HIERARCHICAL DATA STRUCTURES
FOR
GEOGRAPHICAL OBJECTS

by

Agus Prijadi Saido

TR92-064 January 1992

This is an unaltered version of the author's
M.Sc.(CS) Thesis

Faculty of Computer Science
University of New Brunswick
P.O. Box 4400
Fredericton, N.B. E3B 5A3

Phone: (506) 453-4566
Fax: (506) 453-3566

HIERARCHICAL DATA STRUCTURES
FOR
GEOGRAPHICAL OBJECTS

by

Agus Prijadi Saido
Ir. Surveying Engineering
Department of Surveying Engineering
ITB, Indonesia, 1980

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
Master of Science in Computer Science
in the
Faculty of Computer Science

This thesis is accepted.

Dean of Graduate Studies and Research

THE UNIVERSITY OF NEW BRUNSWICK
January, 1992

Abstract

Most of the current geographical information systems (GIS) are built for a single scale representation. They are developed on systems which do not usually capture the semantics and the inherent structure of geographical objects. The models are usually based on restricted data types, and the relationships among the objects are generally unnaturally represented.

This thesis introduces a hierarchical geographical structure called IMAS (Intelligent MAp Structure). It is based on the frame structure of expert systems and provides a unifying framework for multiple scale representation of geographical objects. A prototype of this structure has been implemented and tested with EMR map data at scales of 1:2,000,000 and 1:7,500,000. For this data, IMAS required approximately twice as much space as the standard EMR format. Substantial processing is required to correctly form all of the required IMAS relationships.

Contents

Appendices

List of Tables

List of Figures

Acknowledgements

In the name of Allah the most beneficient, the most merciful. Praise only to Allah, the God, the Lord of the universe.

My sincere thanks due to my supervisor Dr. B.G. Nickerson P.Eng., who introduced me to the application of AI for GIS; his advice, guidance, constructive criticism, and encouragement throughout the extent of the research is highly appreciated.

The financial support of the government of Indonesia, New Undergraduate Program Strata-1 in Computer Science (New–S1 IUC CS) The Ministry of Culture and Education in particular, has made this program possible. Special thanks are also addressed to the University of Sebelas Maret (UNS) in general, and Ir. M. Soeprapto MScEng. and Ir. Susy Soesmartini of the Faculty of Engineering UNS in particular, who gave me a great opportunity to pursue a graduate program in Computer Science in Canada.

Thanks to Mr. Kirby Ward – AI Laboratory of Faculty of Computer Science UNB, for help and discussions. Also, a great appreciation is given to Ir. R. Sarno,Ir. K.M. Seminar, Drs. A. Harjoko, Dra. S. Hartati, Drs. JS. Palgunadi,Ir. A. Satriawan, for initial help; Drs. T. Wibowo, Ir. W.Prastowo, Ir. ME. Priyanto, Drs. JP. Tamtomo, and Ir. W. Hendrajit for help in finishing this research.

My deepest feeling of gratitude to my parents Saido Soerjodihardjo and Siti Djuarijah, parents in law Anwari and Sunarti, all sisters and brothers: Itus, Sigit, Atik, Lilik, Jono, Juli, Dini, and Tomo, for constant moral and financial (in partial) support during my study in Canada.

Dedication

To my wife Sri Andayani and my son Dana Aditya:
"He it is Who created for you all that is in the earth."
"It is your Lord, the Most Bounteous,
Who taught by the pen, taught man that which he knew not .."
"We shall show them Our Signs on the
horizons and within themselves until it
will be manifest unto them that it is the Truth .."

Chapter 1

Introduction

The sophistication in computer technology and other disciplines such as telecommunications, photogrametry, and remote sensing, has had a great impact on Geographical Information System (GIS) applications. GIS is being used for scientific research and analysis of spatial data, involving heterogeneous data, and covering large areas [Smith 87,NCGIA 89].

Multiple data sources, a variety of applications, various internal representations, and the diverse needs of users, has revealed some inflexibility and inefficiencies in GISs. These problems are due to a lack of an underlying data model [Brodi 89b] or to an inappropriate approach applied to implement the present day GISs [Clark 86, Smith 87,NCGIA 89,Brodi 89b].

A new approach in programming called the Object Oriented Paradigm (OOP), provides a generalized data type and naturally represents the semantic and the inherent structure of complex data such as geographical data [Brodi 89b,Wonki 90]. A combination of OOP characteristics and expert system capabilities, seems the best solution for GIS's problems [Mohan 88,Tello 89,Smith 87,NCGIA 89].

This research explores the use of a frame-based approach to modelling geographical objects. The designed model must be capable of supporting a multiple scale representation of GIS data, and be flexible and easy to maintain.

1.1 Literature Review

Advanced research in GIS is a relatively new area, with most of it begun in or near the end of 1970's. The notion of integrated applications of artificial intelligence (AI) techniques into databases (DB) in general, and in GISs in particular, is not new, and only a few instances of research directly involved in this area can be found. Bruegger and Frank [Brueg 89], for example, proposed hierarchies over topological data structures to represent multiple resolutions of data suitable for multipurpose GISs.

Hierarchical structures are useful and often used to represent spatial data with different resolutions due to their flexibility to represent data at any level of detail, their conceptual clarity and their ease of implementation [Samet 84]. Topology is useful for network analysis and for 2-d spatial analysis.

Map generalization is still a difficult and expensive task, so that map automation is far from its full application [Brass 88,Nicke 88b]. Since multiple representations of geographical objects for different user levels at different locations is a must, a certain multiple scale (small, medium, large) of geographical object storage seems to be the only solution [Bossl 90,Brueg 89].

There are two possible solutions to the challenging issues on a large, complex database, with a large variety of applications and users. The first approach is by defining a **specialized data model** that captures the semantics of the applications. This is a rigid model, and does not adapt well to other applications. The second approach is by defining an **extensible data model.** The flexibility of this latter approach allows the new data types and or constructs to be added to the model required by the particular application involved [Brodi 89b].

Most of the more recent data models attempt to model more application-oriented relationships, and are generally object-oriented (OO). An OO approach is based on the concept of encapsulation and extensibility. OO programming encapsulates objects with data and codes (or procedures) to operate on the data. This method provides a way to hide both platform and format dependence. If the data moves from one platform to other platforms, or the system has to accept data with a different format, only the pertinent abstractions must be re-implemented. It assures fast prototyping and supports program evolution [Lisko 87,Wonki 90].

It provides at least four types of primitive object relationships :
classification *(instance-of)*, **generalization** *(is-a)*, **aggregation** *(part-of)*, and **cover** *(member-of)* [Brodi 89b]. These relationships, together with the data and procedures attached to the object can be inherited through an inheritance mechanism. The descendant may accept the generic or default value given by the ancestor, or alternatively over-write it. Since the object's behavior, relationships, and attributes can be inherited, the object model is flexible, and it may grow in conjunction with the application. Thus the modularity and extensibility of the data model is enhanced [Wonki 90]. OO concepts have also been used in frame-based KR languages, such as ART and KEE [Tello 89,Wonki 90].

1.1.1 Objects in GIS

A GIS can be viewed as e.g. [Smith 87], [Leeyc 87]:

> DB systems in which most of the data are spatially indexed, and upon which a set of procedures operates in order to answer queries about spatial entities in the DB [Smith 87].
>
> A GIS process involves a series of steps from collecting, analysis, and using the information gathered for a decision making process. GIS can be viewed as having the subsystem components: encoding and input processing, management, retrieval, manipulation, analysis, and display of spatial data [Leeyc 87].

GIS systems are characterized by the spatial and aspatial attributes of their objects. An aspatial element of geographical objects (geo-objects) is characterized by nonlocational attributes and relationships, while a spatial attribute gives locational attributes and topological relationships among the objects [Gupti 90]. Geo-objects are naturally organized as classes or sets of similar phenomena.

There are two basic alternatives available for the construction of a geographical data model; the *vector-based* and *raster-based* model. In a vector-based model, each object has spatial location as its essential property. The representation of vector-based models can be either unlinked or topologically linked. Conversely, in a raster-based model each location is characterized by a set of object properties.
Ideally, any GIS should be able to handle both vector and raster-based models. Topological vector-based models are good for geometric object operations, while raster-based models are superior for object analysis.

The relational data model has been used extensively and successfully for non-geometrical data. The absence of pointers and simplicity of the relational model has made it an attractive choice for GIS, and it has also been used for modelling geographical objects. The topological vector-based models, for example, naturally suggest the relational model. A relational model is not totally appropriate for modelling in a GIS. Many modifications on either the GIS relational model [Smith 87] or the standard query language (SQL) [Frank 82,Egenh 87] have been derived to handle the current problems in GIS applications. However, the extension of the relational model will not be able to support the core requirements of an OO paradigm [Wonki 90].

The recent emergence of AI and the OO approach has allowed these techniques to be employed in designing GIS data modeling. The US Geological Survey (USGS) Digital Line Graph Enhanced (DLG-E) Design and Spatial Archive and Interchange Format (SAIF), are the examples of a spatial database model which were designed using this approach. DLG-E is a feature-based data model for digital spatial data bases that represent geographic phenomenon [Gupti 90]. This model is an enhancement of the previous DLG model, wherein cartographic feature layers were built upon the topology. SAIF is a Geographical Object interchange format which was based an OO model. It describes geomatics data both structurally and semantically, and it defines a hierarchy of classes which identify geometry and attributes [SAIF 91]. The two parts of the SAIF data model are the structures, or objects, which make up the model; and the relationships and behaviors between the structures of the model and the operation which can be performed on the structures [SAIF 91].

1.1.2 Topological Structures

Topology is a branch of geometry which deals with a rubber sheet geometry that considers geometric properties invariant under continuous distortion. Within a planar graph, topological relationships can be viewed as boundary and co-boundary relationships among the graph elements. It characterizes the relative position among the objects, which describe neighbouring points as inside, leftside, rightside, or outside, relative to a specified object.

A planar graph is a graph which can be laid out on a planar surface, where any arcs on it cross only at the nodes. Since geographical objects are inherently layered in nature, line graphs are used to model a single theme representation on each layer of geographical objects [Smith 87,Leeyc 87]. The vertical relationships among geo-objects can be defined as an attribute relationships among different surfaces, and are used with various feature instances to indicate the relative vertical position of the surface.

A topological map is a map containing explicit topological information which defines the topological relationships among geographical objects in the map [Smith 87,Leeyc 87]. An example of a spatial database model which contains topological relationships is the USGS DLG-E. The advantages of topological information in GIS databases are that they enable the following to be done [Smith 87,Leeyc 87]:

1. Automatic consistency checking,
2. Efficient network analyses, and
3. Studies of optimum network configuration analyses.

Spatial analysis is made much simpler if we have knowledge about the topological relations of an object with its neighbours.

1.1.3 Multiple Representation of Geo-objects

Visual displays can be used to analyse as well as to illustrate information. They can be considered as tools for generating hypotheses as well as for interpreting the results of scientific research [NCGIA 89].

In the GIS context, the visual display of geographical objects is a major component and one of the most important tools in spatial analysis. The diversity of geo-objects, the different skills of the users involved, and the wide range of applications of GIS, would be well served by GIS models that allow multiple representation of geo-objects. It means, ideally, that multiple objects can be displayed in multiple scales and resolutions from a single GIS database.

Within a multiple representation environment, the systems have to provide the user(s) a way to acquire and update their data in a common effort, without the possibility of duplicate work [Brueg 89]. A GIS model must formally describe the objects at each resolution level, and provide the relations between them. Changes applied to one resolution should be able to propagate to the others, allowing objects in the other resolution levels to be deduced automatically [Leeyc 87,Brueg 89].

1.2 Frame Data Structures

In AI and Expert System applications, a language for representing knowledge must fulfill the criteria of expressiveness, understandability, and accessibility. It must have various components for knowledge representation to support rapid prototyping. Applied to a GIS environment, it must also have good interfaces to other systems, and application portability across platforms.

A frame language is mainly an elaboration of the semantic network. The emphasis is on the structure of types themselves called frames, and their attributes called slots.

A typical specification of the frame's slots are [Brach 85]:

> *values*, stating the value of the slot's instance; alternatively, it may be a default value, whereby in any case, an inherited individual does not override it.
>
> *restrictions*, stating *constraints* that must be satisfied by the attribute's value. These restrictions can be value restrictions, specifying the type of the slot's instance, or number restrictions determining the minimum and maximum number of the slot's value.
>
> attached *procedures*, providing a procedural advice, on how to use the attributes. An *if-needed* procedure is intended to calculate attribute values if none have been specified, and an *if-added* procedure determined the action to be taken when a new value is supplied.

The information stored in the frames, together with facts and rules is known as the knowledge base. Presently, the best-known commercial examples of frame shells are ART, KEE, and KnowledgeCraft [Boley 90].

Minsky's primary motivation when introducing frames [Minsk 75] was to motivate semantically the reasoning of scene-analysis systems. Most of the subsequent implementations are focused on structural representation issues rather than on the control of reasoning [Fikes 85].

A frame structure is more structured compared to other representations. It provides a structured representation of an object or class of objects, and also has a construct for representing frame taxonomy or frame conceptual hierarchies. These constructs allow a knowledge engineer to describe each class as a specialization (subclass) of other more generic classes. Subclasses may inherit their parent information, modify and or add their own behavior. In general, parent classes represent metadata which hold the default knowledge about its class. The object itself is represented in the class instances.

Allowing a superclass on top of a class hierarchy means that all intended subclasses of the family are made a subclass of the superclass. This approach is known as **polymorphism**, a data abstraction that works for many different types. One abstraction is related to another abstraction hierarchically. This implies a flexibility in representing the complex relationships among objects. Objects can be organized and represented in any level of detail [Lisko 87].

Figure 1.1 shows the components of frames compared to objects. Figure 1.2 depicts an object computer hierarchy through inheritance, and its corresponding frames are shown in Figure 1.3.

Frame ≺ – – – – – – – ≻ Object
Class ≺ – – – – – – – ≻ Class
Slot ≺ – – – – – – – ≻ Attribute
Value ≺ – – – – – – – ≻ Value

Figure 1.1: Frames compared to objects.

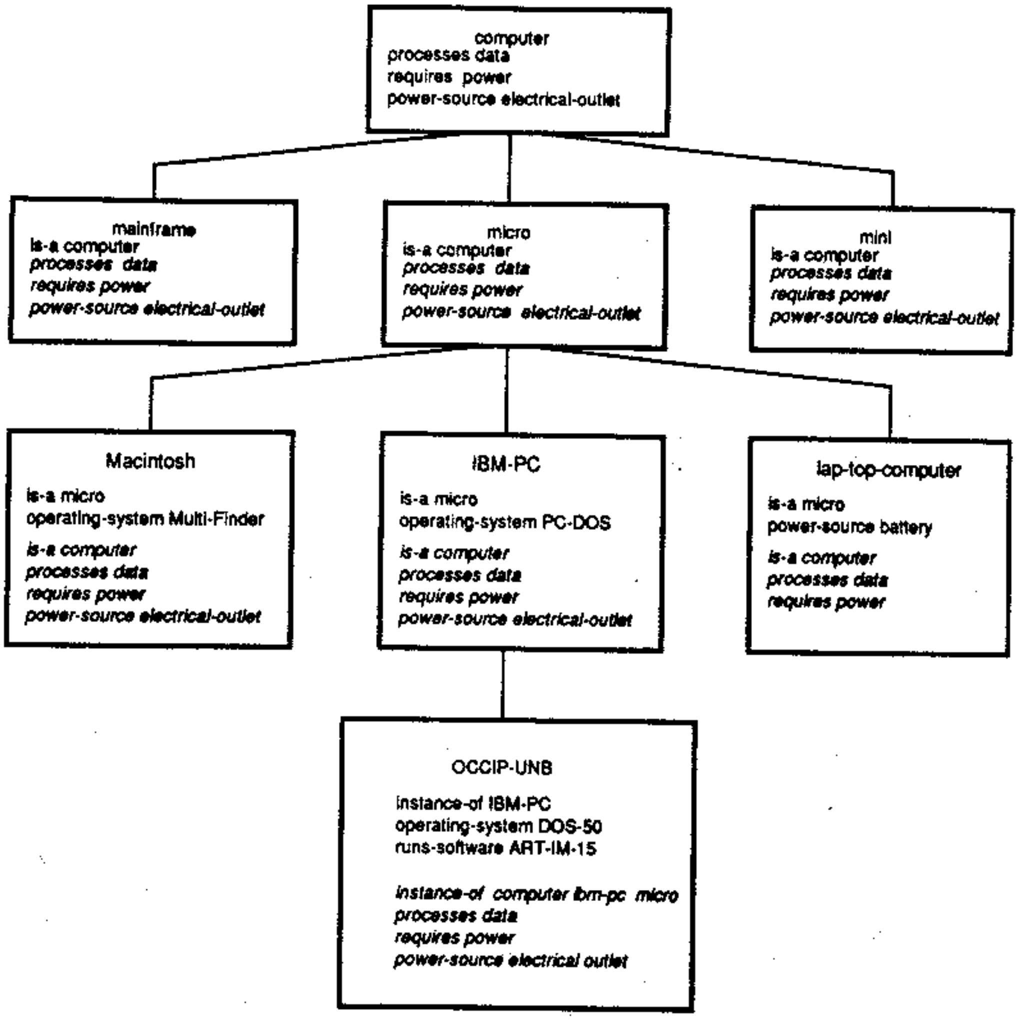


Figure 1.2: A computer hierarchy [Infer 89a, page 3-13]: slanted text is an inheritated value. OCCIP-UNB is an *instance-of* IBM-PC, its operating system is DOS 5.0, and it runs software ART-IM 1.5.

```
(defschema computer
  (processes data)
  (requires power)
  (power-source electrical-outlet))

(defschema micro
  (is-a computer))

(defschema IBM-PC
  (is-a micro)
  (operating-system PC-DOS))

(defschema OCCIP-UNB
  (instance-of IBM-PC)
  (operating-system DOS-50)
  (runs-software ART-IM))
```

Figure 1.3: Frame representation of the computer hierarchy shown in Figure 1.2.

1.3 Problem Definition

Besides the problem of the type restrictions, and the inflexible and unnatural structure in some current GIS models, there are also difficulties in working with large scale differences and wide coverage areas [NCGIA 89,Smith 87,Tomli 87,Leeyc 87]. A large data volume, combined with the heterogeneity of the data involved sometimes causes problems with data integration. Systems are required to be able to accommodate a variety of applications and users.

A new approach to modelling geographical objects is needed which can provide both flexibility and modularity. The model should be capable of managing and providing the services required for efficient and consistent retrieval, update, display, and modification of graphical objects at large scale ranges and over a wide coverage area.

The way we perceive geographical objects naturally is (e.g. from a map): (1) identify its classification, (2) determine its shape and size, and (3) find its relative position among neighbouring objects. The first two observations can be independently or jointly applied in describing and interpreting geographical objects, which largely depends on the nature of the objects and the scale of the representation. These abstractions suggest the notion of the integration of geographical object descriptions and all their behavior into a single data model. The frame based representation is a promising model which allows flexibility and modularity, and provides a means to perform efficient geographical operations.

This research concentrates on geographical object modeling, and its structural implementation using a frame based approach. Frame based models naturally represent the inherent structure of spatial objects, and can consistently accommodate geographical object management in a large scale range and a wide coverage area, as well as providing for extensibility, and ease of modification.

1.4 The Approach

Two conceptual models proposed by Lee [Leeyc 87] and by the USGS [Gupti 90] are suitable for a single resolution of geographical object representation. In these models, both the spatial and aspatial attributes of geographical objects are well structured. However, there are no concepts provided to allow relating different resolution representations.

In a multiple scale representation, generalized objects at smaller scale must be connected to more detailed representations at a larger scale. A small-scale object coexists with a larger scale representation of the object. Intuitively, object(s) could be accessed directly in different degrees of detail. Object class hierarchies must be provided to maintain the object's structure at different resolutions.

As an analog to a single representation model where the relationships among the theme layers are maintained, in a multiple representation there must be a way to maintain the **resolution layer**. A single layer resolution representing all topological relationships in one resolution has to be connected to other layer resolutions.
The advantage of storing object descriptions, preprocessed relations, and other contextual behavior explicitly in the data model, is for efficiency in retrieval of and topological operations on the objects. Much of this information is computationally difficult and expensive to obtain.

An object-oriented approach, which is embedded in a frame-based representation is used here for modeling geographical object representations. A frame-based representation has the benefits of both knowledge and object-oriented approaches. Details of its superiority in modeling the structure of geographical object representations will be discussed in the subsequent chapters of the thesis.

Chapter 2

A Model for Geographical Objects

The most appropriate representation of geographical objects is obtained by considering them as complete object units. This allows the representation of differing object types along with their unique characteristics and relationships to other objects.

A frame-based model of knowledge representation called a *schema* [Infer 89a], facilitates reasoning and provides a means to represent the structural knowledge and spatial properties of objects at various levels and for different views [Mohan 88].

This chapter provides the conceptual design of frame-based geo-object data structures. Initially, the characteristics of geo-objects – in an OO dialect, and the kinds of operations required in a GIS environment must be defined. These findings are then use to define the frame-based model.

A vector data model with complete topological relations is used. Extension of the model to include raster data is not considered here, but could easily be incorporated.

2.1 Requirements of the Geo-Object Model

Requirements of the geo-object model, in general, can be derived from the tasks carried out by the system using these objects [Smith 87]. Typically, research in system design, databases, and knowledge-based systems related to GIS is concentrated on how to manage a large number of complex geo-objects naturally and efficiently. Some specific examples include, Roman [Roman 90], on the qualification or validation of location, time, and accuracy of the spatial objects; Egenhofer [Egenh 90], who concentrates on the graphical representation, user interface, and spatial query; and Bruegger [Brueg 89], who investigates a hierarchy of topological structures to meet the requirements of a multipurpose GIS.

By considering GIS as a technology for scientific analysis and research in sciences dealing with spatially distributed phenomena [Smith 87,NCGIA 89], the future environment of GIS will be an intelligent and cooperative environment. It must be able to handle large and complex objects with a multi-view representation. It will be used by many different users who are at various geographical locations. A general requirements list for design and implementation of a GIS is that they should be able to:

1. Handle heterogeneous, large, and complex spatial data.
2. Provide queries on the spatial data consistently and efficiently.
3. Represent the data in different views.
4. Support a variety of applications and users.
5. Monitor how the data is used by each user.
6. Be integrated easily with other software systems, including knowledge based systems.

Intuitively, the frame-based model of knowledge based systems seems to be capable of being adapted to meet these requirements.

2.2 Design of the Geo-object Model

Geographical objects represent hierarchical real-world phenomena. They also resemble an object clustering at various levels and degrees. Geo-objects are connected horizontally or vertically, and they have relations with other geo-objects commonly represented as different layers. There exist nested complex objects, where one geo-object contains other geo-objects of the same or different type. As an illustration of the complex relationships among geo-objects, consider the following example: New Brunswick is one of Canada's provinces, and it has transportation networks comprising railways and highways. The railway is operated by several companies, and the highways can be classified as national, primary, connector, or local. The Trans Canada highway no. 2 (NB2), is a national highway as well as part of NB transportation network. It crosses the St. John River in Fredericton.

If a model of geo-objects is well designed, it captures the semantics and represents the inherent structure of these geo-objects. The resulting implementation must simplify both maintenance and modification of the model requirements. From the geo-object illustration given above, it seems that a conventional database model (most of them are based on a relational model), is inappropriate and insufficient to represent the requirements of geo-object representation. Besides lacking the necessary data types, operations and queries over relational tables of large, complex objects are inefficient and may give inconsistent results [Clark 86,Smith 87,NCGIA 89,Brodi 89b, Wonki 90].

A frame-based model can meet the requirements of geo-object representation. In a framed-based environment, a model of geo-objects encompasses attributes and relationships which are the properties of the geo-object, and procedures to operate on geo-objects. As in the OO paradigm, it encapsulates both data and programs to operate on data. Compared to existing standard definitions, for example the DLG-E of USGS [Gupti 90], or DIGEST [DGIWG 90], this description reflects a more complete model, since it has its own methods about how to configure itself, and on how to interact with its environment.

Based on an extensive use of the OO paradigm [Tello 89,Wonki 90], frames are a data structure that can represent all information of a particular object in its slots. Hierarchical structuring through an inheritance mechanism is a logical and convenient way to organize classes of geo-objects. Many consequences of class hierarchy can be imposed on a conventional database system, e.g. the query model [Wonki 90].

2.2.1 Geo-object Hierarchy

Hierarchical data structures are commonly used to represent spatial data. Hierarchical structuring through **IS-A** inheritance has two benefits; it is a good structure to organize geo-objects, and it is a useful tool for program design and development.

The IS-A inheritance represents a class, subclass or metaclass hierarchy. A class is a generalization (superclass) of its subclass (specialization). If the class/subclass relationships can be well-defined in advance, then a group of related types can be defined in any level. In some circumstances, the top superclass is just a place holder for its descendant's family. A combination of locality provided by object encapsulation and polymorphism through inheritance enhances frames as a prominent choice for representing geo-objects. In fact, many geo-objects are polymorphic. Figure 2.1 shows a hierarchy of political division. It also shows a grouping of geo-objects based on their political division level.

Geo-object modularity is obtained from the geo-object's capability to inherit at any level of the representation. The model can be designed incrementally using this modularity. From the same illustration in Figure 2.1, a new or finer object classification can be added, again, at any level, by attaching the object to the ancestor where it belongs, and making a parental connection to the object's sub-class, if necessary. For example, a new province of the country can be added and included to the set class of the province, and subsequently the sub-classes counties and cities which belongs to

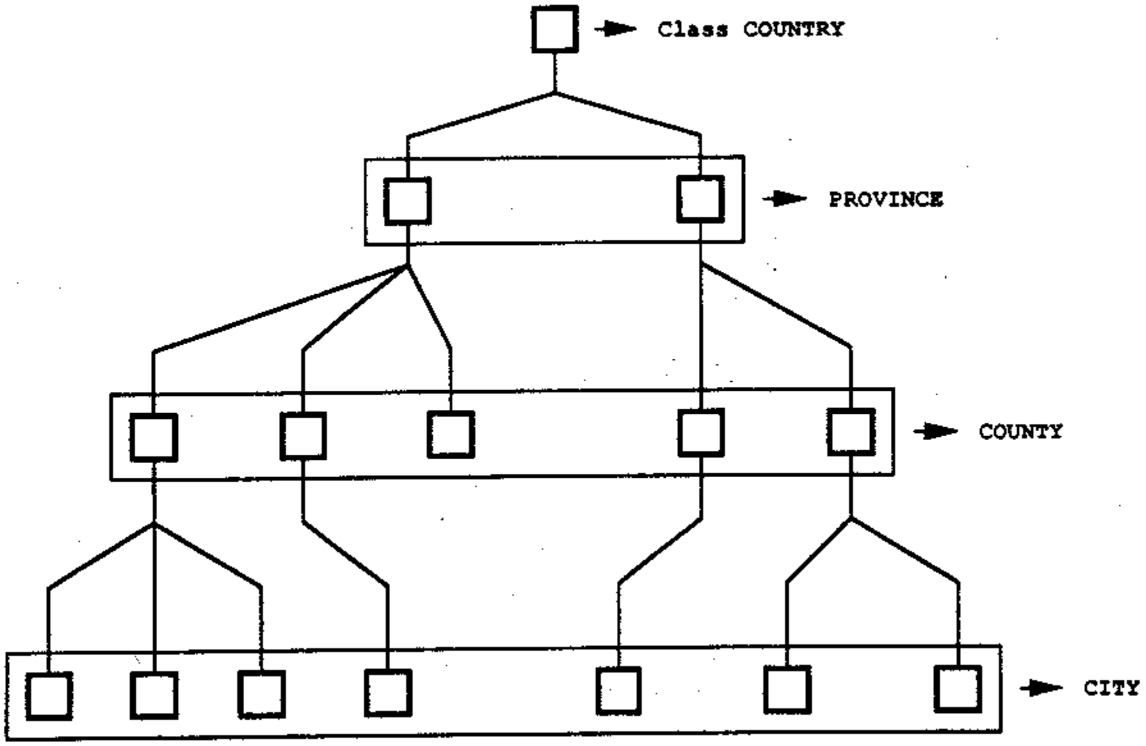


Figure 2.1: A containment hierarchy for political sub-division.

the province are also included to the sub-hierarchies of this new province hierarchy.

Another benefit of inheritance is providing for the maintenance of *integrity constraints* in the geo-object's attributes. This can be achieved by adding methods to each object in the class hierarchy. Since methods are also inherited, they will always make a self-integrity check whenever an inheritance is attached to a particular object in the hierarchy. For example, a method could check that for every plane triangle object the sum of its inner angles equals 180 degree.

In addition to IS-A inheritance, the **INSTANCE-OF** relation can be used to construct the leaf level of an inheritance hierarchy. This last constructor indicates that no other object can be inherited from the object that is an instance-of another. If IS-A objects represent the set or subset of the object class, then INSTANCE-OF objects represent specific objects of this class type.

Geo-objects can inherit attributes from two or more different classes. This is called a *multiple inheritance.* A complete illustration of a logical organization of geo-objects at a particular scale resolution is illustrated in Figure 2.2. As will be seen later, the resolution hierarchy must be built to support multiple object representations consistently and correctly.

2.2.2 Attributes

Attributes are properties that characterize the object. The domain attribute values of geo-objects are : (1) the primitive data type (e.g. integer or real), (2) the object type, and (3) methods attached to the object. The second attribute allows a nested object definition. The third attribute points to procedures that act on the object. Attributes appropriately represent the inherent structures and semantics of geo-objects, and promote the aggregation hierarchy of classes [Wonki 90].

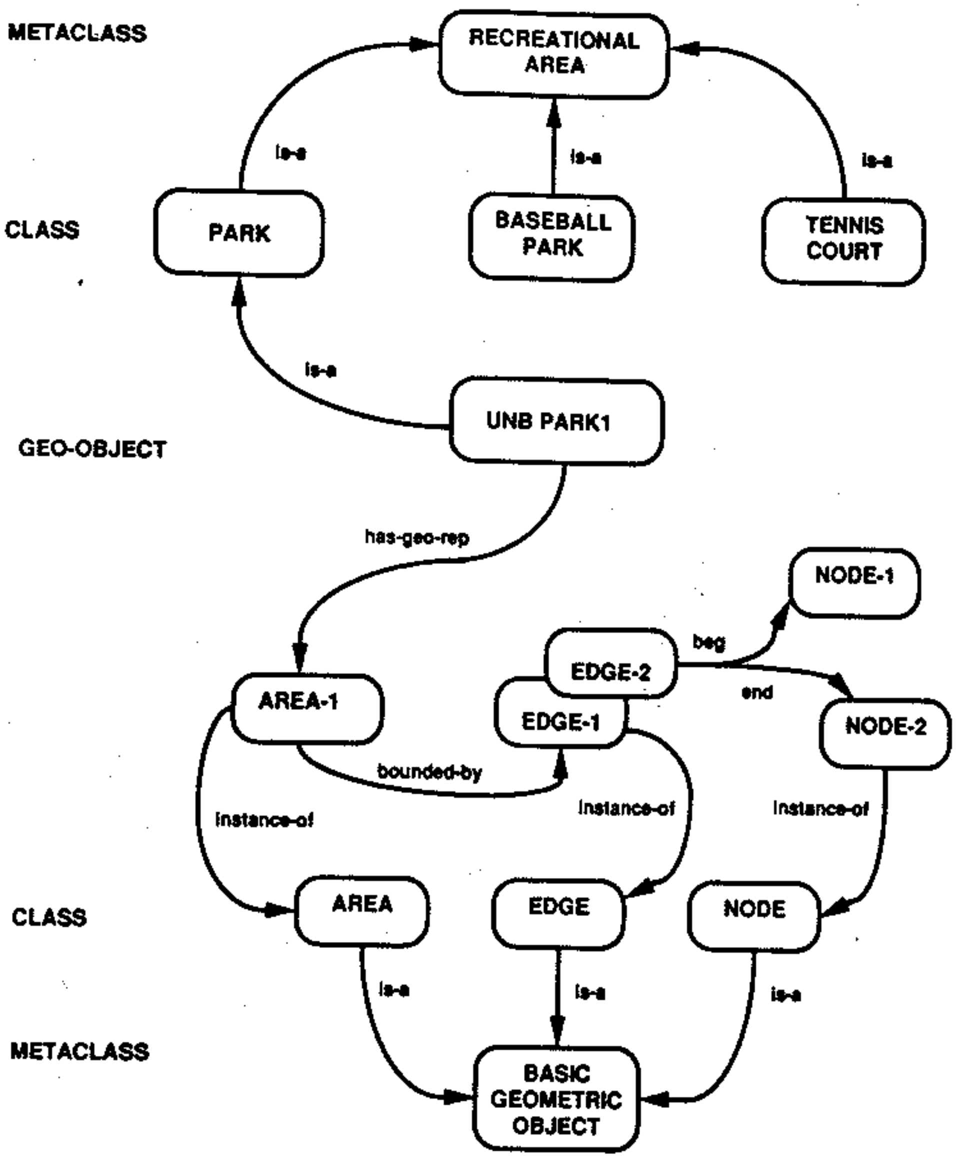


Figure 2.2: Logical organization of a geo-object representation.

The domain attributes of geo-objects are also classified as either locational (the position and geometric characteristics of the object) and non-locational attributes that characterize the object. In the proposed model, the topological relationships of geometric attributes can be assigned to any level of detail.

2.2.3 Relationships

The notion of relationships in the geo-object model is extended to include methods. The richness of the attribute values in the geo-object model indicates that object queries are richer than in the relational model. Since the class properties are inherited, the model supports the relationship hierarchy at various levels of abstraction. This provides more direct connections between geo-object levels.

Within the geo-object classes, the relationships cover topological and nontopological links between geo-objects. More object oriented relationships can be established than in the currently known relational model. Consequently, more geo-related analysis can be performed. Some examples are : (1) *composed-of/part-of* which indicates the object formation, (2) *bounded-by/bounds* describes the boundary between contiguous objects, and (3) *within/contains* depicts object containment.

2.3 Representation of Geographical Objects

Currently, geographical objects may have many forms of representation. This section provides a case example of a representation in vector form. A traditional map scale definition will be used to represent the resolution of geo-objects.

Based on the topological definitions in section 1.1.2., a model of topological objects that conforms to the model outlined in sections 2.1 and 2.2 must be defined. A more detailed description for multi-scale applications is given subsequently in a single scale and a multiple scale definition.

2.3.1 Topological Model

The topological objects used to represent the locational component of the proposed model are: (1) **basic-object** *(node, edge, area)*, and (2) **complex-object** *(junction, chain, polyplex).* This definition is extensible for application to 3D objects, since 3D objects are constructible from their 2D elements. A basic object is a simple object that is commonly formed in 2D space. Complex objects are obtained by a combination of basic objects, basic objects and complex objects, or other complex objects.

A node is characterized by its position, for example (x,y). It may uniquely represent a point, a beginning and or an ending of an edge. An edge is an ordered set of points. It may represent a straight line or an arc beginning and ending at different nodes. An area is an object delimited by at least one edge, beginning and ending at the same node.

A junction is a complex object of node-like form. A chain is a line-like form complex object. Its beginning and ending depend on the objects forming the chain. A polyplex is an area-like complex object. An example of this type of object is an area comprising one area enclosed by a chain of areas. Figure 2.3 illustrates the above-mentioned topological object definitions. Since an attribute of a topo-object may have multiple values of many different types of topo-objects, then it is very easy to see that many geographical objects may share one object representation.

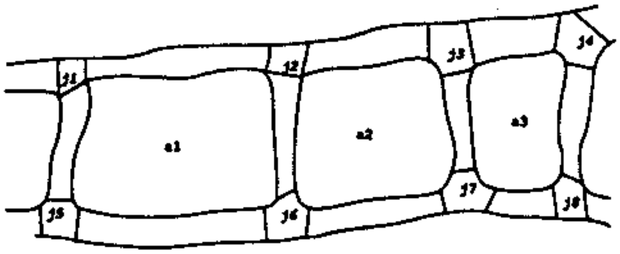
2.3.2 Single Scale Representation

Thematic maps in conventional map making are often represented in a layered model. Geo-objects of different themes overlay each other. Connections must be made at the common location where objects are crossing other objects. The proposed geo-object model is based on a full topological model for the following reasons: (1) In a multi-representation environment, all objects are candidates for either super or sub-representation, and (2) Analysis of objects in the multi-representation model requires

(i) node. (ii) edge. (iii) area.

(a) Basic geometric objects.

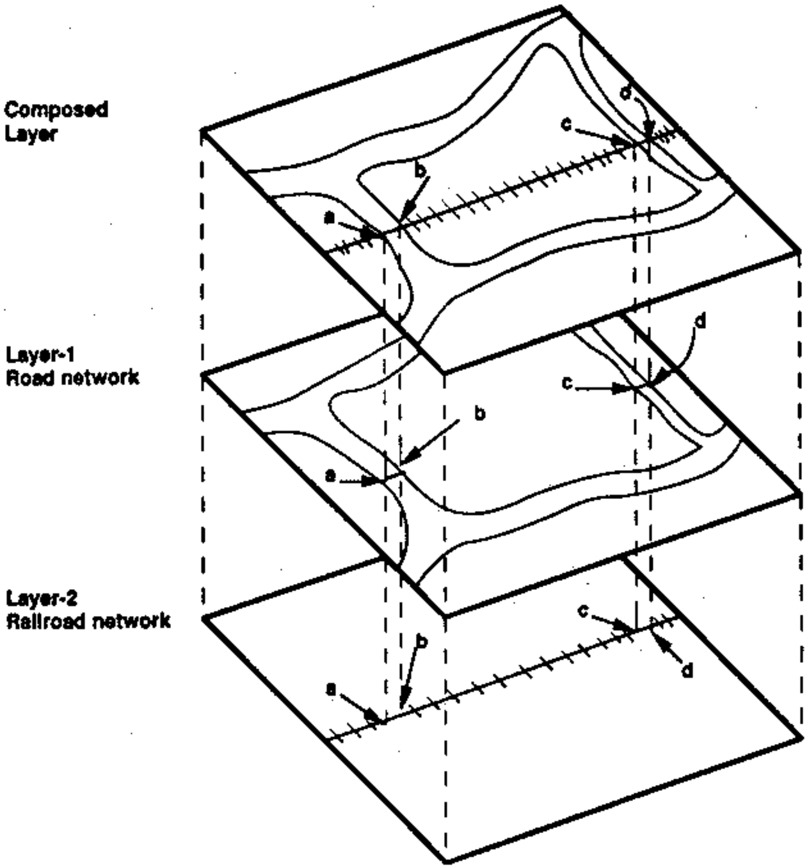
(b) Chain object



(c) Polyplex: ai is an area and ji is a junction.

Figure 2.3: Example of topological components of geo-objects.

access to all the topological relationships between geo-objects. Figure 2.4 shows an example of the layered model for the single scale representation of geo-objects.

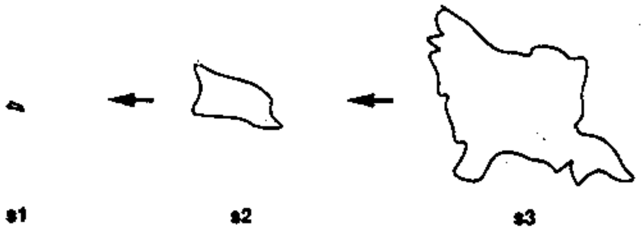


a, b, c, and d are nodes, a common point between objects.

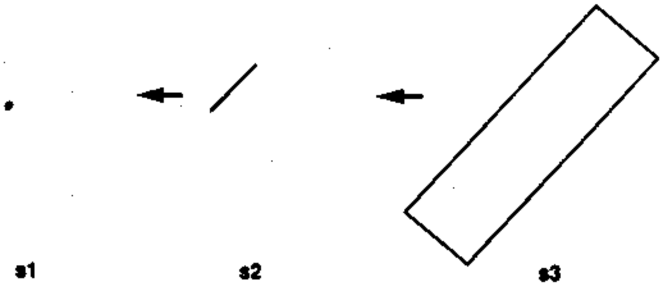
Figure 2.4: A layered model for a single scale representation.

2.3.3 Multiple Scale Representation

The same geo-object in different resolutions might have a different structure of representation. At a certain resolution it may even cease to exist. This is due to the principle of cartographic generalization, where objects at a lower resolution are the product of object simplification from objects of higher resolution. Some objects can be ignored depending on their importance. The objects are represented differently at different scales, but still represent the same physical object. The illustration in Figure 2.5 shows how these objects are different, both in structure and complexity.



(a) Geo-object island represented as an area,
s1 < s2 < s3.



(b) Geo-object bridge represented as an area in S3,
is represented an edge and a point in smaller
scales s2 and s1.

Figure 2.5: An object with different scale representations s1, s2, and s3. At the smaller scales objects are simplified.

By using control inheritance and resolution levels, in this case the map scale, a plane of resolution hierarchy can be arranged such that objects at adjacent upper and lower resolutions are connected. The illustration in Figure 2.6 shows the layers of a multiple plane resolution. Each plane is representing a layered model of a single representation as shown in Figure 2.4. The bottom level 0 indicates an object representation at its highest resolution or most detailed object description. Conversely, the top level 3 indicates an object representation at its lowest resolution or most simple object description. In between these two levels, one object might have been simplified into its simplest representation, or alternatively discarded from the representation.

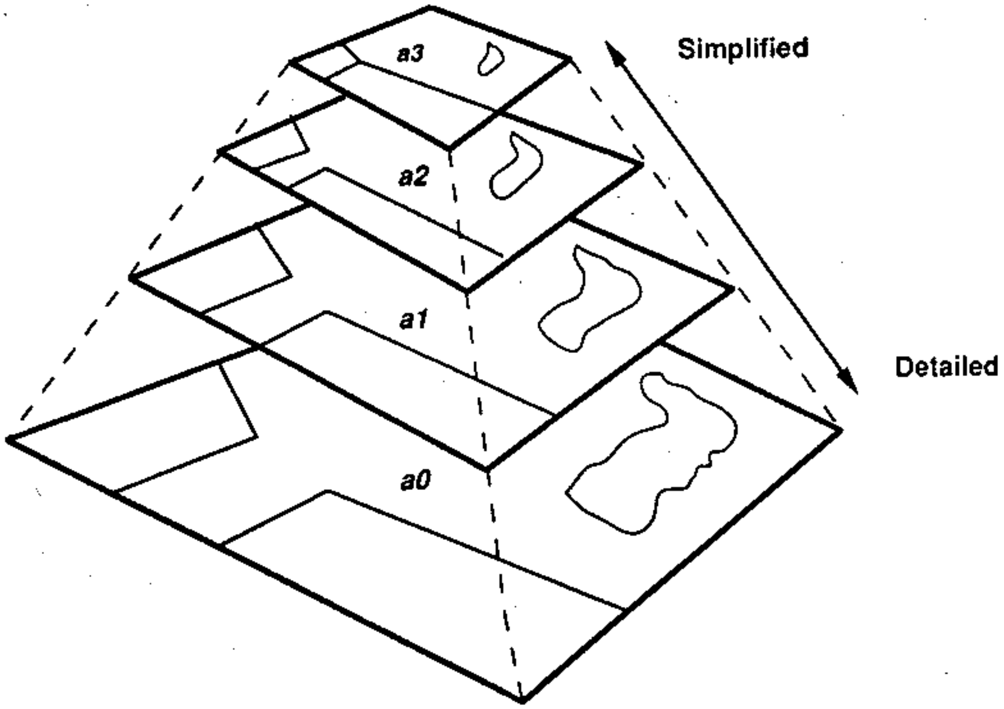


Figure 2.6: A layered model of multiple plane resolutions.

Links can be assigned between two objects on adjacent planes at any resolution level, if the objects exist at the two levels.

2.4 Frame Model for Geographical Objects

In the preceding sections, a geo-object model has been defined to meet the given requirements. A geo-object frame structure is now added to represent the model.

Two main considerations are involved in constructing a frame structure for geo-objects. First, is the classification of geo-objects. Second, specific for a vector representation, is the construction of topological links among different representations.

2.4.1 Multiple View of Geo-objects

There are two ways in which an inheritance hierarchy can be used to organize the domain of geo-objects. It allows the grouping of geo-objects of related types into a metaclass hierarchy, and it provides a convenient way of organizing a library of geo-object classes. A combination of these two aids, enhanced by an efficient organization of geo-object classes, is called a view [Gupti 90]:

> A view is a systematic classification of a set of entities in which all members of the set possess a common defining characteristic.

Both the view domain definition and the geo-object classification process is beyond the scope of this thesis. All of its structure in the next 3 chapters is adapted from a USGS DLG-E domain definition [Gupti 90].

Users can expand their geo-object views by just once implementing *GeoView* as a place holder of a superclass. Subsequently, subclasses can be implemented independently.

2.4.2 Hierarchy Over Topological Relations

The two types of topological objects, basic-object and complex-object must be placed into two different frames. Each of these two components has its own superclass representation. This guarantees their flexibility for supporting any spatial application and future extension of their class.

As part of the geo-object attributes, topological relations have two directions in their construction. The first is in its own resolution plane representation where the topological relationships are defined, and the second is hierarchically between two adjacent planes of resolution. Thus, the topological hierarchy is represented by the hierarchy of planes of resolution. The illustration in Figure 2.7 shows how this topological hierarchy is constructed and is placed in the frame of geo-objects.

2.4.3 Frame Model of Multi-geo-objects

The frame components of multi-geo-objects, that is, a geographical object with a multi-representation of format and resolution, are listed in Table 2.1.

Table 2.1: The frame components of multi-geo-objects.

GeographicalObject-X
  - Classification of GeoObject-X.
  - Non locational attributes.
  - Link to SpatialObject-X.
  - Link to TopologicalHierarchy.

SpatialObject-X
  - Spatial object classification.
  - Link to spatial object constructor, if any.
  - Spatial methods, if specified.
  - Positional attributes, if specified.

These frame structures can also be nested. In the case of geo-objects, the nested element is another geo-object, while for spatial objects the nested element is another finer spatial object definition.

Since locational attributes and topological relations are only valid for each representation scale, a generic object is made for each recognized geo-object. This generic object carries, in its frame, an object classification and non-locational attributes that are generally valid for all representations. Consequently, there will be many geo-object instances for all representations. Every object instance has its own locational attributes and its link in the topological hierarchy. Thus, a name holder for locational attributes is an object name for topological relations on the plane of topological relations. An extension for any new geo-object definition is made by attaching its frame to the frame of the multi-geo-object environment.

In addition to those frames inside the multi-geo-object representation, frames of geographical objects can be created independently. Links to connect them to the multi-geo-object frames that already exist can be added later.

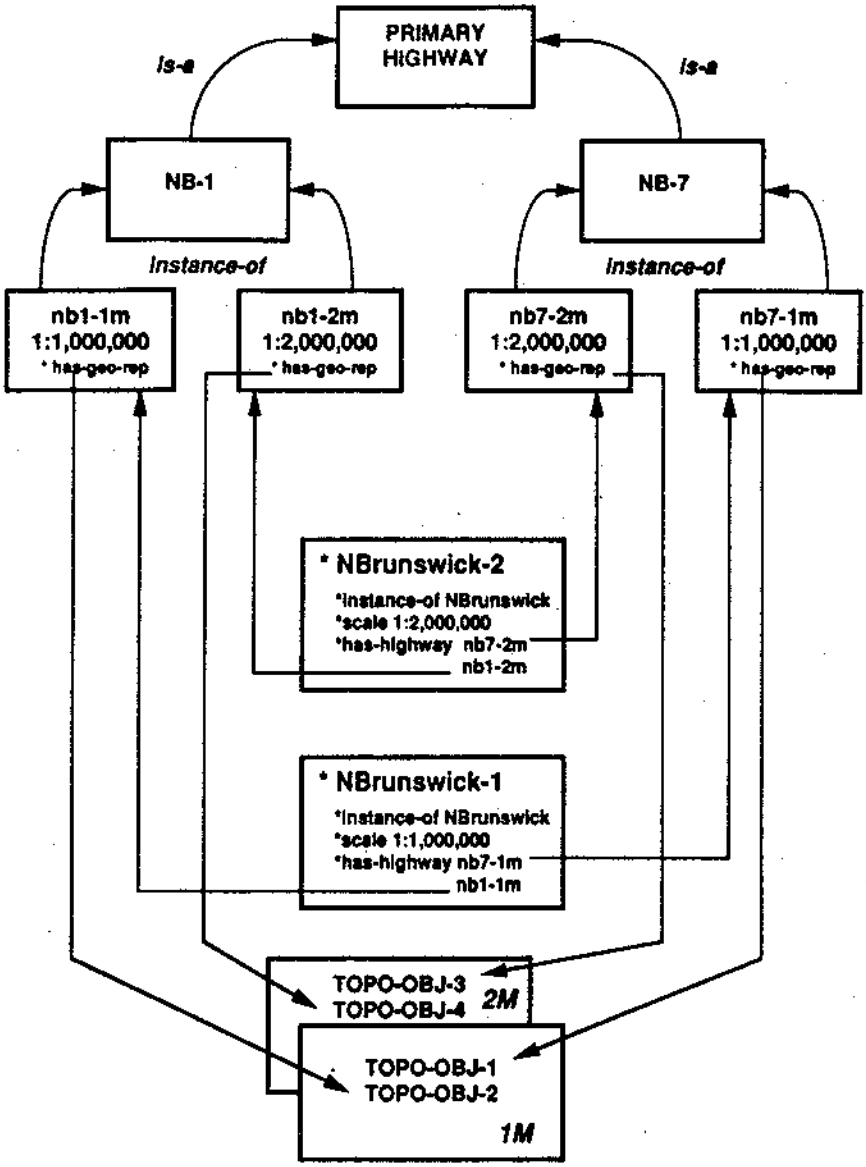


Figure 2.7: An example of frames of a multi-geo-object representation.

Chapter 3

Frame Structure of Geographical Objects

This chapter discusses the implementation of the geo-object model described in the previous chapter. In the ART-IM/MS-DOS expert system shell, frames are called schemas [Infer 89a]. By definition, a schema is a data structure that represents a collection of information about some specific object.

There are two basic schemas that implement the geo-object representation; a schema for a topological object and a schema for the geo-object itself. An occurrence of geo-objects which is independent of its representation is called a generic geo-object. In fact, a topological representation is linked to instances of this generic geo-object. By using a schema representation, geo-objects can be linked in arbitrarily complex semantic nets, representing their taxonomic or conceptual hierarchies. Through inheritance, information is inherited from more general objects to the more specific objects.

The first section of this chapter discusses a general schema structure, and the rest contains detailed discussions of the schema topological object, geo-objects and their classification, and the frame structure of geo-object representation.

3.1 An Expert System Shell Frame Structure

A collection of information about some specific object in the schema comprises the object's data, methods, and relationships to other objects. All of this information, individually, is put into slots of the schema. A *slot* [Infer 89a], is a data structure used to represent some characteristic about an object.

A schema can exist in the knowledge base (1) by defining it using the **defschema** statement, (2) asserting it with rules using the **assert** action, (3) procedurally by using the **schemac** function, or (4) implicitly created by other schemas via slot relations. Figure 3.1 shows a general ART-IM schema structure; a more detailed explanation and syntax are provided in Appendix I.

(*schema-object-name*
    (*relation-slot-name-i* {*value*} *)
    (*slot-name-i* {*value*} *)
)
{*value*} * argument may appear zero or more times.

Figure 3.1: Schema for an ART-IM implementation of frames.

A schema-object-name can either be an object, a class-object, or a relation object. A relation-slot-name-i is a *relation slot*, representing an object relationship. Its slot value is the other schema-name to which it is related. Conversely, a slot-name-i is an *attribute slot*, representing a characteristic or an attribute of the object. Its slot value is not an object, even though it might be a symbol for an object in the knowledge base. A *value* is any legal **art-object**, e.g. *symbols*, *strings*, *numbers*, *sequences*, *streams*.

In conjunction with OOP, an attribute slot may have a value that represents a method or a procedure that characterizes the object. Thus, instead of representing the declarative nature of the object, a slot of this type represents the procedural nature of the object.

3.2 Frame Structure of Topological Objects

The following frame structure of topological objects is mainly concerned with geometrical representations and topological relationships in their simplest form to support multi-representation of geo-objects. By using a frame representation, it is easy to expand the domain of topological objects along with other new object classifications.

On top of the topological object classification is a schema of *topo-object* as the place holder for the class of basic and compound topological object classifications.

Due to the notion of multi-view geo-objects, the proposed structure provides some redundant relationships. In the form of a slot's inverse relations, these redundant relations may have one to many, many to one, or many to many relationships, depending on the specialization and application levels.

The first level of topo-object classification is shown in Figure 3.2. Its slot relationships are listed in Table 3.1. Complete slot descriptions are outlined in Appendix B.

(schema TOPO-OBJECT
(topo-structure un-defined))

(schema BASIC-GEO-OBJECT
(is-a topo-object)
(*topo-structure* **simple**)
(dimension un-defined))

(schema COMPLEX-GEO-OBJECT
(is-a topo-object)
(*topo-structure* **complex**)
(object-function un-defined))

Italic font indicates an inherited argument.
Bold font indicates an *override* value.

Figure 3.2: Topological object classification.

Table 3.1: Relationships between topological objects

Object	Relationship	Object
topo object	-- geo-rep-of -- ≻ ≺ -- has-geo-rep --	geo-object
topo object	-- part-of -- ≻ ≺ -- constructed-by --	topo object
topo object	-- obj-within -- ≻ ≺ -- consists-of --	topo object
topo object	-- vertically-related -- ≻	topo object
node	≺ -- beg/end-node -- -- beg/end-node-of -- ≻	edge and area
junction	≺ -- beg/end-object -- -- beg/end-object-of -- ≻	complex topo object
edge-like	-- right/left-area -- ≻	area
edge-like	-- right/left-polyplex -- ≻	polyplex
chain, polyplex	-- bounded-by -- ≻	chain, polyplex
chain, polyplex	-- bounding-area -- ≻	area

3.2.1 Basic Topological Objects

In a planar graph representation, basic topological object are nodes, edges, and areas. Nested objects are only definable for area objects. A node, an edge, and or an area within an area are possible. An arbitrary combination of these basic objects performs a complex or composite topological object. Conversely, complex objects are decomposeable into other complex objects or basic objects. Figure 3.3 illustrates the schema structure of a node, an edge, and an area.

The main slots of the basic topological object frame structure can be categorized as a:

1. Link(s) to the represented geo-object(s).
2. Link(s) to other higher dimension topological object(s).
3. Directional attributes of the object.
4. Locational attributes of the object (except for an area object).
5. Geometrical attributes of the object.
6. Procedures or methods related to the object construction and composition.

A *geo-rep-of* is a relation slot type (1). It connects to one or more geo-object instances. *part-of* is a multivalued relational slot of type (2). The *beg-node* or *end-node* slot indicates the object direction. The *xy-value* slot describes the locational value of the objects.

The above mentioned slots for structuring basic topological objects are also applicable for structuring frames of complex topological objects (see section 3.2.2). The interesting features of this frame structuring are the modularity and the granularity of the model. Objects at different views and levels of resolution can be linked naturally through a frame's slots.

(schema NODE
(is-a **basic-geo-object** *topo-object*)
(*dimension* **0**)
(geo-rep-of)
(part-of)
(beg-node-of)
(end-node-of)
(x-value)
(y-value)
(related-methods))

(schema EDGE
(is-a basic-geo-object)
(*dimension* **1**)
(geo-rep-of)
(part-of)
(beg-node)
(end-node)
(left-area)
(right-area)
(bound-box)
(x-value)
(y-value)
(related-methods))

(schema AREA
(is-a basic-geo-object)
(*dimension* **2**)
(geo-rep-of)
(part-of)
(beg-node)
(end-node)
(constructed-by)
(consists-of)
(neighbor-area)
(bound-box)
(related-methods))

Bold is an override value.
Italic is an inherited value.

Figure 3.3: Frames for basic topological objects.

3.2.2 Complex Topological Objects

In the representation of topological objects, the notion of a complex or composite object is slightly different from that commonly used in OOP. There is only one slot attribute called *topo-function* which defines the functional attributes of the complex topological object. These attributes depict the equivalent characterization of the node, edge, and area of basic topological objects (see the complex topological description in section 2.3.1). Other slots contain the object instance at a particular resolution or scale.

The nature of complex topological objects is the nature of geographical object representations, where any geo-object may contain other objects of the same or different classes and dimensionality. The need for complex topological object representation also comes from the requirements of spatial analysis applications. A complex topological object represents the composition of a group of interconnected basic, and/or complex topological objects.

From a topological point of view, these compositions depend on the resolution or scale of representation. From a spatial applications point of view, the coverage and its spatial unit definition define the class of the represented complex topological object. For example, a city can be defined as a junction in small scale analysis, but it will be considered as a chain or a polyplex object in large scale analysis.

Figure 3.4 shows a schema structure for complex topological objects of type junction, chain, and polyplex. The *constructed-by* slot is a multivalued or a non-inherited relation slot. These structures are similar to the previous structures of basic topological objects. The difference is in the object constructor which may be nested, or defined recursively. Consequently, they do not carry the actual locational values of the object, but are concerned only with the topology.

(schema JUNCTION
(is-a **complex-object** *topo-object*)
(*topo-function* **topo-node**)
(geo-rep-of)
(beg-object-of)
(end-object-of)
(constructed-by)
(related-methods))

(schema CHAIN
(is-a **complex-object** *topo-object*)
(*topo-function* **topo-edge**)
(geo-rep-of)
(part-of)
(beg-object)
(end-object)
(constructed-by)
(consists-of)
(left-polyplex)
(right-polyplex)
(bound-box)
(related-methods))

(schema POLYPLEX
(is-a **complex-object** *topo-object*)
(*topo-function* **topo-area**)
(geo-rep-of)
(part-of)
(beg-object)
(end-object)
(constructed-by)
(consists-of)
(neighbor-polyplex)
(bound-box)
(related-methods))

Bold is an override value.
Italic is an inherited value.

Figure 3.4: Frames for complex topological objects.

3.3 Frame Hierarchical Structures

The advantages of frame representation in structuring geographical objects are: (1) it provides a concise structural representation of the relations, and (2) it supports a concise definition-by-specialization technique [Fikes 85].

A frame's structure provides an extension for a relational model in two ways. It extends and generalizes the object and domain of an object's relations, and it organizes the corresponding relations hierarchically via inheritance relationships between objects. Applied to a geo-object representation, the frame structure will be used to organize classes of geo-objects, and then for structuring the topological relationships in multiple scale representations.

3.3.1 Hierarchy Over Relationships

Frames are well suited to handle geographical object relationships for the following reasons:

1. They allow aggregation of objects.
2. They support generalization and specialization.
3. The domain of the relationships are extended to include methods related to the objects.
4. The domain attribute of relations are generalized to any arbitrary type.
5. Objects and their classes are organized hierarchically through generalization and specialization using the inheritance facility described in section 2.2.2.

In terms of accessing and processing the information, a relational database is represented by either the relational predicate of slots, or by the relational object representing the slot's value (see the frame structure in Figure 3.1). A broader more, flexible relationship among the objects and their classes is maintained hierarchically.

Two major types of hierarchical relations implemented for the representation of geographical objects are hierarchical classification of geographical objects and hierarchical construction of topological relations.

3.3.2 Hierarchical Classification

A class of geographical objects is a set in which its members possess a common or similar object description. In a frame model, object classification falls naturally into a class inheritance, that is, a class construction obtained by placing the more general thing in the top of the class. The subsequent more specialized subclasses inherit both the structures and methods attached to its parent. Thus, it forms a *hierarchical classification* [Chand 89].

The frame's inheritance, in ART-IM for example, does not make a distinction between a class representation or an instance object. All schemas have the same status. Except for the instance object, where inheritance ends, everything can be inherited from the ancestors, unless it was declared as a having local-private identity. Topological relations are an example of private identity specific to one scale of representation.

A frame representation for hierarchical classification of geo-objects is shown in Figure 3.5. It is based on the DLG-E model for geographical object classification hierarchy [Gupti 90], as shown in Figure 3.6. On top of the hierarchy is the place holder for our abstraction about geo-object phenomena. Subsequently, using an is-a inheritance, more detailed classifications are added for each class or object in the subdivision. Using an inheritance hierarchy, this classification is extensible and easy to maintain to account for new application developments. The bottom classification level is the class of entity which has common properties and relationships called a *feature* [Gupti 90]. Rules are used to maintain the composition for each class.

(schema GEO-VIEW
(view-of geographic-reality)
(**composed-of geo-cover geo-division ...**))

(schema GEO-COVER
(is-a geo-view)
(class physical-material-on-earth)
(**composed-of builtup-land ...**)
(*view-of geographic-reality*))

(schema GEO-DIVISION
(is-a geoview)
(class geo-cultural-demarcation))

(schema BUILTUP-LAND
(is-a geo-cover *geo-view*)
(*class physical-material-on-earth*)
(**composed-of builtup-land-network ...**)
(*view-of geographic-reality*)
(physical-form structured-used-area))

(schema HIGHWAY
(is-a builtup-land-network *builtup-land geo-cover geo-view*)
(*class physical-material-on-earth*)
(**composed-of national-highway ...**)
(*physical-form structured-used-area*)
(*structure-form interconnected*)
(*view-of geographic-reality*))

(schema NATIONAL-HIGHWAY
(is-a highway *built-up-land builtup-land-network geo-cover geo-view*)
(*class physical-material-on-earth*)
(highway-class 1st)
(*physical-form structured-used-area*)
(*structure-form interconnected*)
(*view-of geographic-reality*))

Roman font indicates a user defined argument.
Italic font indicates an inherited argument.
Bold font indicates inserted by rules.

Figure 3.5: Hierarchical classification of geographical objects.

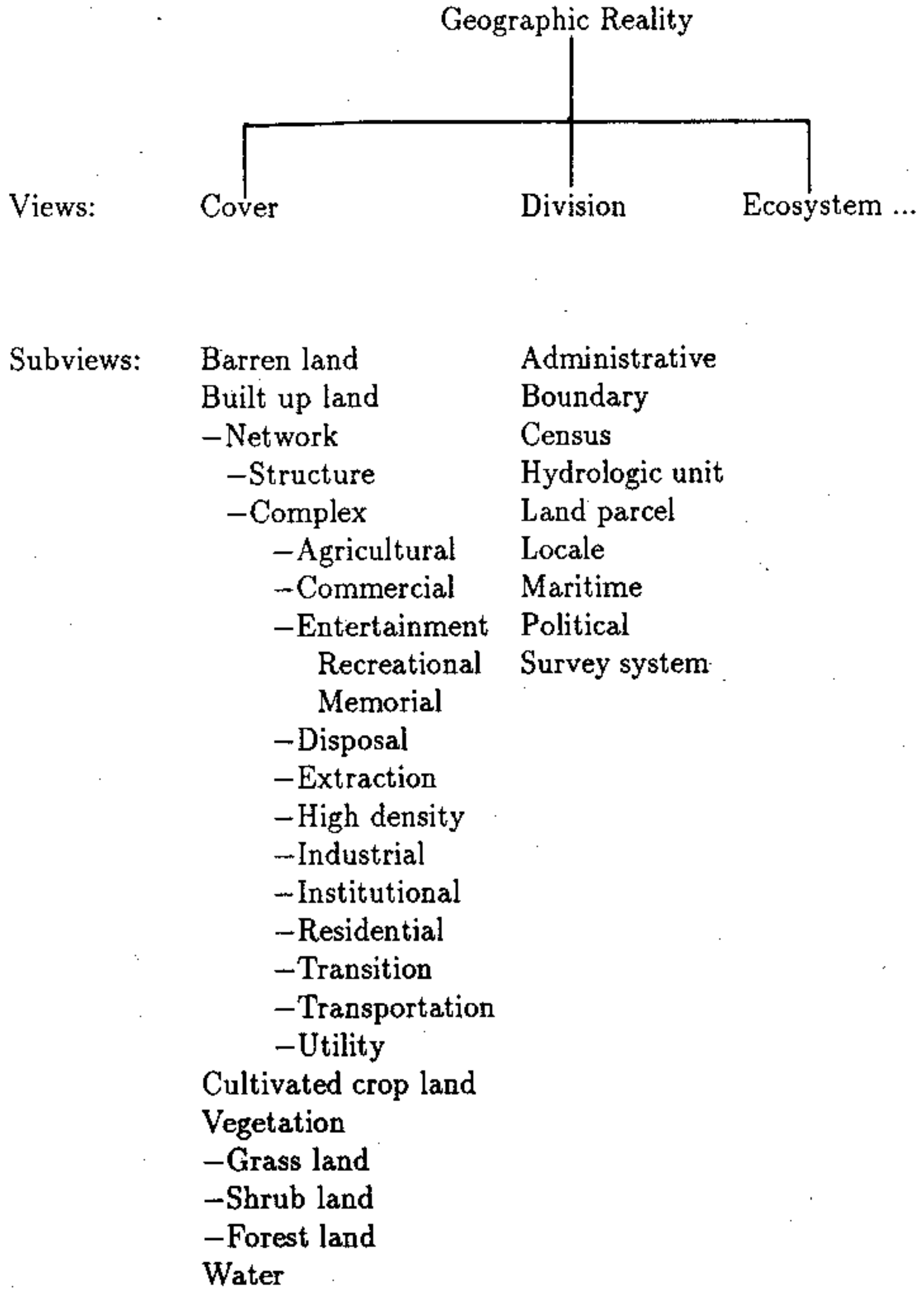


Figure 3.6: Part of the DLG-E model for geographical object classification [Gupti 90].

As an example, consider the frame representation of the Trans Canada highway NB-2 which is located in the province of New Brunswick, Canada. From the classification, this highway is classified as a national highway. At this stage, all information (possibly another frame) related to the object NB-2 must be collected and properly placed in the frame's slots. Some properties for highways are: name, construction type, and ownership. Therefore, a *generic* frame representation of Trans Canada Highway NB-2 is as shown in Figure 3.7.

(schema NB-2
"An independent-scale (generic-object) Trans Canada Highway NB-2."
(is-a national-highway *builtup-land builtup-land-network geo-cover geo-view highway*)
(*class physical-material-on-earth*)
(*highway-class 1st*)
(location new-brunswick)
(nation-name canada)
(*physical-form structured-used-area*)
(*structure-form interconnected*)
(*view-of geographic-reality*)
;(slots-of technical features))

Figure 3.7: Frame representation of the generic object highway NB-2.

So far there is no geometric information. Only the topological relation for the highway has been established; i.e. that NB-2 is located in New Brunswick. This generic information will be always true regardless of the resolution or the scale of representation. More flexible geo-object organizations can be arranged, and more geo-relationships are easily and naturally represented.

3.3.3 Topological Hierarchy

The topological representation of geo-objects varies with the scale of resolution. One can consider different planes as representing different scale representations of geo-object topological relationships. Smaller scales of representation result in more simplified representations of the objects. As illustrated in Figures 2.5 and 2.6, they are different objects topologically, but are in fact one object. An object at a smaller scale is a sub representation of objects at a larger scale. Any part of an object represented at a smaller scale must be recognized at a larger scale representation.

By using a frame structure, topological relationships of geo-objects are easily and naturally represented. A nested structure is formed by acknowledging that one topo-object is constructed by another topo-object. A composition or an inclusion is stated using a slot predicate composed-of. Multiple representations are automatically supported, since a slot of multiple values is allowed. An elevated object can be specified as an over-pass or under-pass object with respect to other objects.

A geometrical object along with its topological relations forms the basic geometrical attributes of a geo-object. Two more slots that must be added to the frame of generic objects are predicate slots *has-geo-rep*, which states that a geo-object has a geometrical representation, and predicate slot *super-rep-of* or *sub-rep-of*, which means that the geo-object is a larger scale or a smaller scale representation of a neighboring scale representation. This last predicate slot will be used and propagated in the topological object representations at all detail levels, performing the hierarchy of topological relations.

The following illustration in Figure 3.8 shows a schema structure for Trans Canada Highway NB-2 at scales 1 : 2,000,000 and 1 : 7,500,000. It has a chain topological representation. Both chains and their constructors may have topological hierarchy relationships to their upper and lower scale representation.

```
(schema h02002
   "Schema Representation of highway NB-2 at 1:2000000.”
   (is-a nb-2)
   (composed-of h0200201 h0200202 ..)
   (has-geo-rep ch02001)                  ; Chain representing h02002.
   (scale 2000000.0)
   (sub-rep-of)                           ; NIL.
   (super-rep-of h07002))                 ; NB-2 at scale 1 : 7,500,000.
(schema ch02001
   "Chain of topological representation NB-2 at 1 : 2,000,000.”
   (is-a chain)
   (geo-rep-of h02002)
   (super-rep-of ch07001)                 ; Chain representing h07002.
   (sub-rep-of)                           ; NIL.
   (constructed-by (ch0200101 E02004)))   ; Ordered topo-obj sequence.


(schema h07002
   "Schema Representation of highway NB-2 at 1:7,500,000.”
   (is-a nb-2)
   (composed-of h0700201 h0700202 ..)
   (has-geo-rep ch07001)                  ; Chain representing h07002.
   (scale 7500000.0)
   (sub-rep-of h02002)                    ; NB-2 at scale 1 : 2,000,000.
   (super-rep-of)                         ; NIL.
(schema ch07001
   "Chain of topo-objects representation NB-2 at 1 : 7,500,000.”
   (is-a chain)
   (geo-rep-of h07002)
   (super-rep-of)                         ; NIL.
   (sub-rep-of ch02001)
   (constructed-by (e07001 e07002 ..)))   ; Ordered topo-obj sequence.
```

Figure 3.8: An example frame structure for part of a highway. For a larger scale of representation, a more detailed object is represented. These two representations are shown in Figure 4.3 and Figure 4.2 respectively.

3.4 Frames of Geographical Objects

This section discusses the integration of concepts reviewed in the previous chapters and sections, to provide the structure that naturally represents the geo-object. In the first two subsections (3.4.1 and 3.4.2), the outline of structuring the aspatial and spatial components of geographical objects is given, followed by the discussion of **IMAS** (Intelligent Map Structure) in subsection 3.4.3. IMAS represents the hierarchical structuring of geo-objects based on a frame structure representation.

3.4.1 Feature Object Attributes

A feature object attribute is a nonlocational or aspatial attribute. Since a geo-object is characterized by its complex and nested representation, a feature object may have other feature objects, and an aspatial attribute may have multiple values of any object type.

Hierarchical classification through an inheritance hierarchy is the easiest way to implement and to maintain the organization of geographical objects. By using an inheritance hierarchy, everything which belongs to the ancestor object can be inherited by its successor object, modifying its value or adding its own characteristics. By placing the most common thing as high as possible on the top of the hierarchy, the more specialized objects can be defined subsequently as the child node in the hierarchy. These nodes represent either the class of objects or the object itself.

The illustrations in Figures 3.6 and Figure 3.5 imply that at the top of geo-object classification hierarchy is the superset of all sub-set geo-objects. Each subset provides the modification and/or adds the set difference compared to its parent set. The inheritance hierarchy accommodates these processes. This is the natural way to represent geographical object phenomena.

3.4.2 Spatial Object Attributes

A spatial object attribute represents the object location. At a particular representation in planar space, spatial objects with a different dimensionality may coexist representing a particular object. Due to map generalization at any scale of representation, the same object will have a different representation or possibly be extinguished. same hierarchically,

3.4.3 IMAS : A Frame Based Geo-object Structure

The requirements for design and implementation of a GIS as given in section 2.1 suggest that the right structure for GIS is one permitting direct interaction with knowledge bases. Intuitively, this structure is a frame structure which implements both an object oriented method, a semantic network, and has hooks to an inference engine. A combination of this structure with a certain querying technique, hypermedia technology, and an intelligent dialog language seems to be the right choice for a GIS environment that supports multiple representations, users, and applications in a large scale range and wide coverage area. The proposed structure can be defined as follows:

$$IMAS = Frame + ObjectOrientation + SemanticNet$$

This structure allows objects to be linked in semantic nets, and to obtain concepts through inheritance.

The basic structure in IMAS is a frame. An IMAS object exhibits different interpretations in different contexts. For example,

- It is a *frame* in an expert system.
- It is an *object* in an object oriented system.
- It is a *relation* in a relational system.

It unifies these diverse conceptual structures, and packages them into a common function representing the spatial and topological information. Objects can be accessed and processed either through their attribute predicate slot name, or via the object reference as a slot value. It is an active data structure which can provide a method or a procedure attached to the predicate slot.

In the IMAS structure, objects are organized hierarchically through inheritance. The spatial representation is assigned to one of the object's slots. A set of spatial entities of IMAS objects at a particular scale forms a plane of topological relations. This plane of topological relations is maintained hierarchically, such that there is a connection between two successive scale representations. The same logical object at different scales is connected and maintained in a correct and consistent way.

A summary of IMAS structures are illustrated as follows:

1. Figure 2.1 shows the hierarchical construction of particular geo-objects, and the frame of geo-object hierarchy represented in Figures 3.5 and 3.8.

2. Through the slot *has-geo-rep*, geo-objects are linked to their geometrical and topological representation, as shown for highway NB-2 in Figure 3.8.

3. All geo-objects represented at one particular scale make up one plane of scale representation. For example, all objects in New Brunswick at scale 1:2,000,000 are organized in one frame NB-2M, which is subsequently divided into NB-Highway-2M, NB-Boundary-2M etc. Figure 2.4 shows this concept.

4. Both planes or objects at any level of detail from two neighboring scales are linked via slots *sup-rep-of* and *sub-rep-of* depending on their relative scales. All objects represented at the smaller scales, i.e. less detailed, must be available at the larger, more detailed representation. Figure 3.8 shows an example of this relation and Figure 2.6 shows a layered model of multiple plane resolutions.

Chapter 4

Implementation and Evaluation

The prototype of the frame-based representation of geographical objects is implemented in an expert system shell called ART-IM/MS-DOS version 1.5 (ART-IM) [Infer 89a]. This shell is a high level programming environment containing user interface tools for text and graphics, interface functions for external functions and data integration, a memory management system, and an application deployment facility.

ART-IM is a subset of ART (Automatic Reasoning Tool). The basic syntax of ART-IM is simple, elegant, and reminiscent of LISP; LISP-like prefix notation and the familiar LISP parentheses are used [Brook 92]. ART-IM is also completely compatible with other ART-IM products available for mainframe platforms (e.g. ART-IM/MVS) and UNIX platforms. ART-IM is written in Microsoft C, which allows easy integration with other software [Infer 89b].

The prototype of the designed IMAS was implemented on an IBM PS/2 Model 70 386, with DOS 3.30, ART-IM 1.5, and Essential Graphics 2.0 embedded in it. It was tested using data provided by the National Atlas Information Service of the Department of Energy, Mines, and Resources (EMR), Canada.

4.1 Environment

The environment for designing and developing geographical applications which may incorporate expert knowledge should be a general purpose environment which supports reasoning. A general purpose expert system shell with industrial strength qualities will meet these requirements. ART-IM is one of the available shells. Other shells are also available (e.g. KEE, Nexpert Object, TIRS).

4.1.1 ART-IM/MS-DOS

ART-IM is a hybrid general purpose rule-based expert system development environment. The three parts of this tool are [Infer 89a] :

1. A declarative knowledge base which contains individual information called *facts*, or knowledge about the state of the relevant *objects* in the domain.
2. Heuristics or rules of thumb for the *rules* of rule-based programming, in the form
   "WHENEVER *conditions*, DO *actions*".
3. An inference engine that surveys the current circumstances of the knowledge base, and matches it against the rules in the knowledge base.

Conditions in the left-hand-site (LHS) describe *patterns* or descriptions of data in the knowledge base that must be matched to the rules. In ART-IM rules, data is a fact (state) or a frame or schema (object). ART-IM provides both object orientation and a reasoning facility.

The right-hand-site (RHS) actions change the database by modifying, creating, or retracting facts or schemas. There are three controlling methods to execute the action. They are (1) procedural programming, (2) rule-based programming, and (3) object-oriented programming. Different situations require a different controlling method. In implementing IMAS, all three methods were used.

4.1.2 ART-IM Tools

To support incremental application development, ART-IM provides both an interactive development environment called the STUDIO, and a library function user interface toolkit for rapid construction of textmode standard interfaces. Application programs can be loaded, run, browsed, and edited directly from the interactive interface.

A complete ART-IM development package is suitable for the design and development of complex GIS applications. It is provided with

1. A powerful interactive ART-IM Studio development environment.
2. Three different levels of user interface tools, including standard (text), graphics oriented, and screen handling primitive tools.
3. The capability for extension and integration with external data via the C interface.
4. A memory management system allowing applications to consume up to 16 Mbyte in protected mode.
5. A deployment facility which enables the developer to produce a version of the application that can be deployed with a minimum of start-up time and memory overhead.

4.1.3 Graphics Tools

Graphics display is an important functionality in GIS applications. It allows for the display of represented geo-objects. ART-IM provides an interface to an external Essential Graphics (EG) package [Essen 88]. EG's user graphics functions are written in Microsoft C, and can be interfaced to ART-IM by using the def-user-fun function described in [Infer 89a].

4.2 Test Data Description

Any point or geometrical object on a plane map sheet is a representation of its idealized position on the surface of earth's mathematical model, an ellipsoid. The position of geographical objects is represented by their longitude and latitude (ϕ,λ).

The implementation test data used for IMAS is from an EMR Canada electronic atlas data base of New Brunswick at scales of 1:7,500,000 (1:7.5M) and 1:2,000,000 (1:2M). These ASCII files are created using ARC/Info's UNGENERATE command, where feature code and coordinate strings are provided. Cartesian (x,y) coordinates in meters are used. The map projection is the Lambert Conformal Conic projection, with two standard parallels at 49° and 77°N, and the map origin at (49°N, 95°W).

4.2.1 The EMR Data Format

The format of EMR data used to test the IMAS prototype is shown in Figure 4.1. Basically, these are spaghetti files consisting of a collection of edges and areas. Each spatial object starts with the feature code id, follow by a list of (x,y) values. The lack of an object name makes it difficult to identify the objects. A paper copy of the complete feature descriptions (i.e. code, name, definition, and drawing color) is provided separately [EMR 90]. Test data at the 1:2M scale includes all the Atlantic provinces. The whole of Canada is covered by the 1:7.5M test data.

```
          35
2193348.250000   -55407.398500
2222269.000000 -206419.922000
2259288.500000 -235218.406000
2498716.000000 -380033.844000
END
```

Figure 4.1: Original EMR test data format [EMR 87].

In order to implement and test the IMAS prototype, only a portion of the data for the province of New Brunswick was taken, namely the Trans Canada highway NB-2, some primary and collector highways, the Saint John River, the province boundaries, sea coast lines, some lakes, and the Fundy National Park. These objects represent a total of 57 edges, (total 2603 coordinates pairs) for the 1:2M data, and 31 edges (total 942 coordinates pairs) for 1:7.5M data. The objects were displayed in Figure 4.3 and Figure 4.2, respectively.

4.2.2 Converting to IMAS

The representation of a *super model* [Leeyc 90], which is a superset model of all geographical object database formats, could be built using frames. Frames can encapsulate both the structure and the behavior of objects. Provided that all formats to be exchanged are well known, then the super model will hold these description in its slots. Thus, the super model is the interface model for all known formats, and converting the EMR format to the IMAS format can be considered as a format conversion problem.

The process of converting the test EMR data to the IMAS format encompasses the determination of the classes of the objects and the construction of the topological links for the geometrical objects. Automating this conversion process involves a substantial interactive graphics procedures and low level processing, and is beyond the scope of this research. The EMR data do not have a complete feature and topological description. For the purposes of this research a few objects were selected, displayed on the CRT, and an identification was made based on the source map. For each object on the display, its topological components were determined manually.

Assuming that there are no errors in the data, the complete topological construction as well as data display operations such as scale modification, zooming, panning, and object query are accomplished in the simple interactive display environment, developed for IMAS, called MAPBETA.

4.3 Plotting Software

MAPBETA was developed by the author using EG's graphical functions embedded in the ART-IM environment. Using rules for control, it provides a basic display facility and can be modified to add additional GIS functionality.

4.3.1 Software Description

MAPBETA is a simple knowledge-based display tool, developed for testing the implementation of the IMAS prototype using ART-IM/MS-DOS. The ART-IM environment supports incremental applications development. It allows the user to compile incrementally against all facts, schemas, and rules during the debugging session made with the EMACS editor [Infer 89b]. It also has a facility to interpret and integrate with the external environment using its ability to call C functions. It is flexible enough to accommodate any future additions to IMAS.

The two main components of MAPBETA are the user interface and the underlying geo-object representation. It allows the user to examine, manipulate, and receive instantly the desired answer about the graphical representation of the underlying geo-objects.

The essential components of IMAS incorporated into MAPBETA are as follows :

1. A Knowledge base for a simple interactive mapping environment.
2. A Knowledge base representation for hierarchical classification of geo-objects.
3. A Knowledge base representation for topological construction of geo-objects.

Which objects are displayed depends on the particular scale of presentation. The same physical object at different scales of representation are different geo-objects. In the MAPBETA implementation, the one that is closest to the required display scale is chosen.

MAPBETA was implemented using the ART-IM expert system shell environment version 1.5, running under DOS 3.0 or later releases, and using Microsoft C 5.0 along with the Essential Graphics package embedded in it. It requires an IBM PC/AT or PS/2 models or PC 100% compatible hardware, with at least 640K plus 2 MB extended memory, one hard disk with at least 8 MB free space [Infer 89b].

The prototype of MAPBETA has been developed in ART-IM development mode. It is a graphical application oriented system. Therefore, MAPBETA's user interface was designed separately from ART-IM's ui-tool-kit, since ART-IM's ui-tool-kit was for text mode application only. Both external C functions and ART-IM user functions were coded to implement MAPBETA. The main functions to implement MAPBETA are:

1. Screen definition which describes the display type and determines the transformation from map coordinates into screen coordinates and vice versa.
2. Clipping functions for lines and areas.
3. Display routines for the basic geometrical objects node, edge, area, box, and circle.
4. Menu functions for MAPBETA's interface.

All of the external functions in C and their interface ART-IM functions were collected into one file. Global variables along with all of MAPBETA's slot definitions comprise scale independent object definitions, and are placed into one file. Each scale's representation consisting of scale dependent object definitions and scale dependent data is placed in another file. ART-IM functions and all related rules for a particular IMAS construction are also organized in individual files. The overall diagram of MAPBETA components and the corresponding processes are given in section 4.4.1, Figure 4.5 and Figure 4.6.

4.3.2 The User Interface

The user interface for geo-objects, as well as its query language are significantly different from those using alphanumeric data exclusively. The MAPBETA user interface provides the user with an interactive mapping environment in which he/she can query interrogatively, and manipulate either the geometry or the representation of the geo-objects.

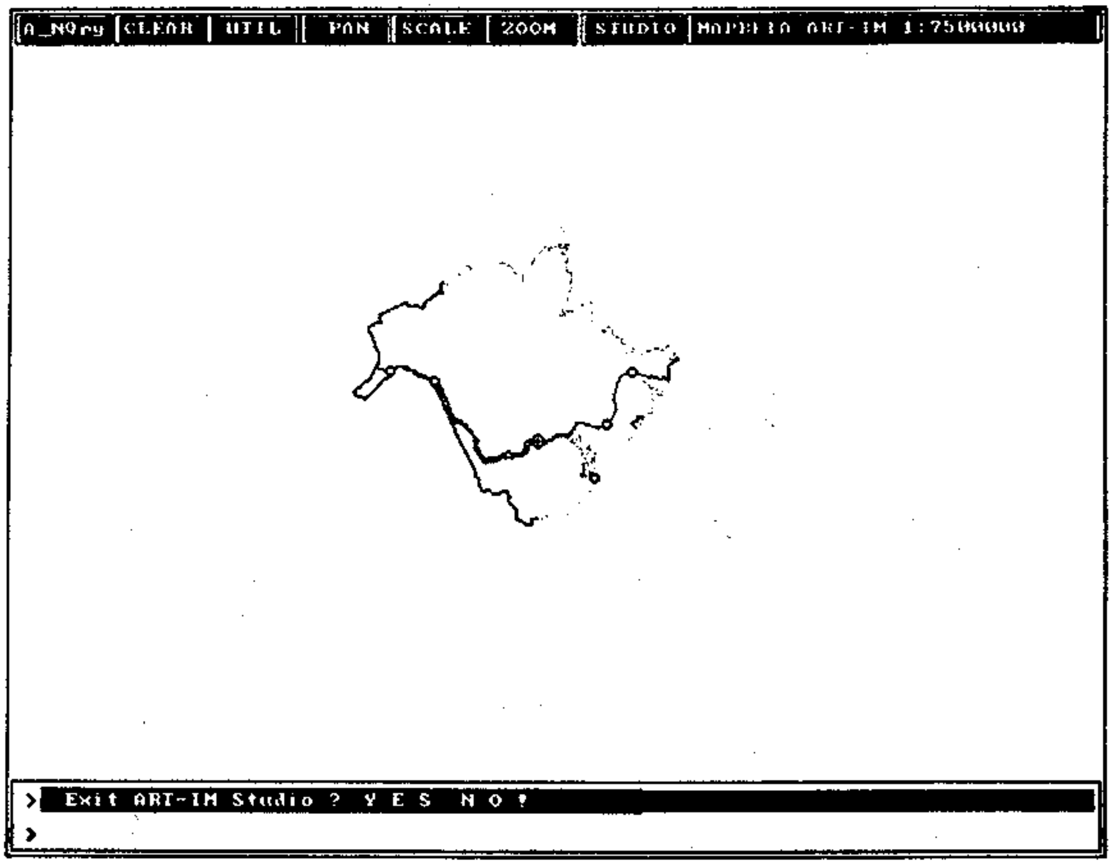
MAPBETA's user interface implements the basic characteristics of an interactive GIS user interface. It uses a frame representation to define the user interface and its menu or sub-menu. The whole user interface is a frame object. All the interface functions are implemented using rules. Figure 4.2 shows the MAPBETA interface. Prior to initial display, MAPBETA assumes that the hierarchical classification of geo-objects, and the topological relations among objects of one scale and between objects of different scales have been formed. The user responds interactively to the instructions, and provides the necessary values for the display.

For example, if a user wants to change the scale of display, he/she first selects the scale menu, and then types the desired scale number in the query area. In response to this new scale assertion, MAPBETA determines and searches for the appropriate object scale in the data base, refreshes the screen, and displays the object representation closest to the desired scale. The illustration in Figure 4.4 shows the display after the user chose a new scale equal to 1:1M. It shows a more detailed display compared to the 1:7.5M representation. The map menus which have been implemented are:

1. Scaling: determines the objects closest to the user's scale, refreshes the screen and displays the corresponding objects.

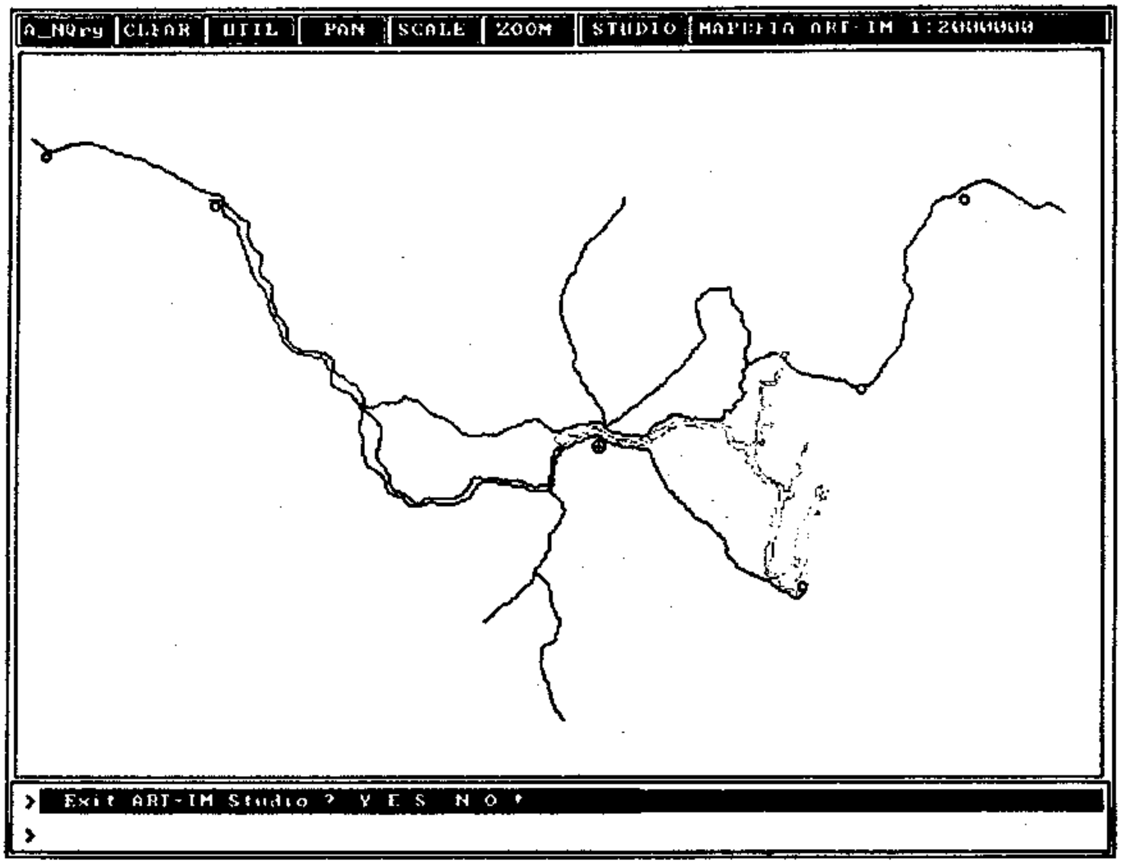
2. Panning: selects the user's view center, refreshes the screen and redisplays the object with the same scale representation.

3. Zooming: a combination of scaling and panning by selection of a certain bounding area from the current display.
4. Query: (1) object query by it's frame's name displays it if found and provides a highlight for the found object; (2) pick a point from the screen, return the map (x,y) coordinates, and enter the the identity for this node. These query types are useful for object identification and topological definition.
5. Utility: (1) display a default map at 1:7.5,000,000, (2) refresh the screen, and (3) dump all geo-objects and topological objects represented in the database.
6. Studio: return to the ART-IM Studio environment.



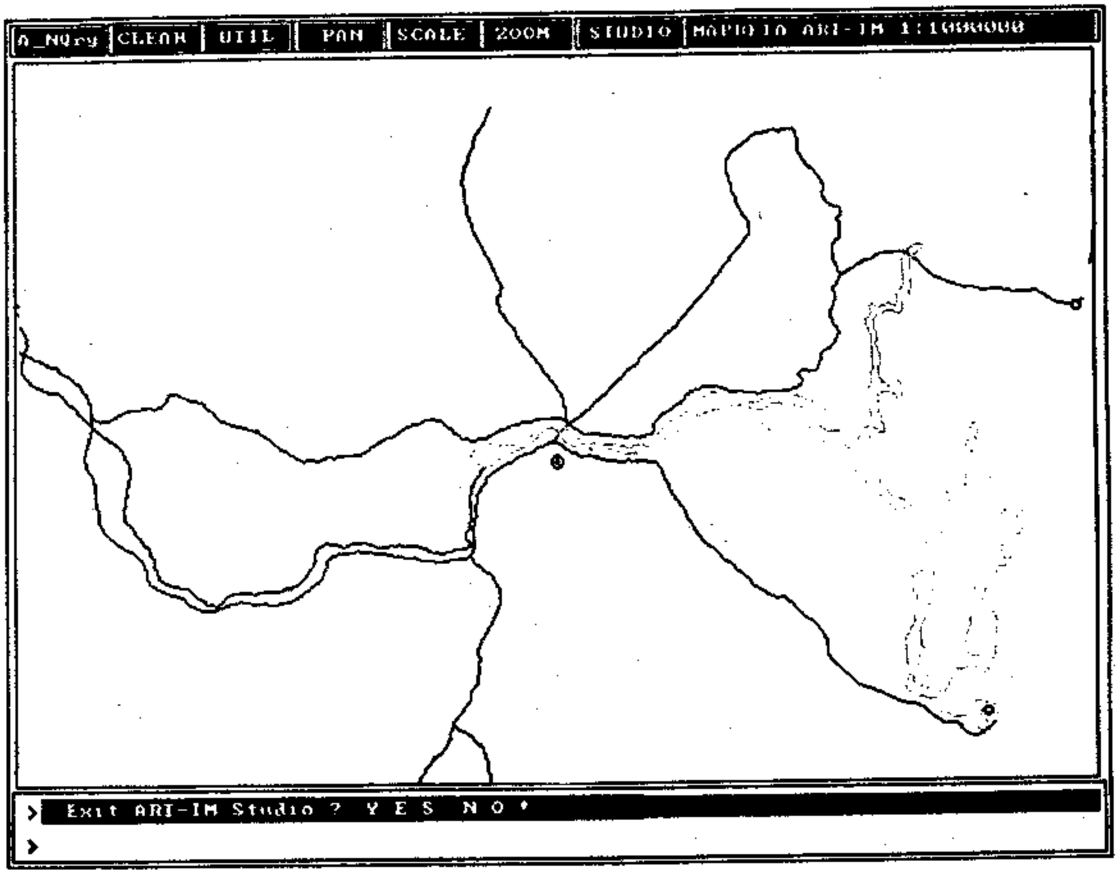
```
> Exit ART-IM Studio ? YES NO !
>
```

Figure 4.2: The MAPBETA interface. Also shown here the display of the Trans Canada Highway NB-2, Saint John River, and the provincial boundary of New Brunswick at a scale of 1:7.5M. The data used are from the 1:7.5M representation.



```
> Exit ART-IM Studio ? Y E S  N O ?
>
```

Figure 4.3: MAPBETA display at the scale 1:2M. The displayed objects are from the 1:2M geo-objects representation.



```
> Exit ARI-IM Studio ? Y E S  N O *
>
```

Figure 4.4: MAPBETA display at a new scale of 1:1M. The displayed objects are from the 1:2M geo-object representation. The crossing point between the Saint John river and Trans Canada Highway NB-2 in Fredericton, NB, Canada, at scale 1:7.5M is represented as a line using the 1:2M representation.

4.4 Evaluation

The ART-IM development environment provides various tools both for text and graphical applications. The IMAS prototype has been developed in ART-IM development environment. The technique for construction of IMAS objects is given, followed by an overview of their storage, memory, and time used to build the IMAS objects.

4.4.1 Construction of IMAS objects

Ideally, all object classes must exist in the geo-object classification hierarchy. Since only a few objects in the subset of geographic reality (see Figure 3.5) were implemented for the prototype IMAS structures, a geo-object classification hierarchy was provided only up to the classes of the objects used.

Only those objects available in particular scales are kept in the schema network plane of resolutions or plane of scale dependent object definitions. Figure 4.5 shows the procedures for the construction of one definition of scale dependent objects and its topological data for the corresponding scale. Figure 4.6 shows the construction of IMAS topological structures for one scale of representation and the IMAS topological hierarchy for multiple scale representations.

The geo-object class hierarchy which defines the scale independent object definitions and the EMR data at one particular scale together define one set of scale dependent objects. The user interactively describes the topological level of the IMAS model through the CRT display of the objects. More geometrical and topological information is calculated and added (e.g. bounding-box, begin-node and end-node of edges, merging of duplicate representations of geometrical objects). The output of this processing is IMAS data for one particular scale.

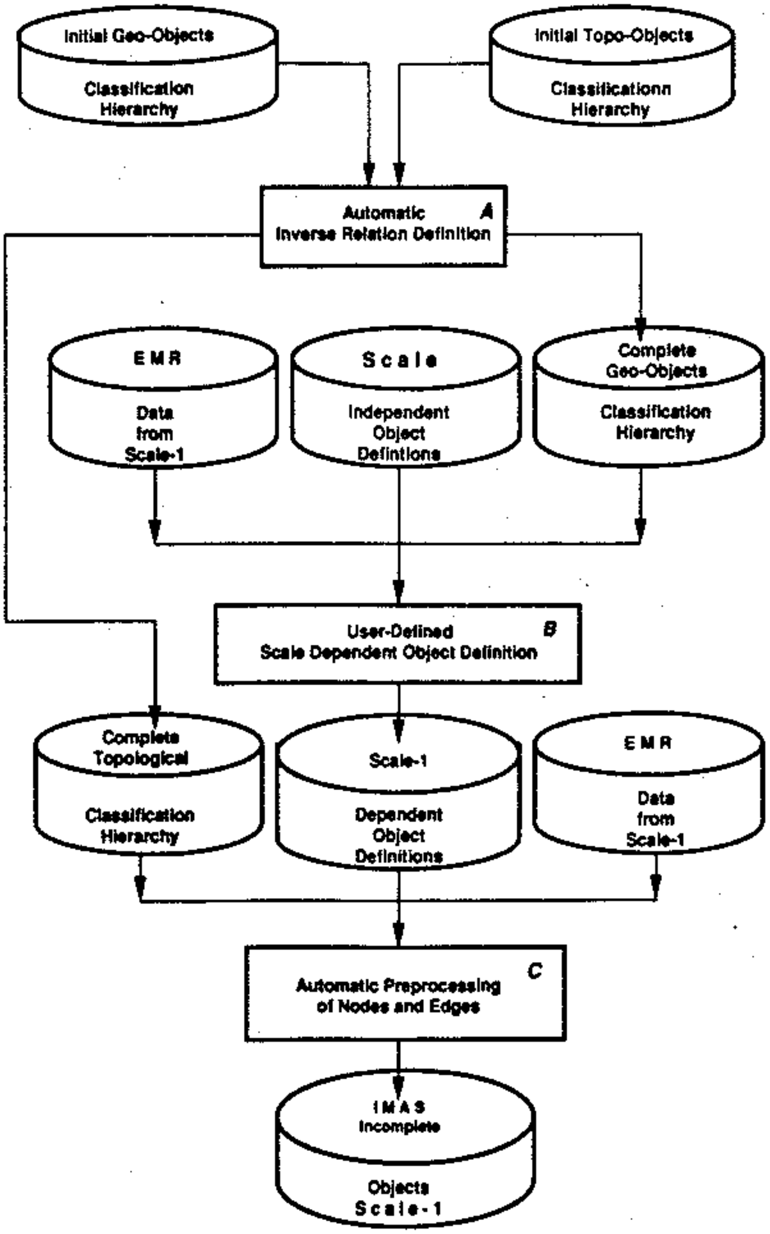


Figure 4.5: Flow diagrams for the IMAS construction of one scale representation.

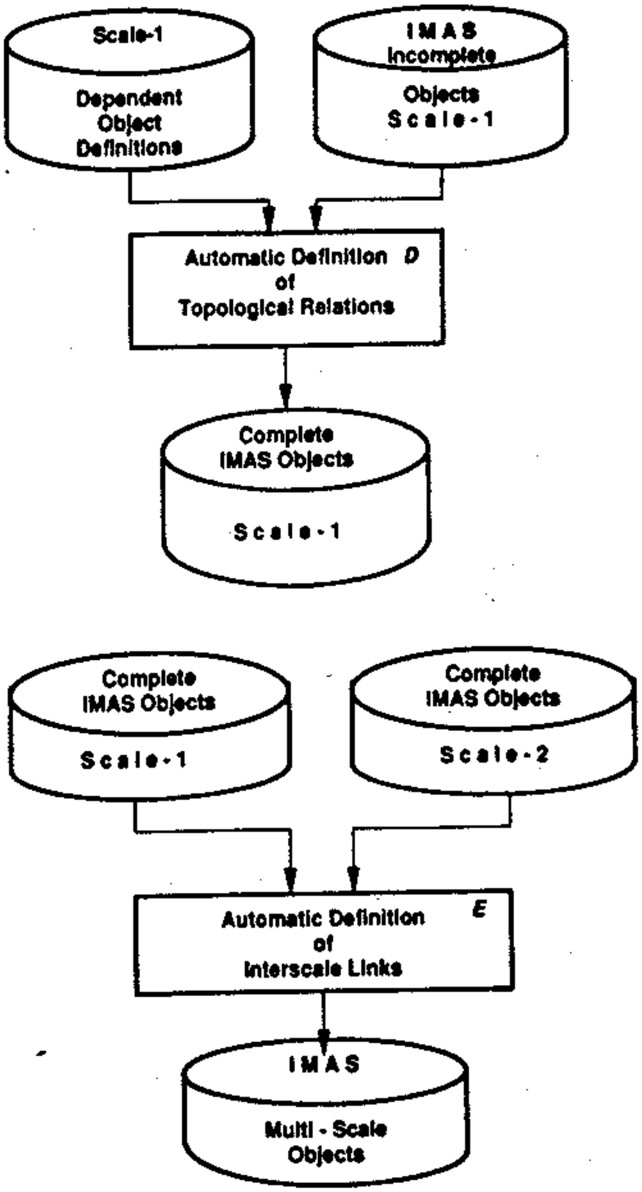


Figure 4.6: Flow diagram for the IMAS topological construction for one scale representation and for multiple scale representation.

The next step of this process is the topological processing for one scale of representation as shown in Figure 4.6. IMAS topological construction produces a complete set of topological links between neighboring scales as shown in Figure 4.6. This process is done interactively.

Map analysis as well as map display will be performed based on the output of a complete IMAS topological structure. Object dependent scale definitions and their corresponding data will be delivered and used for the next map's topological modification or construction.

4.4.2 Size of Data Structures

As mentioned in section 2.3.3., the size of geographical object representation depends on the shape of the objects and the scale of their representation. Additionally, the level of topological relations also affects the size of the topological representation. Trans Canada Highway NB-2 is represented by 1 chain of 6 edges (231 points) in 1:7.5M, and is represented by 1 chain of 5 sub-chains comprising 22 edges (613 points) in the 1:2M representation.

In IMAS frame topological structures, the number of schemas for topological objects, along with their slot attribute values will also increase with an increased level of topological relations. The amount of computer memory used by the IMAS frame structures can be found by using ART-IM's development function **print-memory-usage**. The memory usage are listed in Table 4.1 is different for MAPBETA with and without topological schemas included. MAPBETA itself occupies 741.2 KB.

Table 4.1 shows a comparison of file size and memory usage for geo-objects represented in 1:2M and 1:7.5M. The size of the original EMR data after porting them into IBM MVS TSO files are fixed at 80 bytes per record. For example, the edge shown in Figure 4.1 took 480 bytes.

Table 4.1: Size of IMAS Structures.

	TopoObjects		EMR Files	IMAS (Bytes)		
	Edges	Points	(Bytes)	Header	Data	Memory Used
1:2M	57	2603	52,560	22,357	88,502	1,415,096
1:7.5M	32	942	19,440	22,775	40,998	885,908
Both	89	3545	n/a	45,132	129,500	2,856,960

4.4.3 Speed in Building The Structures

ART-IM also has a timing function for application development (i.e. the *time* function). Table 4.2 and Table 4.3 show the time used for various processes involved in building IMAS structures discussed in section 4.4.1 on the IBM PS/2 Model 70/386. Process *E* takes 818.720 sec., 177.360 sec., and 0.720 sec. for load, reset, and run respectively. The same test on an IBM PS/2 Model 95/XP 486 shows a running time almost four times faster than for the IBM PS/2 Model 70. Most of the MAPBETA functions were written in the ART-IM language.

Table 4.2: Running time to build IMAS Structures 1:7.5M.

Process	Run Time (sec)		
	Load	Reset	Run
Process *A*: inverse relation hierarchy.	89.530	78.920	1.320
Process *C*: automatic preprocessing nodes and edges.			98.620
Process *D*: automatic definition of topological relations.			22.050

Preprocessing and the construction of IMAS objects takes a much longer time compared to the time required to display them. Loading time is the time used to load and interpret, the application and objects from disk to the computer's memory. The long loading time is due to the substantial schema processing to build the Rete network [Forgy 79] for all complex relationships. Reset time is the time used to initialize the knowledge base. Run time is the time used to process all facts and

Table 4.3: Running time to build IMAS Structures 1:2M.

Process	Run Time (sec)		
	Load	Reset	Run
Process *A*: inverse relation hierarchy.	98.590	88.380	1.320
Process *C*: automatic preprocessing nodes and edges.			187.460
Process *D*: automatic definition of topological relations.			114.740

schemas that matched to the patterns of the rules in the knowledge base.

4.4.4 Speed of Display Through the User Interface

Table 4.4 shows the time required for searching the corresponding topological objects, and displaying them, based on the geo-objects available at a particular scale of representations. The display time does not include the load, reset, and run operations required to initialize MAPBETA. It include the times to select the valid scale closest to the user's scale, select the corresponding valid geo-objects, and display them.

Table 4.4: Display time for IMAS structures.

	TopoObjects		Display Time (seconds)
	Edges	Points	
1:2M	57	2603	15.540
1:7.5M	32	942	8.180
Both	89	3545	23.720

Chapter 5

Summary

5.1 Summary

This thesis introduced a hierarchical geographical structure called IMAS (Intelligent Map Structures). It is based on the frame structure of expert systems, and provides a unifying framework for multiple scale representation of geographical objects. The components of IMAS are :

$$IMAS = Frame + ObjectOrientation + SemanticNet$$

IMAS is a powerful and flexible structure for storing geographical objects in particular, and spatial data in general. It maintains the topological links between two scales or resolutions hierarchically, such that there is a connection between two successive scale representations. The same logical object at different scales, at any level of detail, is connected and maintained in a correct and a consistent way.

IMAS also provides a natural link to expert systems since the frame representation can be used directly in rules which reason about geographical objects.

A set of software tools has been built to process unstructured geographical objects into the structured IMAS representation.

5.2 Recommendations for Future Research

Some suggestions for future research follow from this work:

1. A more complete consideration needs to be given to the integration of different data types. How can raster data be incorporated into IMAS? How well would the IMAS structure work in a hyper-text environment?
2. IMAS can be extended by considering other available major GIS data formats, such as the full EMR format for 1:50,000 and 1:250,000 scale data in Canada, and the DLG-E in the USA.
3. The speed of display of IMAS objects using the MAPBETA software is somewhat slow. How much faster is the display with a compiled version of the IMAS objects?
4. The IMAS structure serves well as a unifying representation for geographical objects. How can it be integrated with existing relational databases, or existing GISs?
5. The power of IMAS should become apparent in applications which require multiple scale of representations of geographical data (e.g. automated cartographic generalization). How difficult is it for such applications to be developed to take advantage of IMAS?
6. The existing MAPBETA software works for small amounts of data. What other software tools would be required to build the IMAS representation of much larger amounts of data? How can the bottleneck of providing the user-defined object definitions be speeded up?
7. The current prototype handles only two scales of data. What changes are necessary to extend IMAS to handle a larger scale range (e.g. 1:7.5M, 1:2M, 1:250,000 and 1:50,000)?

References

[Abler 87] R.F. Abler, "The National Science Foundation National Center for Geographic Information Systems", *Int. J. Geographical Information Systems*, Vol. 1, No. 4, pp. 303–326, 1987.

[Armst 90] M.P. Armstrong and P.J. Densham, "Database Organization Strategies for Spatial Decision Support Systems", *Int. J. Geographical Information Systems*, Vol. 4, No. 1, pp. 3–20, 1990.

[Barbe 88] T.J. Barber, G. Marshall and J.T. Boardman, "Tutorial – A Philosophy And Architecture For Rule-Based Frame System: RBFS", *Engineering Applications of AI*, Vol. 1, pp. 67–86, June 1988.

[Barst 84] D.R. Barstow and H.E. Shrobe, *Interactive Programming Environment,* McGraw-Hill, New York, pp. 558–570, 1984.

[Boley 90] H. Boley, "Expert Systems Shells: Very-high Level Languages for Artificial Intelligence", *Expert Systems*, Vol. 7, No. 1, pp. 1–8, February 1990.

[Bossl 90] J.D. Bossler, et. al., "Spatial Data Needs: The Future of the National Mapping Program", *Cartography and Geographic Information Systems*, Vol. 17, No. 3, pp. 237–242, 1990.

[Brach 85] R.J. Brachman and H.J. Levesque, "A Fundamental Tradeoff in Knowledge Representation and Reasoning", in *Readings In Knowledge Representation*, R.J. Brachman and H.J. Levesque, editors, Morgan Kaufmann Publishers, Inc., pp. 42–70, 1985.

[Brass 88] K.E. Brassel and R. Weibel, "A Review and Conceptual Framework of Automated Map Generalization", *Int. J. Geographical Information Systems*, Vol. 2, No. 3, pp 229–244, 1988.

[Brodi 89a] M.L. Brodie, "Future Intelligent Information Systems: AI and Database Technologies Working Together", in *Readings in Artificial Intelligence and Databases*, J. Mylopolous and M. Brodie, editors, Morgan Kaufmann Publishers, Inc., Palo Alto, CA, pp. 623–672, 1989.

[Brodi 89b] M.L. Brodie and F. Manola "Database Management: A Survey", in *Readings in Artificial Intelligence and Databases*, J. Mylopolous and M. Brodie, editors, Morgan Kaufmann Publishers, Inc., Palo Alto, CA, pp. 9–34, 1989.

[Brook 92] T. Brooke, "The Art of Production Systems", *AI EXPERT*, pp. 31–35, January, 1992.

[Brueg 89] B.P. Bruegger and A.U. Frank, "Hierarchies Over Topological Data Structures", *1989 ASPRS/ACSM Annual Convention*, Vol. 4 GIS/LIS, Baltimore–Maryland, pp. 137–145, 1989.

[Burro 87] P.A. Burough, *Principles of Geographical Information Systems for Land Resource Assessment*, Oxford Science Publications, 1987.

[Chand 89] B. Chandrasekaran, "Building Blocks for Knowledge-based Systems Based on Generic Task: The Classification and Routine Design Examples", in *Structuring Expert Systems : Domain, Design, and Development*, J. Liebowitz and D.A. De Salvo, editors, Yourdon Press., pp. 215–240, 1989.

[Clark 86] K.C. Clark, "Advances In Geographic Information Systems", *Comput. Environ. Urban Systems*, Vol. 10, No. 3/4, pp. 175–184, 1986.

[Cleme 90] E. Clementini, A. D'Atri and P.D. Felice, "Browsing in Geographic Databases: an Object Oriented Approach", *Proceeding of The 1990 IEEE Workshop on Visual Languages*, University of Pittsburgh, Illinois, USA, pp. 125–131, October 1990.

[DGIWG 90] DIGEST, *Final Draft Digital Geographic Information Exchange Standards*, DGIWG, Ottawa, Sept. 21, 1990.

[EMR 87] EMR, *National Atlas Data Base*, National Atlas Information Service EMR, Ottawa, 1990.

[EMR 90] EMR Canada, *Description of National Atlas Electronic Database File Formats*, Personal communication between Dr. B.G. Nickerson and Mr. Brian W. Cromie, December, 1990, 3pp.

[Egenh 87] M.J. Egenhofer, "An Extended SQL Syntax To Treat Spatial Objects", *Proceeding Second International Seminar on Trends and Concerns of Spatial Sciences*, Y.C. Lee, editor, UNB Fredericton, New Brunswick, Canada, 1987.

[Egenh 90] M.J. Egenhofer, "Manipulating the Graphical Representation of Query Results in Geographic Information Systems", *Proceeding of The 1990 IEEE Workshop on Visual Languages*, University of Pittsburgh, Illinois, USA, pp. 119–124, October, 1990.

[Essen 88] Essential Software Inc., *Essential Graphics 2.0 User's Manual*, So. Orange, NJ, 1988.

[Fikes 85] R. Fikes and T. Kehler, "The Role of Frame-Based Representation In Reasoning", *Communication of the ACM*, Vol. 28, No. 9, pp 904–920, September 1985.

[Forgy 79] C.L. Forgy, "*On the Efficient Implementation of Production Systems*", PhD Thesis, Department of Computer Science – Carnegie–Mellon University, USA, 1979.

[Frank 82] A. Frank, "Data Base Query Language for Retrieval of Geometric Data and Their Graphical Representation", *ACM Computer Graphics*, Vol. 16, No. 3, July 1982.

[Gaheg 88] M.N. Gahegan and S.A. Roberts, "An Intelligent, Object-Oriented GIS", *Int. J. Geographical Information Systems*, Vol. 2, No. 2, pp 101–110, 1988.

[Gibso 90] E. Gibson, "Objects — Born and Bred", *B Y T E*, pp. 245–254, October 1990.

[Goodc 90] M.F. Goodchild, "The National Center For Geographic Information and Analysis", *ACSM Bulletin*, pp. 35–40, December 1990.

[Gowri 90] T.R. Gowrishankar and N.G. Bourbakis, "Specifications For The Development Of A Knowledge-based Image Interpretation Systems", *Engineering Applications of AI*, Vol. 3, pp. 79–90, June 1990.

[Gupti 90] S.C. Guptill, editor, "An Enhanced Digital Line Graph Design", *U.S. Geological Survey Circular 1048*, 1990.

[Hemby 90] D.M. Hembry, "Knowledge-based systems in the AD/Cycle environment", *IBM Systems Journal*, Vol. 29, No. 2, 1990.

[Infer 89a] Inference Corporation, *ART-IM Programming Language Reference,* Los Angeles, CA, 1989.

[Infer 89b] Inference Corporation, *ART-IM in The DOS Environment,* Los Angeles, CA, 1989.

[Infer 89c] Inference Corporation, *DOS/16M User's Manual,* Los Angeles, CA, 1989.

[Infer 89d] Inference Corporation, *ART-IM Function Library Reference,* Los Angeles, CA, 1989.

[Jones 86] C.B. Jones and I.M. Abraham, "Design and Considerations for A Scale Independent Cartographic Database", *Proceeding Second International Symposium on Spatial Data Handling*, pp 384–398, Seattle, Washington, 1986.

[Leeyc 87] Y.C. Lee, "*Conceptual Models for Geographic Information Systems*", PhD Thesis, Department of Surveying Engineering – University of New Brunswick, Canada, 1987.

[Leeyc 90] Y.C. Lee, "An Object-Oriented Environment For GIS Data Exchange", *Proceedings of GIS for the 1990s*, CISM, Ottawa, Canada. March 1990.

[Leung 90] K.S. Leung and M.H. Wong, "An Expert-System Shell Using Structured Knowledge", *IEEE Computer*, pp. 38–47, March 1990.

[Lisko 87] B. Liskov "Data Abstraction and Hierarchy", *Addendum to the Proceedings of OOPSLA '87*, pp 17–34, October 1987.

[Magui 90] D.J. Maguire, M.F. Worboys and H.M. Hearnshaw, "An Introduction To Object-Oriented Geographical Information Systems", *Mapping Awareness*, Vol. 4, No. 2, pp. 36–39, March 1990.

[Micro 87] Microsoft Corporation, *Microsoft C 5.1 Optimizing Compiler Code View and Utilities,* Redmont, WA, 1987.

[Minsk 75] M. Minsky, "A Framework for Representing Knowledge", in *Reading in Knowledge Representation*, R.J. Brachman and H.J. Levesque, editors, Morgan Kaufmann Publishers, Inc., pp. 245–262, 1985.

[Mohan 88] L. Mohan and R.L. Kashyab, "An Object-Oriented Knowledge Representation for Spatial Information", *IEEE Transactions On Software Engineering*, Vol. 14, No. 5, pp. 675–681, May 1988.

[Mylop 89] J. Mylopolous, M. Brodie, "Introduction", in *Reading in Artificial Intelligence and Databases*, J. Mylopolous and M. Brodie, editors, Morgan Kaufmann Publishers, Inc. pp. 1–7, 1989.

[NCGIA 89] NCGIA, "The Research Plan of The National Center for Geographic Information and Analysis", *Int. J. Geographical Information Systems*, Vol. 3, No. 2, pp. 117–136, 1989.

[Nicke 87] B.G. Nickerson, "Extraction of Data Within a Window From a USGS 1:24000 Scale DLG Data File", *E T L*, 1987.

[Nicke 88a] B.G. Nickerson, "Data Structure Requirements for Automated Cartographic Generalization", *3rd International Seminar on Trends and Concerns of Spatial Sciences*, Laval University, Canada, June 1988.

[Nicke 88b] B.G. Nickerson, "Automated Cartographic Generalization for Linear Features", *Cartographica*, Vol. 25, No. 3, pp. 15–66, 1988.

[Ooste 90] P.v. Oosterom, "A Modified Binary Space Partitioning Tree for Geographic Information Systems", *Int. J. Geographical Information Systems*, Vol. 4, No. 2, pp. 133–146, 1990.

[Peuqu 84] D.J. Peuquet, "A Conceptual Framework and Comparison of Spatial Data Models", *Cartographica*, Vol. 21, No. 4, pp. 66–113, 1984.

[Roger 85] D.R. Rogers, *Procedural Elements for Computer Graphics,* McGraw-Hill, New York, pp. 179–185, 1985.

[Roman 90] G.C. Roman, "Formal Specification of Geographic Data Processing Requirements", *IEEE Transactions On Knowledge And Data Engineering*, Vol. 2, No. 4, pp. 370–380, 1990.

[SAIF 91] CGSB 91, "Draft National Standard of Canada: SAIF Vol. I – IV", *B.C. Surveys and Resource Mapping Branch*, Release 1.0, August, 1991.

[Samet 84] H. Samet et. al., "Use of Hierarchical Data Structures in GIS", *Proceeding International Symposium on Spatial Data Handling*, Zurich, Switzerland, pp 392-403, 1984.

[Smith 87] T.R. Smith, "Requirements And Principles for The Implementation And Construction of Large-scale Geographic Information Systems", *Int. J. Geographical Information Systems*, Vol. 1, No. 1, pp. 13–31, 1987.

[Tello 89] E.R. Tello, *Object-Oriented Programming For Artificial Intelligence: A Guide To Tools And System Design*, Addison–Wesley, pp. 26–28, 1989.

[Thomp 89] J.P. Thompson, *DATA WITH SEMANTIC : Data Models and Data Management,* Van Nostrand Reinhold, 1989.

[Tomli 87] R.F. Tomlinson, "Current And Potential Uses of Geographical Information Systems The North American Experience", *Int. J. Geographical Information Systems*, Vol. 1, No. 3, pp 203–218, 1987.

[Webst 90] B. Webster, Rule-based Spatial Search, *Int. J. Geographical Information Systems*, Vol. 4, No. 3, pp. 241–259, 1989.

[Weile 80] K. Weiler, "Polygon Comparison Using a Graph Representation", *Computer Graphics*, Vol. 14, pp. 10–18, 1980.

[Wilki 87] G.G. Wilkinson and P.F. Fisher, Recent Development and Future Trends in Geo-Information Systems, *The Cartographic Journal*, Vol. 24, pp. 64–70, 1987.

[Wonki 90] K. Won, "Research Directions in Object-Oriented Database Systems", *Proceedings of the 9th ACM SIGACT-SIGMOD-SIGART Symposium on Principle of Database Systems*, Nashville, Tennessee, USA, April 1990.

Appendix A

IMAS Schema Notation

This appendix provides the description of the IMAS frame syntax based on the ART-IM "BNF-like" schema grammar [Infer 89a, pp 1-7 – 1-11]. ART-IM borrowed a Lambda-list notation from COMMON LISP for describing the functions. In ART-IM, a frame is called a *schema*. IMAS domain objects implemented in MAPBETA are provided in Appendix B.

A.1 Regular Schema

A *regular schema* is a schema that can:

- represent an object.
- inherit information.
- have a symbolic name.
- contain zero or more slots of any type or content.

It is defined as:

defschema ::= **defschema** *schema-name*
[*documentation-string*]
{*slot-form*} *

The symbols are interpreted as follows :

- **defschema** is an ART-IM expression (*special-form*) which does not returning a value, used to define both regular and slot schema.
- *special-form* is an ART-IM expression that accepts *arguments*, may or may not evaluate those arguments, and may or may not return a value.
- An *argument* may be any legal ART-IM data type, variable, or other ART-IM form.
- *schema-name* must be a symbol and represents the name of the object or class of objects. It is required. Applied to IMAS, a schema-name may be any *geographical-object* class or *topological-object* class.
- *documentation-string* is an optional string providing some informational comment about the schema.

A **slot-form** is a set of one or more elements, enclosed in parentheses ().

slot-form ::= (*slot-form* {*value*} *)

- The first element is the required *slot-name* and must be a symbol.
- The remaining elements are the *values* of the slots.
- Parentheses, (), are literal. It is presumed that a pair of parentheses surround the entire ART-IM form.

- Braces,{}, surround a set of enclosed arguments, the meaning is derived from the associated symbol:
    - {} * the enclosed argument is optional and may appear *zero or more times.*
    - {} + the enclosed argument is required and may appear one or more times.
    - {*a* | *b*} indicates that either *a* or *b* must be used.
- A `slot` is described by a special type of schema called a `slot schema` (see Section I.2 below).
- If the slot is a *relation-slot*, the *value(s)* must be a symbol.
- If the slot is an *attribute-slot*, the *value(s)* may be any legal `art-object`.
- A *slot-form* having no *value* specified , it creates an "empty" slot.
- A *relation-slot* is a slot representing a relationship between objects, (e.g., `is-a highway`).

A.2 Slot Schema

A **slot schema** is a specialized schema used to describe the characteristics of a particular slot. It has the following properties:

- describe an attribute or relation.
- do not themselves inherit information, but use of these slots may be inherited from one object to another.
- have a symbolic name.
- may only has specific slots specifying type, inheritability, and any constraints on slot values.

The slot schema is defined as:

defschema ::= **defschema** *slot-name*
[*documentation-string*]
{*slot-specifier*} +

- *slot-name* is the name of the slot, and it must be a symbol.
- *documentation-string* is an optional information command.

A *slot-specifier* is a slot in the slot schema used to define the behavior of *slot-name*.

slot-specifier ::= {(`istance-of` *slot-type*) |
[(`inherits-value` *inherits-value*)] |
[(`cardinality` *cardinality-value*)]}

- The *slot-type* must be one of the following symbols:
  - `slot` for an attribute slot, representing an attribute or characteristic of an object.
  - `relation` for a relation slot, representing a relationship between objects.
- Unless (`instance-of slot`) or (`instance-of relation`) is specified, the schema is assumed to be a *regular schema*.
- The *inherits-value* must be one of the following symbols:
  - **`yes`**, indicating that uses of this slot are inherited via inheritance links.
  - `no`, indicating that uses of this slot are *not* inherited via inheritance links.
- The *cardinality-value* must be one of the following symbols:
  - `single`, the slot may accept zero or one *value*.
  - `multiple`, the slot may accept zero or more *values*.

- The default characteristics of the slot schema are:
  - *attribute slot*
    - `instance-of - slot`
    - `inherits - yes`
    - `cardinality single`
  - *relation slot*
    - `instance-of - relation`
    - `inherits - no`
    - `cardinality multiple`

A.3 Geographical Object

Geographical objects are grouped into classes or sets in the geographical object hierarchy. A feature of geographical objects, e.g. primary-highway, is at the leaf of this classification hierarchy. An independent scale object abstraction or a "generic" object is a distinct feature object, e.g. NB-2 is a Trans Canada highway located in New Brunswick and named as NB-2.

A *geo-meta-class-name* is a class/metaclass or a subset/set hierarchy of geographical phenomena.

geo-meta-class ::= **defschema** *geo-meta-class-name*
[*documentation-string*]
{(`is-a geo-class`)}
(*class-specifier* {*value*} *)

A `geo-class` is the name of particular class of geo-objects in the hierarchy. A *class-specifier* is a *slot-specifier*, specifying the characteristic of geographical objects in the class. Two specific class specifiers are *part-of* and *composed-of* which indicate the class or objects memberships, and *has-geo-rep* which says that the scale dependent geographical object has a geometrical or topological representation.

class-specifier ::= {(`istance-of` *slot-type*) |
[(`inherits-value` *inherits-value)*] |
[(`cardinality` *cardinality-value*)]}

A *geo-object-name* is the name of the schema representing a scale independent object or a "generic-object". A `leaf-geo-class` is at the leaf level of the geographical object hierarchy.

geo-object-name ::= **defschema** *geo-object-name*
[*documentation-string*]
{(`is-a leaf-geo-class`)}
(*object-specifier* {*value*} *)

An *object-specifier* is the independent representation of attributes of the independent scale objects. These attributes are always true regarding to its scale or resolution representations.

A.4 Topological Object

Topological objects are grouped into a class or set of the topological object hierarchy. Geometrical objects with a geometrical position (x,y) only exist for a simple geometrical object such as a point (x,y) and the edge class of of objects with a sequence of points (x1,y1, .. xn,yn).

A *topo-meta-class-name* is a class/metaclass or a subset/set hierarchy of geometrical and topological objects.

topo-meta-class ::= **defschema** *topo-meta-class-name*
[*documentation-string*]
{(`is-a topo-class`)}
(*class-specifier* {*value*} *)

A `topo-class` is the analog of `geo-class` for geometrical and topological objects. Specific class specifiers for geometrical and topological object is *topo-specifier* which defines the topological relations or topological construction of topological objects. For examples, *part-of* and *constructed-by* which indicate an element and a composite object. The slot *geo-rep-of* says that this object is a geometrical or topological representation of the corresponding dependent scale representation of geographical objects. It also has topo-specifier *x-value* and *y-value*, slots which are valid only for the class of nodes and edges.

Appendix B

IMAS Geographical and Topological Objects

This appendix provides the schema definition for IMAS geographical and topological objects implemented in the MAPBETA environment. First, it gives the slot schema definitions, followed by the schema definition for geographical objects and topological objects. All slot schemas have the default value as described in Appendix I, unless it is otherwise redefined.

B.1 IMAS Slot Schema Definitions.

```
(defschema view-of (instance-of slot))
(defschema view-of-class
   (instance-of slot)
   (inherits yes)
   (cardinality multiple))
;;; Geo-cover
;;;
(defschema physical-form (instance-of slot))
(defschema structure-form (instance-of slot))
;;; Highway
(defschema passage (instance-of slot))
(defschema name (instance-of slot))
(defschema highway-class (instance-of slot))
;;; Attributes status of the highways
(defschema operation-status
   (instance-of slot)
   (inherits yes)
   (cardinality single))
```

```
(defschema access
   (instance-of slot)
   (inherits no)
   (cardinality single))
(defschema road-direction
   (instance-of slot)
   (inherits no)
   (cardinality single))
;;; Attribute definitions for highway construction types.
(defschema construction-type
   "The highway's construction type."
   (instance-of relation)
   (inherits no)
   (cardinality multiple))
(defschema surface (instance-of slot))
(defschema divided (instance-of slot))
(defschema lanes (instance-of slot))
(defschema seasonal-access (instance-of slot))
;;; Sub-View-1: Builtup-land Sub-View-2: Complex
(defschema complex-func
   (instance-of slot)
   (inherits yes)
   (cardinality multiple))
(defschema park-class
   (instance-of slot)
   (inherits no)
   (cardinality single))
(defschema transport-utility
   (instance-of slot)
   (inherits yes)
   (cardinality single))
(defschema interchange-designator (instance-of slot))
;;; View: Geo-Cover Sub-View: Water Cover: River.
(defschema elevation (instance-of slot))
(defschema water-form (instance-of slot))
(defschema flow-direction (instance-of slot))
(defschema delineation-status (instance-of slot))
(defschema hydro-category (instance-of slot))
(defschema hydro-form (instance-of slot))
(defschema land-form (instance-of slot))
(defschema bound-land (instance-of slot))
(defschema hydro-site
   (instance-of relation)
   (inherits yes)
   (cardinality single))

;;; View: Geo-Division.
;;;
(defschema demarcation (instance-of slot))
(defschema sovereignity (instance-of slot))
(defschema settlement-order (instance-of slot))
(defschema settlement-of (instance-of slot))
(defschema division-order (instance-of slot))
(defschema bounding-class (instance-of slot))
;;; NB-Head: Net-header definitions for New Brunswick based on EMR Code.
(defschema emr-code
   "Name and its standard drawing based on EMR-Code."
   (instance-of slot)
   (inherits yes)
   (cardinality single))
```

```
;;; NetSlot Predicates
(defschema net-of
   (instance-of relation)
   (inherits yes)
   (cardinality single))
(defschema has-obj-rep
   "Has-Object-Representations."
   (instance-of relation)
   (inherits no)
   (cardinality multiple))
(defschema location
   (instance-of relation)
   (inherits yes)
   (cardinality single))
(defschema nation-name
   "Nation or country where the objects are located."
   (instance-of relation)
   (inherits yes)
   (cardinality single))
;;; Slot of resolution/scale hierarchies.
;;;
;;;
(defschema has-geo-rep
   (instance-of relation)
   (inherits no)
   (cardinality single))
(defschema sub-rep-of
   "Object 1:2M is the sup-rep-of object 1:7.5M."
   (instance-of relation)
   (inherits no)
   (cardinality multiple))
(defschema super-rep-of
   "Object 1:7.5M is the sub-rep-of object 1:2M."
   (instance-of relation)
   (inherits no)
   (cardinality multiple))
```

B.2 IMAS for Geographical Object Classification

```
(defschema GEO-VIEW
   "Place holder TOP hierarchy of Geographic Reality."
   (view-of geographic-reality))
;;; GEO-COVER:
;;; Physical or material features at a location on or near the earth's surface.
;;; Strictly based on form, at the lowest level may be differentiated by function.
(defschema GEO-COVER
   (is-a geo-view)
   (view-of-class physical-material-on-earth))
(defschema BUILTUP-LAND
   "Structures and areas associated with intensive land-use."
   (is-a geo-cover)
   (physical-form structured-used-area))
(defschema BU-LAND-NETWORK
   "Interconnect set of constructions used for transport or communication."
   (is-a builtup-land)
   (structure-form interconnected-construction))
```

```
(defschema HIGHWAY
   "An open way for PASSAGE of NON-RAILED VEHICLES."
   (is-a bu-land-network)
   (passage non-railed))
(defschema NATIONAL-HIGHWAY
   (is-a highway)
   (highway-class 1st))
(defschema PRINCIPAL-HIGHWAY
   (is-a highway)
   (highway-class 2nd))
(defschema COLLECTOR-HIGHWAY
   (is-a highway)
   (highway-class 3rd))
(defschema LOCAL-HIGHWAY
   (is-a highway)
   (highway-class 4th))
(defschema LOCAL-ROAD
   (is-a highway)
   (highway-class 5th))
(defschema ROAD-CONST-TYPE
   (surface unknown))
(defschema DUAL-HIGHWAY
   (instance-of road-const-type)
   (surface hard)
   (divided yes))
(defschema ROAD-A
   "Hard undivided-road more than 2 lanes."
   (instance-of road-const-type)
   (surface hard)
   (lanes 3)
   (divided no))
(defschema ROAD-A1
   "Hard divided-road non-standard."
   (instance-of road-const-type)
   (surface hard)
   (lanes non-standard)
   (divided yes))
(defschema ROAD-AA
   "Hard divided-road 2 lanes or more."
   (instance-of road-const-type)
   (surface hard)
   (lanes 2)
   (divided yes))
(defschema ROAD-B
   "Hard undivide-road 2 lanes."
   (instance-of road-const-type)
   (surface hard)
   (lanes 2)
   (divided no))
(defschema ROAD-C
   "Hard undivided-road less than 2 lanes."
   (instance-of road-const-type)
   (surface hard)
   (lanes 1)
   (divided no))
(defschema ROAD-D
   "Loose or stabilized road 2 lanes or more."
   (instance-of road-const-type)
   (surface loose)
   (lanes 2)
   (divided))
```

```
(defschema ROAD-E
   "Loose or stabilized road less than 2 lanes."
   (instance-of road-const-type)
   (surface loose)
   (lanes 1)
   (divided))
(defschema ROAD-F
   "Loosed surface - dry weather only."
   (instance-of road-const-type)
   (surface loose)
   (lanes)
   (divided)
   (seasonal-access dry-weather))
(defschema ROAD-G
   "Cart track - winter only."
   (instance-of road-const-type)
   (surface cart-track)
   (lanes)
   (divided)
   (seasonal-access winter-road))
(defschema ROAD-H
   "Trail cutline - unknown seasonal-access."
   (instance-of road-const-type)
   (surface trail); may be cutline or portrail
   (lanes)
   (divided)
   (seasonal-access unknown))
;;; COMPLEX: Intensively used area, much of the land covered by constructions.
;;;
(defschema COMPLEX
   (is-a builtup-land)
   (structure-form intensive-construction-coverage))
(defschema PARK
   "Entertainment / recreational / memorial complex."
   (is-a complex)
   (complex-func recreational memorial))
(defschema NATIONAL-PARK
   (is-a park)
   (park-class 1st));
(defschema TRANSPORT-AREA
   "Designated area for transportation utility."
   (is-a complex)
   (complex-func transportation))
(defschema INTERCHANGE
   "Traffic access area from one road to another."
   (is-a transport-area)
   (transport-utility road-traffic-access-area))
;;; GEO-COVER: Sub-View-1: WATER-COVER.
;;;
(defschema WATER-COVER
   "Flowing or standing water with channels or basins."
   (is-a geo-cover)
   (view-of-class flowing-water standing-water))
(defschema RIVER
   "A body of flowing water."
   (is-a water-cover)
   (water-form flowing-water))
(defschema LAKE
   "A standing water surrounded by land."
   (is-a water-cover)
   (water-form standing-water))
```

```
(defschema SHORE-LINE
   "A natural contact line between a body of water and the land."
   (is-a water-cover)
   (water-form contact-of-water-and-land))
;;; GEO-MORPHOLOGY : is the world view based on land form.
;;; This land form must be named, labeled, or symbolized as distinct entities.
;;;
(defschema GEO-MORPHOLOGY
   "View based on land form - named - labeled or symbolized."
   (is-a geo-view)
   (view-of-class physical-land-form))
(defschema ISLAND
   "Dry or relatively dry land area surrounded by water or
   low wetland."
   (is-a GEO-MORPHOLOGY)
   (land-form dry-area)
   (bound-land water))
(defschema LAKE-ISLAND
   "Island on the lake."
   (is-a ISLAND)
   (hydro-site lake))
(defschema RIVER-ISLAND
   "Island on the river."
   (is-a ISLAND)
   (hydro-site river))

;;; GEO-DIVISION
;;;
(defschema GEO-DIVISION
   (is-a geo-view)
   (view-class geo-cultural-demarcation))
(defschema GEO-POLITIC
   (is-a geo-division)
   (demarcation world-politic-sovereignity))
(defschema NATION
   "An area under the jurisdiction of a sovereign government."
   (is-a geo-politic)
   (sovereignity 1st))
(defschema PROVINCE
   (is-a geo-politic)
   (sovereignity 2nd))
(defschema TERRITORY
   (is-a geo-politic)
   (sovereignity 3rd))
(defschema DISTRICT
   (is-a geo-politic)
   (sovereignity 4th))
(defschema COUNTY
   (is-a geo-politic)
   (sovereignity 5th))
(defschema SETTLEMENT-AREA
   (is-a geo-politic)
   (demarcation population-area))
(defschema CAPITAL-CITY
   (is-a settlement-area)
   (settlement-order 1)
   (settlement-of government-official))
(defschema CITY
   (is-a settlement-area)
   (settlement-order 2))
```

```
(defschema TOWN
   (is-a settlement-area)
   (settlement-order 3))
(defschema VILLAGE
   (is-a settlement-area)
   (settlement-order 4))
(defschema COMMUNITY
   (is-a settlement-area)
   (settlement-order 5))
;;; View: GEO-DIVISION Sub-View-1 GEO-BOUNDARY.
(defschema GEO-BOUNDARY
   "Part or all of a bounding/separating line on earth surface."
   (is-a geo-division)
   (demarcation geo-bounding))
(defschema GEO-BOUNDARY-LINE "Significance separation between divisions."
   (is-a geo-boundary)
   (bounding-class separation-of-divisions))
(defschema INTERNATION-B-LINE
   "Inter-nation boundary line."
   (is-a geo-boundary-line)
   (division-order nation))
(defschema PROVINCE-B-LINE
   "Inter-province boundary line."
   (is-a geo-boundary-line)
   (division-order province))
(defschema TERRY-B-LINE
   "Inter-territory boundary line."
   (is-a geo-boundary-line)
   (division-order territory))
(defschema DISTRICT-B-LINE
   "Inter-district boundary line."
   (is-a geo-boundary-line)
   (division-order district))
(defschema COUNTY-B-LINE
   "Inter-county boundary line."
   (is-a geo-boundary-line)
   (division-order county))
(defschema CITY-B-LINE
   "Inter-city boundary line."
   (is-a geo-boundary-line)
   (division-order settlement-area))
(defschema GEO-BOUNDARY-POINT
   "An identified location on a boundary."
   (is-a geo-boundary)
   (bounding-class identified-point))
(defschema GEO-FENCE-LINE
   "Fence or field line visible on an aerial photograph."
   (is-a geo-boundary)
   (bounding-class fence-line-on-aerophoto))
(defschema GEO-DATE-LINE
   "Meridian line designated for calendar day."
   (is-a geo-boundary)
   (bounding-class meridian-date-line))
```

B.3 IMAS for Topological Object Representation

```
;;; Inverse slot predicate has-geo-rep of Geographical-Objects
;;;
(defschema geo-rep-of
   "Topological and geometrical representation of geograpical object."
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
;;; Slots definition for simple topological objects.
;;;
(defschema dimension
   (instance-of slot)
   (inherits yes)
   (cardinality single))
(defschema x-value
   (instance-of slot)
   (inherits yes)
   (cardinality single))
(defschema y-value
   (instance-of slot)
   (inherits yes)
   (cardinality single))
(defschema bound-box
   (instance-of slot)
   (inherits yes)
   (cardinality single))
;;; Slot schema definition for NODE :
(defschema beg-node
   "Highger dimOfGeometricalObjects begins/ends from/to the node."
   (instance-of relation)
   (inherits yes)
   (cardinality single))
(defschema end-node
   (instance-of relation)
   (inherits yes)
   (cardinality single))
(defschema beg-node-of
   "Reverse relation of beg/end-node."
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema end-node-of
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
;;; Slot schema definitions for EDGE.
(defschema left-area
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema right-area
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema obj-connected-in
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
```

```
(defschema obj-connected-out
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema obj-contained-in
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema left-polygon
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema right-polygon
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
;;; Slots schema definition for AREA
;;;
(defschema composed-of
   (instance-of relation)
   (inherits no)
   (cardinality multiple))
(defschema consist-of
   (instance-of relation)
   (inherits no)
   (cardinality multiple))
(defschema bounded-by
   "Ordered sequence of edges composing the area."
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema constructed-by
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema part-of
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema bounding-area
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema obj-within
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema neighbor-area
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema neighbor-polygon
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema connection-in-of
   (instance-of relation)
   (inherits no)
   (cardinality multiple))
```

```
(defschema connection-out-of
   (instance-of relation)
   (inherits no)
   (cardinality multiple))
;;; TOPOLOGICAL OBJECT :
(defschema TOPO-OBJECT
   (topo-structure un-defined)
   (composed-of))
(defschema BASIC-GEO-OBJECT
   (is-a topo-object)
   (topo-structure simple)
   (dimension un-defined)
   (composed-of))
(defschema NODE
   (is-a basic-geo-object)
   (dimension 0)
   (beg-node-of)
   (end-node-of)
   (part-of)
   (x-value)
   (y-value)
   ;(topo-format fix-topo-node) ; Topology formator for node.
   ;(gr-display gr-node)) ; Display method.
(defschema EDGE
   (is-a basic-geo-object)
   (dimension 1)
   (beg-node)
   (end-node)
   (geo-rep-of)
   (part-of)
   (left-area)
   (right-area)
   (bound-box)
   (x-value)
   (y-value)
   ;(topo-format fix-topo-edge) ; Topology formator of edge.
   ;(gr-display poly-line)) ; Dispaly function.
(defschema AREA
   (is-a basic-geo-object)
   (dimension 2)
   (beg-node)
   (end-node)
   (geo-rep-of)
   (part-of)
   (bound-box)
   (consist-of)
   (constructed-by) ; Ordered of edges/arcs enclosing the area.
   (neighbor-area) ; OTHER AREA surround.
   ;(topo-format fix-topo-area)
   ;(gr-display gr-poly-clip)) ; Topology formator of area.
;;; Slots definition for COMPLEX-OBJECT.
(defschema object-function
   (instance-of slot)
   (inherits yes)
   (cardinality single))
(defschema beg-object
   "Highger complex dimOfGeometricalObjects begins/ends from/to the junction."
   (instance-of relation)
   (inherits yes)
   (cardinality single))
```

```
(defschema end-object
   (instance-of relation)
   (inherits yes)
   (cardinality single))
(defschema beg-object-of
   "Reverse relation of beg/end-object."
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema end-object-of
   (instance-of relation)
   (inherits yes)
   (cardinality multiple))
(defschema COMPLEX-OBJECT
    (is-a topo-object)
    (topo-structure complex)
    (object-function un-defined)
    (composed-of))
(defschema JUNCTION
    (is-a complex-object)
    (object-function topo-node)
    (beg-object-of) ; All objects begin from junction.
    (end-object-of) ; All objects ending into junction.
    (geo-rep-of) ; GeoMetricalRepOf GEO-OBJECT.
    (topo-format fix-topo-junction) ; Topology formator of junction.
    (constructed-by)) ; OBJECT/s COMPOSING the JUNCTION.
(defschema CHAIN
    (is-a complex-object)
    (object-function topo-edge)
    (beg-object) ; JUNCTION BEGINning of the CHAIN.
    (end-object) ; ENDing of the CHAIN.
    (geo-rep-of)
    (part-of)
    (obj-connected-in) ; ALL OBJECTS ENDING in the CHAIN.
    (obj-connected-out) ; BEGIN from the CHAIN.
    (left-polygon) ; All POLYGONS on the LEFT of CHAIN.
    (right-polygon) ; the RIGHT of CHAIN.
    (bound-box) ; BOX ENCLOSING objects composing the CHAIN.
    (constructed-by)) ; ORDERED of OBJECTS COMPOSING the CHAIN.
    ;(topo-format fix-topo-chain)) ; Topology formator of chain.
(defschema POLYPLEX
    (is-a complex-object)
    (object-function topo-area)
    (beg-object) ; JUNCTION begin.
    (end-object) ; JUNCTION begin.
    (geo-rep-of)
    (part-of)
    (obj-connected-in) ; ALL OBJECTS ENDING in the polygon.
    (obj-connected-out) ; All objects BEGIN from the polygon.
    (consist-of) ; All other geo-topo objects rep.  contained in.
    (bound-box) ; BOX enclosing the bounding AREA.
    (bounding-area) ; EDGES composing the AREA of the POLYGON.
    (neighbor-polygon) ; OTHER POLYGONS surround.
    (constructed-by)) ; ORDERED of OBJECTS ENCLOSING the POLYGON.
    ;(topo-format fix-topo-polygon)) ; Topology formator of polygon.
```

Appendix C

IMAS As Built in MAPBETA

The following Appendix III describes the schema definitions for IMAS objects selected from the EMR data for New Brunswick scales 1:2,000,000 and 1:7,500,000. The schemas show the schema values after IMAS objects completely constructed and ready for use, for example to display the objects etc. The geographical objects selected are as follows :

Table C.1: Geo-objects represented in MAPBETA.

		Representation	
No.	Object Name:	1:7.5M	1:2M
1.	Provincial boundary of New Brunswick	x	-
2.	Trans Canada Highway NB-2	x	x
3.	Principal Highway NB-104, NB-105, NB-3	-	x
	NB-4, NB-7, NB-8, NB-10.	-	x
4.	St John River	x	x
5.	Coast line of St John River	x	x
6.	Fundy Bay coast line	x	
7.	Atlantic coast line	x	
8.	Fundy National Park	x	

C.1 IMAS Geo-Objects

C.1.1 Scale Independent Representation

```
(DEFSCHEMA NB-HIGHWAY
   "Net of New-Brunswick-Highway."
   (COMPOSED-OF NB-NAT-HIGHWAY NB-PRI-HIGHWAY)
   (HAS-OBJ-REP NB-HIGHWAY-2 NB-HIGHWAY-7)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF HIGHWAY))
(DEFSCHEMA NB-NAT-HIGHWAY
   "Net of NB-National-Highway GENERIC Level."
   (IS-A NB-HIGHWAY)
   (COMPOSED-OF NB-2)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF NATIONAL-HIGHWAY))
(DEFSCHEMA NB-2
   "TransCanHighway NB-2."
   (IS-A BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
         HIGHWAY NATIONAL-HIGHWAY)
   (HIGHWAY-CLASS 1ST)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-2")
   (NATION-NAME CANADA)
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA NB-PRI-HIGHWAY
   "Net of NB-Principal-Highway GENERIC Level."
   (IS-A NB-HIGHWAY)
   (COMPOSED-OF NB-10 NB-104 NB-105 NB-3 NB-4 NB-7 NB-8)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF PRINCIPAL-HIGHWAY))
(DEFSCHEMA NB-10
   "PrincipalHighway NB-10."
   (IS-A BU-LAND-NETWORK BUILT UP-LAND GEO-COVER GEO-VIEW
         HIGHWAY PRINCIPAL-HIGHWAY)
   (HIGHWAY-CLASS 2ND)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-10")
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA NB-105
   "PrincipalHighway NB-105."
   (IS-A BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
         HIGHWAY PRINCIPAL-HIGHWAY)
   (HIGHWAY-CLASS 2ND)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-105")
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (VIEW-OF GEOGRAPHIC-REALITY)
```

```
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA NB-104
   "PrincipalHighway NB-104."
   (IS-A BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
         HIGHWAY PRINCIPAL-HIGHWAY)
   (HIGHWAY-CLASS 2ND)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-104")
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA NB-8
   "PrincipalHighway NB-8."
   (IS-A BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
         HIGHWAY PRINCIPAL-HIGHWAY)
   (HIGHWAY-CLASS 2ND)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-8")
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA NB-7
   "PrincipalHighway NB-7."
   (IS-A BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
         HIGHWAY PRINCIPAL-HIGHWAY)
   (HIGHWAY-CLASS 2ND)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-7")
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA NB-4
   "PrincipalHighway NB-4."
   (IS-A BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
         HIGHWAY PRINCIPAL-HIGHWAY)
   (HIGHWAY-CLASS 2ND)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-4")
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA NB-3
   "PrincipalHighway NB-3."
   (IS-A BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
         HIGHWAY PRINCIPAL-HIGHWAY)
   (HIGHWAY-CLASS 2ND)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-3")
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (VIEW-OF GEOGRAPHIC-REALITY)
```

```
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA NB-PROVINCE-B-LINE
   "Rel.-Frame Set of Location vs ObjHierarhy."
   (IS-A NB-BOUNDARY-LINE)
   (COMPOSED-OF NB-MAI-BDY NB-NS-BDY NB-PQ-BDY)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF PROVINCE-B-LINE))
(DEFSCHEMA NB-BOUNDARY-LINE
   "SET of boundary-line in NB."
   (COMPOSED-OF NB-PROVINCE-B-LINE)
   (HAS-OBJ-REP NB-BOUNDARY-LINE-7)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF GEO-BOUNDARY-LINE))
(DEFSCHEMA NB-NS-BDY
   (IS-A GEO-BOUNDARY GEO-BOUNDARY-LINE GEO-DIVISION
         GEO-VIEW PROVINCE-B-LINE)
   (BOUNDARY-OF (NEW-BRUNSWICK NOVA-SCOTIA))
   (BOUNDING-CLAS SEPARATION-OF-DIVISIONS)
   (DEMARCATION GEO-BOUNDING)
   (DIVISION-ORDER PROVINCE)
   (LOCATION NEW-BRUNSWICK)
   (NAME "New Brunswick - Nova Scotia Border")
   (VIEW-CLAS GEO-CULTURAL-DEMARCATION)
   (VIEW-OF GEOGRAPHIC-REALITY))
(DEFSCHEMA NB-PQ-BDY
   (IS-A GEO-BOUNDARY GEO-BOUNDARY-LINE GEO-DIVISION
         GEO-VIEW PROVINCE-B-LINE)
   (BOUNDARY-OF (NEW-BRUNSWICK QUEBEC))
   (BOUNDING-CLAS SEPARATION-OF-DIVISIONS)
   (DEMARCATION GEO-BOUNDING)
   (DIVISION-ORDER PROVINCE)
   (LOCATION NEW-BRUNSWICK)
   (NAME "New Brunswick - Quebec Border")
   (VIEW-CLAS GEO-CULTURAL-DEMARCATION)
   (VIEW-OF GEOGRAPHIC-REALITY))
(DEFSCHEMA NB-MAI-BDY
   (IS-A GEO-BOUNDARY GEO-BOUNDARY-LINE GEO-DIVISION
         GEO-VIEW PROVINCE-B-LINE)
   (BOUNDARY-OF (NEW-BRUNSWICK MAINE))
   (BOUNDING-CLAS SEPARATION-OF-DIVISIONS)
   (DEMARCATION GEO-BOUNDING)
   (DIVISION-ORDER PROVINCE)
   (LOCATION NEW-BRUNSWICK)
   (NAME "New Brunswick - Maine Border")
   (VIEW-CLAS GEO-CULTURAL-DEMARCATION)
   (VIEW-OF GEOGRAPHIC-REALITY))
(DEFSCHEMA NB-RIVER
   (IS-A NB-WATER-COVER)
   (COMPOSED-OF ST-JOHN-RIVER)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF RIVER))
(DEFSCHEMA NB-COAST-LINE
   (IS-A NB-WATER-COVER)
   (COMPOSED-OF NB-ATLANTIC-COAST NB-FUNDY-COAST-E
         NB-FUNDY-COAST-W ST-JOHN-RIVER-COAST)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF SHORE-LINE))
(DEFSCHEMA NB-WATER-COVER
   "Net of New-Brunswick-Water-Cover."
   (COMPOSED-OF NB-COAST-LINE NB-RIVER)
```

```
   (HAS-OBJ-REP NB-WATER-COVER-2 NB-WATER-COVER-7)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF WATER-COVER))
(DEFSCHEMA ST-JOHN-RIVER
   (IS-A GEO-COVER GEO-VIEW RIVER WATER-COVER)
   (CONSIST-OF)
   (ELEVATION 100.1)
   (FLOW-DIRECTION ONE-WAY)
   (HYDRO-CATEGORY PERENNIAL)
   (LOCATION NEW-BRUNSWICK)
   (NAME "St.  John River")
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS FLOWING-WATER PHYSICAL-MATERIAL-ON-EARTH
         STANDING-WATER)
   (WATER-FORM FLOWING-WATER))
(DEFSCHEMA ST-JOHN-RIVER-COAST
   (IS-A GEO-COVER GEO-VIEW SHORE-LINE WATER-COVER)
   (HYDRO-AREA ST-JOHN-RIVER)
   (LOCATION NEW-BRUNSWICK)
   (NAME "StJohn Coast")
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS FLOWING-WATER PHYSICAL-MATERIAL-ON-EARTH
         STANDING-WATER)
   (WATER-FORM CONTACT-OF-WATER-AND-LAND))
(DEFSCHEMA NB-FUNDY-COAST-W
   (IS-A GEO-COVER GEO-VIEW SHORE-LINE WATER-COVER)
   (HYDRO-AREA FUNDY-BAY)
   (LOCATION NEW-BRUNSWICK)
   (NAME "Fundy Coast of Canada")
   (SITE WEST)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS FLOWING-WATER PHYSICAL-MATERIAL-ON-EARTH
         STANDING-WATER)
   (WATER-FORM CONTACT-OF-WATER-AND-LAND))
(DEFSCHEMA NB-FUNDY-COAST-E
   (IS-A GEO-COVER GEO-VIEW SHORE-LINE WATER-COVER)
   (HYDRO-AREA FUNDY-BAY)
   (LOCATION NEW-BRUNSWICK)
   (NAME "Fundy Coast of Canada")
   (SITE EAST)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS FLOWING-WATER PHYSICAL-MATERIAL-ON-EARTH
         STANDING-WATER)
   (WATER-FORM CONTACT-OF-WATER-AND-LAND))
(DEFSCHEMA NB-ATLANTIC-COAST
   (IS-A GEO-COVER GEO-VIEW SHORE-LINE WATER-COVER)
   (HYDRO-AREA ATLANTIC)
   (LOCATION NEW-BRUNSWICK)
   (NAME "Atlantic Coast of Canada")
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS FLOWING-WATER PHYSICAL-MATERIAL-ON-EARTH
         STANDING-WATER)
   (WATER-FORM CONTACT-OF-WATER-AND-LAND))
(DEFSCHEMA NB-NATIONAL-PARK
   "All national parks in the Province of New Brunswick."
   (IS-A NB-PARK)
   (COMPOSED-OF FUNDY-NATIONAL-PARK)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF NATIONAL-PARK))
(DEFSCHEMA NB-PARK
```

```
   "All park in the Province of New Brunswick."
   (COMPOSED-OF NB-NATIONAL-PARK)
   (HAS-OBJ-REP NB-PARK-7)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF PARK))
(DEFSCHEMA FUNDY-NATIONAL-PARK
   "Fundy National Park."
   (IS-A BUILTUP-LAND COMPLEX GEO-COVER GEO-VIEW
         NATIONAL-PARK PARK)
   (COMPLEX-FUNC MEMORIAL RECREATIONAL)
   (LOCATION NEW-BRUNSWICK)
   (NAME "Fundy National Park")
   (PARK-TYPE RECREATIONAL)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (STRUCTURE-FORM INTENSIVE-CONSTRUCTION-COVERAGE)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
```

C.1.2 IMAS Geo-Object Schemas for Scale 1:2,000,000

```
(DEFSCHEMA NB-HIGHWAY-2
   "All highways in NB at scale 1:2M."
   (INSTANCE-OF NB-HIGHWAY)
   (COMPOSED-OF NB-NAT-HIGHWAY-2 NB-PRI-HIGHWAY-2)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF HIGHWAY)
   (SCALE 2000000.0))
(DEFSCHEMA NB-NAT-HIGHWAY-2
   "All nat-highways in NB at scale 1:2M."
   (INSTANCE-OF NB-HIGHWAY NB-NAT-HIGHWAY)
   (COMPOSED-OF H02002)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF NATIONAL-HIGHWAY)
   (SCALE 2000000.0))
(DEFSCHEMA NB-PRI-HIGHWAY-2
   ”Representation of all nat-highways in NB at scale 1:2M.”
   (INSTANCE-OF NB-HIGHWAY NB-PRI-HIGHWAY)
   (COMPOSED-OF H02003 H02004 H02007 H02008 H02010 H02104
        H02105)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF PRINCIPAL-HIGHWAY)
   (SCALE 2000000.0))
(DEFSCHEMA H02002
   "Schema Representation of highway NB-2 at 1:2000000."
   (IS-A BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
        HIGHWAY NATIONAL-HIGHWAY NB-2)
   (COMPOSED-OF H0200201 H0200202 H0200203 H0200204 H02002051
        H02002052 H02002053 H0200206 H0200207)
   (CONSIST-OF)
   (EMR-CODE (107 5))
   (HAS-GEO-REP CH02001)
   (HIGHWAY-CLASS 1ST)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-2")
   (NATION-NAME CANADA)
   (OPERATION-STATUS OPERATIONAL)
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
```

```
   (SCALE 2000000.0)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (SUB-REP-OF)
   (SUPER-REP-OF H07002)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA H0200201
   "Object of nb-2 at 1:2M."
   (INSTANCE-OF BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
        H02002 HIGHWAY NATIONAL-HIGHWAY NB-2)
   (ACCES NONE)
   (CONSIST-OF)
   (CONSTRUCTION-TYPE ROAD-AA)
   (EMR-CODE (107 5))
   (HAS-GEO-REP CH0200101)
   (HIGHWAY-CLASS 1ST)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-2")
   (NATION-NAME CANADA)
   (OPERATION-STATUS OPERATIONAL)
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (ROAD-DIRECTION BI-DIRECTIONAL)
   (SCALE 2000000.0)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (SUB-REP-OF)
   (SUPER-REP-OF)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA H0200202
   (INSTANCE-OF BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
        H02002 HIGHWAY NATIONAL-HIGHWAY NB-2)
   (ACCES NONE)
   (CONSIST-OF)
   (CONSTRUCTION-TYPE ROAD-AA)
   (EMR-CODE (107 5))
   (HAS-GEO-REP E02004)
   (HIGHWAY-CLASS 1ST)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-2")
   (NATION-NAME CANADA)
   (OPERATION-STATUS OPERATIONAL)
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (ROAD-DIRECTION BI-DIRECTIONAL)
   (SCALE 2000000.0)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (SUB-REP-OF)
   (SUPER-REP-OF)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
```

C.1.3 IMAS Geo-Object Schema for Scale 1:7,500,000

```
(DEFSCHEMA NB-HIGHWAY-7
   "Plane or Net of all nbClasshighway at scale 1:7.5M."
   (INSTANCE-OF NB-HIGHWAY)
```

```
   (COMPOSED-OF NB-NAT-HIGHWAY-7)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF HIGHWAY)
   (SCALE 7500000.0))
(DEFSCHEMA NB-NAT-HIGHWAY-7
   "Plane or Net of all nat-highway in NB at 1:7.5M."
   (INSTANCE-OF NB-HIGHWAY NB-NAT-HIGHWAY)
   (COMPOSED-OF H07002)
   (LOCATION NEW-BRUNSWICK)
   (NET-OF NATIONAL-HIGHWAY)
   (SCALE 7500000.0))
(DEFSCHEMA H07002
   "The whole stretch highway NB-2 at 1:7.5M."
   (IS-A BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
        HIGHWAY NATIONAL-HIGHWAY NB-2)
   (COMPOSED-OF H0700201 H0700202 H0700203 H0700204 H0700205
        H0700206)
   (CONSIST-OF)
   (EMR-CODE (107 4))
   (HAS-GEO-REP CH07001)
   (HIGHWAY-CLASS 1ST)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-2")
   (NATION-NAME CANADA)
   (OPERATION-STATUS OPERATIONAL)
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (SCALE 7500000.0)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (SUB-REP-OF H02002)
   (SUPER-REP-OF)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
(DEFSCHEMA H0700204
   (INSTANCE-OF BU-LAND-NETWORK BUILTUP-LAND GEO-COVER GEO-VIEW
        H07002 HIGHWAY NATIONAL-HIGHWAY NB-2)
   (ACCES NONE)
   (CONSIST-OF)
   (CONSTRUCTION-TYPE ROAD-AA)
   (EMR-CODE (107 4))
   (HAS-GEO-REP E07004)
   (HIGHWAY-CLASS 1ST)
   (LOCATION NEW-BRUNSWICK)
   (NAME "NB-2")
   (NATION-NAME CANADA)
   (OPERATION-STATUS OPERATIONAL)
   (PASSAGE NON-RAILED)
   (PHYSIC-FORM STRUCTURED-USED-AREA)
   (ROAD-DIRECTION BI-DIRECTIONAL)
   (SCALE 7500000.0)
   (STRUCTURE-FORM INTERCONNECTED-CONSTRUCTION)
   (SUB-REP-OF)
   (SUPER-REP-OF)
   (VIEW-OF GEOGRAPHIC-REALITY)
   (VIEW-OF-CLAS PHYSICAL-MATERIAL-ON-EARTH))
```

C.2 IMAS Topo-Objects

The following schema topo-objects are the geo-representations of the geographical objects listed in section III.1.

C.2.1 IMAS Topological Object 1:2,000,000

```
(DEFSCHEMA CH02001
   "Chain ch02001 is a geo-topo-representation-of NB2 1:2M."
   (IS-A CHAIN COMPLEX-OBJECT TOPO-OBJECT)
   (BEG-OBJECT JU02001)
   (BOUND-BOX (1952504 243530 2325661 120506))
   (CONSTRUCTED-BY CH0200101 CH0200102 CH0200103 CH0200104
        CH0200105 E02004 E020091 E020092 E020093)
   (END-OBJECT JU02010)
   (GEO-REP-OF H02002)
   (LEFT-POLYGON)
   (OBJ-CONNECTED-IN)
   (OBJ-CONNECTED-OUT)
   (OBJECT-FUNCTION TOPO-EDGE)
   (PART-OF)
   (RIGHT-POLYGON)
   (TOPO-STRUCTURE COMPLEX))
(DEFSCHEMA CH0200101
   (INSTANCE-OF CHAIN COMPLEX-OBJECT TOPO-OBJECT)
   (BEG-OBJECT JU02001)
   (BOUND-BOX (1952504 243530 2025701 221633))
   (CONSTRUCTED-BY E02001 E02002 E02003)
   (END-OBJECT JU02012)
   (GEO-REP-OF H0200201)
   (LEFT-POLYGON)
   (OBJ-CONNECTED-IN)
   (OBJ-CONNECTED-OUT)
   (OBJECT-FUNCTION TOPO-EDGE)
   (PART-OF CH02001)
   (RIGHT-POLYGON)
   (TOPO-STRUCTURE COMPLEX))
(DEFSCHEMA E02001
   "Trans Can Highway NB no.2 - 107 CLOCK W-E."
   (INSTANCE-OF BASIC-GEO-OBJECT EDGE TOPO-OBJECT)
   (BEG-NODE ND02001)
   (BOUND-BOX (1952504 243530 1967354 236878))
   (DIMENSION 1)
   (END-NODE N-95)
   (GEO-REP-OF)
   (GR-DISPLAY POLY-LINE)
   (LEFT-AREA)
   (PART-OF CH0200101)
   (RIGHT-AREA)
   (TOPO-STRUCTURE SIMPLE)
   (X-VALUE (1952504 ..  1967354))
   (Y-VALUE (243530 ..  237013)))
(DEFSCHEMA E020034
   "NB7-4 Prim H NB no.7 N-S"
   (INSTANCE-OF BASIC-GEO-OBJECT EDGE TOPO-OBJECT)
```

```
   (BEG-NODE N-50)
   (BOUND-BOX (2209587 99179 2226463 92047))
   (DIMENSION 1)
   (END-NODE ND02021)
   (GEO-REP-OF)
   (GR-DISPLAY POLY-LINE)
   (LEFT-AREA)
   (PART-OF CH0200701)
   (RIGHT-AREA)
   (TOPO-STRUCTURE SIMPLE)
   (X-VALUE (2209587 ..  2226463))
   (Y-VALUE (99179 ..  93954)))
(DEFSCHEMA E02004
   "Trans Can Highway NB no.2 - 107 CLOCK W-E."
   (INSTANCE-OF BASIC-GEO-OBJECT EDGE TOPO-OBJECT)
   (BEG-NODE ND02012)
   (BOUND-BOX (2025311 221633 2063400 165051))
   (DIMENSION 1)
   (END-NODE ND02003)
   (GEO-REP-OF H0200202)
   (GR-DISPLAY POLY-LINE)
   (LEFT-AREA)
   (PART-OF CH02001)
   (RIGHT-AREA)
   (TOPO-STRUCTURE SIMPLE)
   (X-VALUE (2025580 ..  2063400))
   (Y-VALUE (221633 ..  165051)))
(DEFSCHEMA JU02001
   (INSTANCE-OF COMPLEX-OBJECT JUNCTION TOPO-OBJECT)
   (BEG-OBJECT-OF CH02001 CH0200101)
   (CONSTRUCTED-BY ND02001)
   (END-OBJECT-OF)
   (GEO-REP-OF)
   (OBJECT-FUNCTION TOPO-NODE)
   (TOPO-FORMAT FIX-TOPO-JUNCTION)
   (TOPO-STRUCTURE COMPLEX))
(DEFSCHEMA ND02001
   (INSTANCE-OF BASIC-GEO-OBJECT NODE TOPO-OBJECT)
   (BEG-NODE-OF E02001)
   (DIMENSION 0)
   (END-NODE-OF)
   (GR-DISPLAY GR-NODE)
   (PART-OF JU02001)
   (TOPO-FORMAT FIX-TOPO-NODE)
   (TOPO-STRUCTURE SIMPLE)
   (X-VALUE (1952504))
   (Y-VALUE (243530)))
(DEFSCHEMA JU02010
   (INSTANCE-OF COMPLEX-OBJECT JUNCTION TOPO-OBJECT)
   (BEG-OBJECT-OF)
   (CONSTRUCTED-BY ND02010)
   (END-OBJECT-OF CH02001 CH0200105)
   (GEO-REP-OF)
   (OBJECT-FUNCTION TOPO-NODE)
   (TOPO-FORMAT FIX-TOPO-JUNCTION)
   (TOPO-STRUCTURE COMPLEX))
(DEFSCHEMA ND02010
   (INSTANCE-OF BASIC-GEO-OBJECT NODE TOPO-OBJECT)
   (BEG-NODE-OF)
   (DIMENSION 0)
```

```
   (END-NODE-OF E020021)
   (GR-DISPLAY GR-NODE)
   (PART-OF JU02010)
   (TOPO-FORMAT FIX-TOPO-NODE)
   (TOPO-STRUCTURE SIMPLE)
   (X-VALUE (2325661))
   (Y-VALUE (222660)))
(DEFSCHEMA JU02012
   (INSTANCE-OF COMPLEX-OBJECT JUNCTION TOPO-OBJECT)
   (BEG-OBJECT-OF CH0200201)
   (CONSTRUCTED-BY ND02012)
   (END-OBJECT-OF CH0200101)
   (GEO-REP-OF)
   (OBJECT-FUNCTION TOPO-NODE)
   (TOPO-FORMAT FIX-TOPO-JUNCTION)
   (TOPO-STRUCTURE COMPLEX))
(DEFSCHEMA ND02012
   (INSTANCE-OF BASIC-GEO-OBJECT NODE TOPO-OBJECT)
   (BEG-NODE-OF E0200101 E02004)
   (DIMENSION 0)
   (END-NODE-OF E0200102 E02003)
   (GR-DISPLAY GR-NODE)
   (PART-OF JU02012)
   (TOPO-FORMAT FIX-TOPO-NODE)
   (TOPO-STRUCTURE SIMPLE)
   (X-VALUE (2025580))
   (Y-VALUE (221633)))
```

C.2.2 IMAS Topological Object 1:7,500,000

```
(DEFSCHEMA CH07001
   "Chain ch07001 is-a geo-topo-representation-of NB-2 1:7.5M."
   (IS-A CHAIN COMPLEX-OBJECT TOPO-OBJECT)
   (BEG-OBJECT JU07018)
   (BOUND-BOX (1946176 239965 2325358 116435))
   (CONSTRUCTED-BY E07001 E07002 E07003 E07004 E07005 E07005A)
   (END-OBJECT JU07023)
   (GEO-REP-OF H07002)
   (LEFT-POLYGON)
   (OBJ-CONNECTED-IN)
   (OBJ-CONNECTED-OUT)
   (OBJECT-FUNCTION TOPO-EDGE)
   (PART-OF)
   (RIGHT-POLYGON)
   (TOPO-STRUCTURE COMPLEX))
(DEFSCHEMA E07004
   "Trans Can Highway NB no.2 - 107"
   (INSTANCE-OF BASIC-GEO-OBJECT EDGE TOPO-OBJECT)
   (BEG-NODE ND07004)
   (BOUND-BOX (2070430 151784 2157353 116435))
   (DIMENSION 1)
   (END-NODE ND07005)
   (GEO-REP-OF H0700204)
   (GR-DISPLAY POLY-LINE)
   (LEFT-AREA)
   (PART-OF CH07001)
   (RIGHT-AREA)
   (TOPO-STRUCTURE SIMPLE)
```

```
   (X-VALUE (2070430 ..  2157353))
   (Y-VALUE (151784 ..  140922)))
(DEFSCHEMA JU07018
   (INSTANCE-OF COMPLEX-OBJECT JUNCTION TOPO-OBJECT)
   (BEG-OBJECT-OF CH07001 CH0700301 PO07001 PO07009 POQUEBEC)
   (CONSTRUCTED-BY ND07018)
   (END-OBJECT-OF CH0700302 PO07001 PO07009 POQUEBEC)
   (GEO-REP-OF)
   (OBJECT-FUNCTION TOPO-NODE)
   (TOPO-FORMAT FIX-TOPO-JUNCTION)
   (TOPO-STRUCTURE COMPLEX))
(DEFSCHEMA JU07023
   (INSTANCE-OF COMPLEX-OBJECT JUNCTION TOPO-OBJECT)
   (BEG-OBJECT-OF CH0700502 PONOVA)
   (CONSTRUCTED-BY ND07023)
   (END-OBJECT-OF CH07001 CH0700501 PONOVA)
   (GEO-REP-OF)
   (OBJECT-FUNCTION TOPO-NODE)
   (TOPO-FORMAT FIX-TOPO-JUNCTION)
   (TOPO-STRUCTURE COMPLEX))
(DEFSCHEMA ND07004
   "EQU to nd07012."
   (INSTANCE-OF BASIC-GEO-OBJECT NODE TOPO-OBJECT)
   (BEG-NODE-OF E07004 E07012)
   (DIMENSION 0)
   (END-NODE-OF E07003 E07011)
   (GR-DISPLAY GR-NODE)
   (PART-OF JU07004)
   (TOPO-FORMAT FIX-TOPO-NODE)
   (TOPO-STRUCTURE SIMPLE)
   (X-VALUE (2070430))
   (Y-VALUE (151784)))
(DEFSCHEMA ND07005
   "EQU to nd07013."
   (INSTANCE-OF BASIC-GEO-OBJECT NODE TOPO-OBJECT)
   (BEG-NODE-OF E07005 E07013)
   (DIMENSION 0)
   (END-NODE-OF E07004 E07012)
   (GR-DISPLAY GR-NODE)
   (PART-OF JU07005)
   (TOPO-FORMAT FIX-TOPO-NODE)
   (TOPO-STRUCTURE SIMPLE)
   (X-VALUE (2157353))
   (Y-VALUE (140922)))
(DEFSCHEMA ND07018
   "EQU to nd07001 of nbrds7m.art"
   (INSTANCE-OF BASIC-GEO-OBJECT NODE TOPO-OBJECT)
   (BEG-NODE-OF E07001 E07017)
   (DIMENSION 0)
   (END-NODE-OF E07018)
   (GR-DISPLAY GR-NODE)
   (PART-OF JU07018)
   (TOPO-FORMAT FIX-TOPO-NODE)
   (TOPO-STRUCTURE SIMPLE)
   (X-VALUE (1946176))
   (Y-VALUE (239550)))
(DEFSCHEMA ND07023
   (INSTANCE-OF BASIC-GEO-OBJECT NODE TOPO-OBJECT)
   (BEG-NODE-OF E07023)
   (DIMENSION 0)
```

```
(END-NODE-OF E07005A E07022)
(GR-DISPLAY GR-NODE)
(PART-OF JU07023)
(TOPO-FORMAT FIX-TOPO-NODE)
(TOPO-STRUCTURE SIMPLE)
(X-VALUE (2325358))
(Y-VALUE (215369)))
```

Appendix D

Rules for Automatic IMAS Construction

The following rules are the basic rules used to execute the construction of IMAS objects as shown in Figure 4.5 and Figure 4.6.

D.1 Object Classification Hierarchy

The object classification hierarchy comprises both a geographical object and a topological object hierarchy. The following rules provide the set memberships for each class.

D.1.1 Geographical Object Classification Hierarchy

```
(defrule GET-INIT-DESCEND-GEO-VIEW
   "Get all descendants from the TOP of Geo-Object Hierarchy."
   (schema ?geo-view)
   (schema ?geo-view-obj
       (is-a ?geo-view))
   (Top-Hierarchy ?geo-view) ; FACT-CONTROL TopHierarchy.
   (Construct-Mode Obj-Hierarchy) ; FACT-CONTROL operation.
=>
    (if (not (slotp ?geo-view-obj scale))
        then (if (not (slotp ?geo-view-obj location))
                 then (assert (DESCEND ?geo-view-obj))
                 else (bind ?location (get-schema-value ?geo-view-obj location))
                      (assert (GENERIC-OBJ ?LOCATION ?geo-view-obj))
```

```
                    (for ?feature in-schema-parents-of ?geo-view-obj do
                        (assert (FEATURE-OBJ ?feature))
                    )
            )
       else
            (for ?obj in-schema-parents-of ?geo-view-obj do
                (bind ?scale (get-schema-value ?geo-view-obj scale))
                (bind ?location (get-schema-value ?geo-view-obj location))
                (assert (GEN-OBJ-SCALE ?LOCATION ?SCALE ?geo-view-obj))
            )
    )
)
(defrule INIT-GEOVIEW-HIERARCHY
   "Initial composition of Geo-View hierarchy."
   (schema ?geo-view-kinds)
   ?geo-hier <- (DESCEND ?geo-view-kinds)
   (Construct-Mode Obj-Hierarchy) ; FACT-CONTROL operation.
=>
    (for ?sub-geo-view-kinds in-schema-children-of ?geo-view-kinds do
        (if (NOT (slotp ?sub-geo-view-kinds location))
            then (assert (schema ?geo-view-kinds
                             (composed-of ?sub-geo-view-kinds))
                 )
        )
    )
    (retract ?geo-hier)
);
```

D.1.2 Topological Object Classification Hierarchy

```
(deffacts init-bas-geometrical (bas-geometric un-defined BASIC-GEO-OBJECT))

(defrule INIT-BASIC-GEOMETRIC-COMPOSITION
   "Set the initial hierarchy of basic topo-object."
   (schema ?basic-geo-object)
       (bas-geometric UN-DEFINED ?basic-geo-object)
   (Construct-Mode GeoMet-Obj-Class) ; FACT-CONTROL.
=>
    (for ?sub-basic-rep in-schema-children-of ?basic-geo-object do
        (assert (schema ?basic-geo-object
                     (composed-of ?sub-basic-rep))
        )
    )
);

(defrule GET-BASIC-GEOMETRIC-OBJ-HIERARCHY
   "Get all descendants of basic topo-object classifications."
   (schema ?basic-geo-object)
   (schema ?sub-bas-geo-obj
       (is-a ?basic-geo-object))
   (bas-geometric UN-DEFINED ?basic-geo-object)
   (Construct-Mode GeoMet-Obj-Class) ; FACT-CONTROL.
=>
    (for ?bas-geo-dim in-schema-children-of ?sub-bas-geo-obj do
        (if (is-a-p ?bas-geo-dim ?sub-bas-geo-obj) then
            (assert (schema ?sub-bas-geo-obj
                        (composed-of ?bas-geo-dim))
            )
```

```
          )
      )
);

(deffacts init-geotopo-object (bas-geotopo un-defined COMPLEX-OBJECT))

(defrule INIT-BAS-GEOTOPO-COMPOSITION
   "Set initial geo-topo representations of complex representations."
   (schema ?complex-object)
   (bas-geotopo UN-DEFINED ?complex-object)
   (Construct-Mode GeoMet-Obj-Class) ; FACT-CONTROL.
=>
    (for ?sub-geotopo in-schema-children-of ?complex-object do
         (assert (schema ?complex-object
                     (composed-of ?sub-geotopo))
         )
    )
);

(defrule GET-GEOTOPO-OBJ-HIERARCHY
   "Get all descendant of complex object representations."
   (schema ?complex-object)
   (schema ?sub-geotopo
       (is-a ?complex-object))
   (bas-geotopo UN-DEFINED ?complex-object)
   (Construct-Mode GeoMet-Obj-Class) ; FACT-CONTROL.
=>
    (for ?geo-topo-obj in-schema-children-of ?sub-geotopo do
         (if (is-a-p ?geo-topo-obj ?sub-geotopo) then
             (if (slot-null ?geo-topo-obj geo-rep-of) then
                 (assert (schema ?sub-geotopo
                             (composed-of ?geo-topo-obj))
                 )
             )
         )
    )
);
```

D.2 Pre-processing of Nodes and Edges

The following rules show the patterns in the LHS and its actions in the RHS for edges which have an empty slot "bound-box". The next rule shows the deletion of duplicate edges.

```
(defrule FIX-EDGES-BOUND-BOX
   "Fix EDGE's bounding-box."
   (declare (salience 7500))
   (schema ?object-edge
       (instance-of EDGE)
       (x-value ?x-val)
       (y-value ?y-val)
       (bound-box))
   (?object-edge INITIATED)
```

```
   (Operation-Mode CONSTRUCTION fix-topo-obj) ; FACT-CONTROL.
=>
    (bind ?b-box (def-bound-obj-edge ?object-edge))
    (assert (schema ?object-edge
                (bound-box ?b-box))
    )
);

(defrule REPLACE-OBJ-REF-TO-EQ-FIX-EDGE
   (schema ?edge-1
      (instance-of EDGE))
   (schema ?edge-2
      (instance-of EDGE))
   (fix-edge ?edge-1)
   ?d-edge < - (duplicate ?edge-1 ?edge-2)
   (Operation-Mode Pre-Process-Obj)
=>
   (if (NOT (slot-null ?edge-2 geo-rep-of)) then
       (for ?obj-rep in$ (get-schema-value ?edge-2 geo-rep-of) do
           (if (NOT (member$ ?obj-rep (get-schema-value ?edge-1 geo-rep-of)))
               then (assert (schema ?edge-1
                                (geo-rep-of ?obj-rep))
                    )
           )
       )
   )
   (if (NOT (slot-null ?edge-2 part-of)) then
       (for ?obj-rep in$ (get-schema-value ?edge-2 part-of) do
       (if (NOT (member$ ?obj-rep (get-schema-value ?edge-1 part-of)))
          then (assert (schema ?edge-1
                           (part-of ?obj-rep))
               )
          )
       )
   )
   (retract (schema ?edge-2) ?d-edge)
);
```

D.3 Processing Topological Relations

Processing topological relations begins with a single scale topological construction, and finishes interscale links between neighboring scales.

D.3.1 Single Scale Relations

```
(defrule FIX-CHAIN-BOUND-BOX
   "Fix CHAIN's bounding-box based on its sub-elements."
   (declare (salience 7450))
   (schema ?chain-objects)
   (schema ?geo-objects&~?chain-objects
       (part-of ?chain-objects)
       (bound-box ?b-box))
```

```
   (?chain-objects INITIATED)
   (Operation-Mode CONSTRUCTION fix-topo-obj) ; FACT-CONTROL.
   (test (OR (is-a-p ?chain-objects CHAIN)
        (instance-of-p ?chain-objects CHAIN)))
=>
    (if (slot-null ?chain-objects bound-box)
       then (assert (schema ?chain-objects
                      (bound-box ?b-box)))
       else (bind ?curr-obj-box (get-schema-value ?chain-objects bound-box))
            (bind ?new-obj-box (fix-object-window ?curr-obj-box ?b-box))
            (modify-schema-value ?chain-objects bound-box ?new-obj-box)
    )
);

(defrule COMPOSE-CHAIN-OF-BASIC-OBJ
   "COMPOSE basic-obj part-of CHAIN - ordered from beg-CHAIN."
   (declare (salience 7400))
   (schema ?CHAIN-OBJECT
       (beg-object ?beg-object)
       (end-object ?end-object)
       (constructed-by $?cons-obj))
   (schema ?e-construct (part-of ?end-object))
   (schema ?geo-object
       (beg-node ?b-node-obj)
       (end-node ?e-node-obj)
       (bound-box ?b-box)
       (part-of ?CHAIN-OBJECT))
   (?CHAIN-OBJECT INITIATED)
   ?compose < - (compose ?CHAIN-OBJECT ?geo-object); OBJECT Compositions.
   ?comp-b < - (chain-bas ?CHAIN-OBJECT ?cur-node) ; TRAVERSE PTr.
   (Operation-Mode CONSTRUCTION fix-topo-obj) ; FACT-CONTROL.
   (test (OR (is-a-p ?chain-object CHAIN)
             (instance-of-p ?chain-object CHAIN)))
   (test (OR (equal ?b-node-obj ?cur-node)
             (equal ?e-node-obj ?cur-node)))
=>
    (if (= (length$ ?cons-obj) 0)
       then (assert (schema ?CHAIN-OBJECT (constructed-by ?geo-object)))
       else
            (bind ?new-construct (append ?geo-object ?cons-obj))
            (retract (schema ?CHAIN-OBJECT (constructed-by $?cons-obj)))
            (assert (schema ?CHAIN-OBJECT (constructed-by $?new-construct)))
    )
    (if (equal ?b-node-obj ?cur-node)
       then (bind ?cur-node ?e-node-obj)
       else (bind ?cur-node ?b-node-obj)
    )
    (if (equal ?cur-node ?e-construct)
       then (assert (?end-object INCOMPLETE))
       else (assert (chain-bas ?CHAIN-OBJECT ?cur-node))
    )
    (retract ?comp-b)
    (retract ?compose)
);
```

D.3.2 Interscale Links

```
(defrule CONNECT-LOW-TO-HIGH-RESOLUTION
```

```
   "Connect low-resolution to high-resolution objects."
   (schema ?generic-obj)
   (schema ?generic-obj-1
       (is-a ?generic-obj)
       (scale ?scale-1))
   (schema ?generic-obj-2
       (is-a ?generic-obj)
       (scale ?scale-2))
   (generic-obj ?location ?generic-obj)
   (low-high ?scale-1 ?scale-2)
   (Operation-Mode CONNECT-SCALE-OBJ Map-Display) ; FACT-CONTROL.
=>
    (modify-schema-value ?generic-obj-1 sub-rep-of ?generic-obj-2)
);

(defrule CONNECT-HIGH-TO-LOW-RESOLUTION
   "Connect high-resolution to low-resolution objects."
   (schema ?gener-obj)
   (schema ?gener-obj-1
       (is-a ?gener-obj)
       (scale ?scal-1))
   (schema ?gener-obj-2
       (is-a ?gener-obj)
       (scale ?scal-2))
   (generic-obj ?location ?gener-obj)
   (high-low ?scal-1 ?scal-2)
   (Operation-Mode CONNECT-SCALE-OBJ Map-Display) ; FACT-CONTROL
=>
    (modify-schema-value ?gener-obj-1 super-rep-of ?gener-obj-2)
);
```

VITA

Candidate's full name : Agus Prijadi Saido
Place and date of birth : Kediri, Jawa Timur, Indonesia,
May 1, 1955.
Permanent address : Jalan Pinang 12 No. 1 - Cemani,
Surakarta, Jawa Tengah,
Indonesia.
Schools attended : Elementary School
SD Sukorame I, Kediri, Indonesia,
1967.
: Junior High School
SMPN IV, Kediri, Indonesia,
1970.
: Senior High School
SMAN Nganjuk, Indonesia,
1973.
Universities attended : Department of Surveying Engineering
Bandung Institute of Technology (ITB)
Bandung, Indonesia,
Ir. Surveying Engineering, 1980.
Publication : "Doppler Translocation Problems."
Ir. Thesis, ITB, 1980.