ministry of Environment, Lands and Parks
OMT	Object Modelling Technique
OO	Object Oriented
RIC	Resources Inventory Committee
SAIF	Spatial Archive and Interchange Format
SQL	Structured Query Language
TRIM	Terrain Resource Information Management

References

R-1 BC-MA-50-7367 LandData BC Corporate Data Model Framework

R-2 Object Oriented Modelling & Design, James Rumbaugh et al, Prentice Hall, New Jersey, 1991

R-3 ORACLE *CASE*Method Entity Relationship Modelling* by Richard Barker, published by Addison Wesley, 1990

R-4 ERwin *Methods Guide,* Logic Works Inc., 1995

R-5 *Spatial Archive and Interchange Format: Formal Definition,* Release 3.2, January 1995, Surveys and Resource Mapping Branch, B.C. Ministry of Environment, Lands and Parks

R-6 *Introduction, SAIFLite,* Release 1.1, March 1996, at Internet URL http://www.env.gov.bc.ca/gdbc/fmebc/comment4.htm

R-7 *Consolidated Wildlife Inventory Data Model Project*, Habitat Inventory Section, Wildlife Branch, Ministry of Environment, Lands and Parks (MELP), September 1995.

Queen's Printer for British Columbia©
Victoria

7680000656