INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

CONSISTENT GENERALIZATION OF SPATIAL DATA ACROSS THE INTERNET

By Jianxue Huang

**A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of**

Master of Arts

Department of Geography

**Carleton University
Ottawa, Ontario
September, 3, 1998**

**© copyright
1998, Jianxue Huang**

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-36827-0

**The undersigned recommend to the Faculty of Graduate Studies
and Research acceptance of the thesis**

**CONSISTENT GENERALIZATION OF SPATIAL DATA
ACROSS THE INTERNET**

**Submitted by Jianxue Huang
in partial fulfillment of the requirements for
the degree of Master of Arts**

Thesis Supervisor

Chair, Department of Geography

Carleton University

Sept. 3, 1998

ABSTRACT

Spatial data generalization across the Internet is a challenge. The issues and problems require study and exploration. Several important aspects of this topic are involved in this research. The purposes, characteristics and methods of generalization both in conventional cartography and in a digital environment are discussed. The process of spatial data access and display in an Internet context is investigated. The method and the considerations for carrying out generalization over the Internet are explored. Based on those discussions and explorations, the research focuses on the development of a mechanism (the GNT system) which is used to deal with inconsistencies between adjacent data sets introduced in the generalization processes. The introduction of inconsistency is a serious problem in spatial data generalization, especially in an Internet context where the users' ability to interact with a server's GIS to deal with the problems is very limited. The GNT system helps users to get rid of inconsistencies between data sets introduced in generalization and so obtain compatible results. The development and the implementation of the GNT system are described in detail. A test is conducted to verify the concept of the GNT system and to test the formulas and algorithms involved in the system.

ACKNOWLEDGMENTS

I would first like to give my thanks to my supervisor, Dr. D. R. F. Taylor for his guidance and encouragement throughout the M.A. program.

I would also like to give my thanks to Dr. Dianne Richardson, the Head of Methods Development Section, Canada Centre for Remote Sensing (CCRS), Natural Resources Canada, and adjunct advisor of the M.A. thesis, Dr. E. M. Siekierska, the Manager of Mapping Services Branch, Geomatics Canada, Earth Sciences Sector, Natural Resources Canada, and adjunct advisor of the M.A. thesis, for their guidance and support to the research topic.

Grateful thanks to the Department of Geography, Carleton University for providing me with the opportunity for my M.A. program. Special thanks to Ms. H. Anderson for her kind help during the past two years.

I also wish to thank CCRS and Geomatics Canada for providing research and additional funding support. The thesis was completed under the aegis of the Memorandum of Understanding between the Department of Geography of Carleton University and Natural Resources Canada for research cooperation. Special thanks to Dr. R. Thomson for his reading of many chapters of the thesis and giving me valuable feedback. Thanks also go to K. Kushigbor and J. H. Lefebvre for their help during my work on the thesis in CCRS.

Special thanks go to my friend Aining Zhang for her help in reviewing Chapter 3 of the thesis and for her valuable discussion of the topics relating to the research.

TABLE OF CONTENTS

ACCEPTANCE SHEET	ii
ABSTRACT	iii
ACKNOWLEDGMENTS	iv
TABLE OF CONTENTS	v
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Motivation	2
1.2.1 GIS technology	2
1.2.2 Internet technology	4
1.3 Objectives and methodology	5
1.4 Organization of the thesis	6
CHAPTER 2 THE CONCEPT OF GENERALIZATION	9
2.1 Generalization in conventional cartography	9
2.1.1 The definition and processes of generalization	10
2.1.2 The nature of generalization	13
2.2 Generalization in a GIS environment	15
2.2.1 The characteristics of generalization in a GIS context	16
2.2.2 The needs and purposes	17
2.2.3 Database generalization (model-oriented generalization)	19

2.2.4 View generalization	23
2.2.5 The integrated view of generalization	25
2.2.6 GenSystem	27
CHAPTER 3 SPATIAL DATA ACCESS IN AN	
INTERNET ENVIRONMENT	30
3.1 The components of the Internet	31
3.1.1 The Internet and computer network	31
3.1.2 Protocols	32
3.1.3 The World Wide Web (WWW)	35
3.2 The access and manipulation of geo-data across the Internet	37
3.2.1 Why Internet?	37
3.2.2 Static access to geo-data	38
3.2.3 Dynamic access to geo-data	40
3.2.4 The concept of Common Gateway Interface (CGI)	42
3.2.5 Java approach	44
3.2.6 ActiveX approach	46
3.3 Spatial data display	47
3.3.1 The raster method.....	48
3.3.2 The vector method	51
3.4 Data format conversion	54
3.5 Generalization across the Internet	57

3.5.1	Considerations in generalization applications	58
3.5.2	The problems introduced in generalization implementation	59
3.5.3	A model for bringing a generalization system into the Internet	60
CHAPTER 4 THE DESIGN OF A MECHANISM FOR		
	CONSISTENT GENERALIZATION	63
4.1	The need for the mechanism	64
4.2	The methodology	68
4.3	The adjacent-relationship reference files	77
4.3.1	The definition of the T-T relationship	78
4.3.2	The creation of the T-T relationship file	79
4.3.3	The definition and the determination of the S-S relationship	82
4.4	Generalization track (GNT) files	85
4.4.1	The creation of the borderline nodes	86
4.4.2	The index matching method	95
4.4.3	The coordinate matching method	97
4.4.4	The distance matching method	100
4.4.5	Generalization history information	102
4.5	The method of implementing the GNT system	103
4.6	The analysis of the GNT file system	109
CHAPTER 5 TEST AND IMPLEMENTATION		114
5.1	The testing environment and methodology	114

5.2 The creation of the GNT file	115
5.3 The implementation of the GNT file system	121
5.4 Discussion	125
CHAPTER 6 SUMMARY AND CONCLUSION	131
BIBLIOGRAPHY	137
APPENDIX A: THE C PROGRAM FOR CREATING	
BORDERLINE NODES	146
APPENDIX B: THE INFO PROGRAM FOR IMPLEMENTING	
THE GNT SYSTEM	149

CHAPTER 1. INTRODUCTION

1.1 Background

Generalization is one of the central concepts in spatial data processing. Geographic information, regardless of whether it is in an analog form (such as a map) or in a digital form (such as a data set in a GIS), comes from an abstraction of the real world. Generalization is just such an abstraction process which removes and suppresses the less important information for a particular purpose, and emphasizes the more important aspects of the phenomenon. The term generalization is mostly used in cartography, where it is a critical concept and a core task for dealing with communication issues (including complexity reduction and the presentation arrangement) in the processes of map design and map compilation. In the meanwhile, as pointed out by Brassel and Weibel (1988), generalization is not only used in cartography; it represents a basic human intellectual activity and part of the general scientific process. In digital data processing, generalization is also a key element. It is used to deal with the complexity reduction of data both in information content and in data presentation. In recent decades, automated generalization has become one of the most difficult challenges in the research field of GIS and computer-assisted cartography.

With the development of computer network technology, data (both spatial data and non-spatial data) access and dissemination across the Internet have experienced

rapid growth. Today, the new development of Internet associated browsers allows users not only to access global data, but also to interact with various remote server applications. Some GIS-like operations (such as spatial retrieval, spatial searching, zooming in and zooming out, etc.) can now be performed across the Internet. Through the World Wide Web service of the Internet, users can even employ the data maintained on the server host computers to design maps in real-time.

However, the implementation of automated spatial data generalization across the Internet remains a challenge. The possibility, problems and issues of generalization implementation through the Internet require study and exploration.

1.2 Motivation

The development and application of new technology have greatly promoted geo-spatial data collection, processing, distribution and application. Generalization, which is used as an essential process for data abstraction and communication, is facing increasing demands and challenges.

1.2.1 GIS technology

Geographic information systems (GIS) is a rapidly evolving discipline. GIS technology has been extensively used to handle geo-spatial data. A GIS can be identified as a database system in which every object has a precise geographical location, and upon which a set of procedures can be operated in order to perform functions of input,

management, analysis and output (Maguire, 1991; Goodchild, 1995; Smith *et al.* 1987). GIS technology provides great powers and facilities to the user for spatial data processing and application. Using GIS, a large amount of data can be handled in ways not previously possible.

Geo-spatial data are scale related. Every data set maintained in a GIS corresponds to a certain spatial resolution and accuracy. To meet the demands for data at different scales, there are two possible solutions. One is to create and maintain several scale-dependent data sets so that they can be used for the corresponding applications respectively. The other solution is to generate a master database, and at the same time, develop and provide a generalization tool-kit to the users. Using the generalization tools, users can select the target scales and carry out generalization by themselves. Thus, the data which are at expected scales can be produced from the source data. The first method is simple and has been used by many national mapping agencies (Muller *et al.* 1995). This method has some serious drawbacks. Firstly, it is expensive to regenerate new databases and renew those databases both in money and in time. Secondly, the agencies have to maintain a large amount of duplicate data. Those data take up a very large storage space. And lastly, inconsistency is easily introduced into the databases. The second method is an appealing solution. It has increasingly been brought to people's attention in recent years and will be the focus of this thesis.

1. 2. 2 Internet technology

The Internet is a computer-network technique. It can be roughly defined as a global network of local computer networks. Through the Internet (the World Wide Web service of it in particular), data can be disseminated globally. Users can access Web information from computers that are located anywhere and run any operating system as long as they are connected to the Internet. Since geographic data are communicated visually and often digitally, the Web may be the perfect environment to distribute spatial data (Limp 1997).

As mentioned previously, a GIS equipped with a generalization system will provide extensive facilities to users. The use of the Internet makes it possible that users at a global scale can implement the generalization system through the computer network. This allows them to carry out generalization and get appropriate data packages that best fit their applications without needing to equip their GIS with a generalization system.

An important issue in data generalization is to keep the geographically adjacent data sets (coverages) consistent with each other, so that the user can join the adjacent coverages together without incurring any errors. To eliminate the inconsistencies between the coverages which arise from the generalization, a technique needs to be developed. It will allow users to generalize adjacent coverages separately, yet to get compatible results. This will be developed in this thesis and discussed in detail in Chapter 4.

1.3 Objectives and methodology

The objectives of this research include the following:

- Identifying the concepts of generalization both in conventional cartography and in a digital context. Exploring the functions, characteristics and issues of generalization in a GIS environment.
- Discussing the mechanism of spatial data access and transmission in an Internet context. Exploring the possibility, the method and the problems of carrying out spatial data generalization across the Internet.
- Developing a mechanism (system) for dealing with the inconsistency which is introduced in the generalization processes. This includes: defining and creating the neighbour relationship between geographically adjacent data tiles; designing and establishing generalization track file; designing the mechanism and algorithms for creating the matching between map objects; and implementing the system to adjust the generalization results. This mechanism will enable the generalized coverages to be compatible with each other no matter what time those coverages are generalized and in what order they are generalized. The development of this mechanism will be the core contribution of this thesis.

- Testing the mechanism. The purpose of the test is to verify the concept of the mechanism and to test the algorithms that are developed in this research. The test will be done in a system environment.

Spatial data generalization across the Internet is a broad topic in which not only the issues of automated generalization, but also the Internet methods are involved. It is impossible for a thesis to cover all aspects of this topic. Therefore, the goal of this research is not to develop a whole generalization application for the Internet, but to focus on the issues of geo-spatial data handling. In this study, the properties and the issues of generalization in a digital context, the characteristics of the spatial data access over the Internet and the considerations for bringing an existing generalization system (GenSystem) into the Internet are first discussed. Then, the study will focus on developing a technique to deal with the problem of inconsistency introduced by carrying out generalization across the Internet.

1.4 Organization of the thesis

The thesis consists of six chapters. Those chapters are organized as following:

Chapter 1: As presented previously, an outline of the thesis is given in this chapter. It includes: the background of the research, the motivation (why the research should be done), the objectives and the methodology (what will be done in the research and how to do it), and the structure of the thesis.

Chapter 2: The objective of this chapter is to review the development of generalization, identify the concepts, and explore the purposes, the characteristics and the issues of generalization both in conventional cartography and in a digital environment. The discussion mainly focuses on the digital generalization. The concepts and the issues of model-oriented generalization (database generalization) and view-oriented generalization will be discussed in some detail. A generalization system, named GenSystem, is briefly described in this chapter.

Chapter 3: This chapter discusses the Internet. The relevant components of the Internet which relate the spatial data access and dissemination are summarized. The mechanism of spatial data access through the Internet is discussed. It includes: the static and dynamic access of geo-spatial data, the new techniques used to handle spatial data (such as CGI, Java, ActiveX), the display of spatial data and data format conversion. The possibility, method, and the problems of implementing a generalization system across the Internet are discussed. A model is proposed for bringing a generalization system into the Internet.

Chapter 4: In this chapter, a mechanism for consistent generalization is developed. Firstly, the problems are identified and the methodology for solving the problems is outlined. Then, the development of the mechanism is described in detail. It includes: defining the adjacent relationship between data sets (coverages); creating generalization track files which are used as the references to the generalization of adjacent coverages; creating the matchings between the map objects; and lastly, adjusting

the generalization results. An analysis of the mechanism is conducted at the end of this chapter.

Chapter 5: In this chapter, a test is conducted. Two tiles that are geographically adjacent to each other are used for the test. The process includes: creating the GNT file for each tile, creating the matchings between the map objects on two tiles, and adjusting the generalization results. ARC/INFO and C language are used to do the test.

Chapter 6: This chapter is the summary and the conclusion of the research. Some related topics or aspects which need further study will be presented in this chapter.

CHAPTER 2. THE CONCEPT OF GENERALIZATION

Generalization is a general scientific method which is used to deal with the issues of the abstraction of reality. In cartography, generalization is a central process. It deals with reducing complexity and the graphical representation of cartographic features. In a digital environment, the issues of data handling are significantly different from those in traditional mapping. Generalization, in this context, has obtained a broader meaning. Many new concepts have been introduced. In the meantime, the map view is still an important component of a GIS. Therefore, in a GIS context, the principles of generalization created in traditional cartography remain fundamental. Many techniques used in conventional map generalization are still useful and necessary. In this chapter, the functions and the characteristics of generalization both in conventional cartography and in a digital context will be explored.

2.1 Generalization in conventional cartography

Maps are a kind of carrier of geographic information. The main functions of the maps are their uses in communication and analysis. A map is a simplified abstraction of the real world. On a map, only those features which are critical or important to the purpose are selected for representation. As compared with the phenomena to be

mapped, all maps are scale-reduced because no one map, no matter how large the scale is, can show the exact detail of the real world. From this perspective, maps can be considered as the results of a series of generalization processes.

2. 1. 1 The definition and processes of generalization

Generalization is one of the central concepts in cartography. It can be defined as the process of reducing the amount of detail so that the character or essence of the original features is retained at successively smaller scales (Robinson 1995). Viewed from its operational aspect, generalization is about 'the selection and simplified representation of detail appropriate to the scale and /or the purpose of the map' (ICA 1973). As a key element of cartography, generalization is involved in both the process of map design and the process of map compilation.

When a new map is designed, the first issue for the designer is to determine the theme of the map. The features which are important to the presentation of the theme will be selected for representation via the map. This is an abstraction process. Through this process, cartographers can create an abstract model of reality. This model is much simpler than the real world both in complexity and in detail. In order to portray those selected features clearly at a reduced magnitude, a series of elaborately designed symbols is used for the representation.

The process discussed above is usually considered as the first generalization of the map. In this process, cartographers accomplish the abstraction of reality to a map.

Two types of abstraction activities are involved in this process: selection and symbolization. The selection process focuses on the choice of map contents. Through this process, cartographers can generate a simplified view of the real world. Symbolization is the process in which the geographic features are abstracted as a set of symbols for the purpose of presentation. The symbols are regarded as the language of maps -- a kind of graphic language.

When a larger (or a more general) view is required, the scale of the map has to be reduced. With the reduction of the map scale, the images of features can become too crowded to be identified. In this case, further generalization work is unavoidable. Cartographers must adjust the contents of the map and re-arrange the symbols so it is a recognizable representation of the phenomena. This process can be considered as two stages: the first stage entails deciding what features to select and deciding the level of semantic detail or resolution; the second stage is to arrange a proper representation of the chosen features on the map (Thomson and Richardson 1996). A number of rules and techniques for cartographic generalization have been developed. The methods most in the use of generalization practices are described below. In a specific application, a selection of these may be involved.

- *Selection/Elimination:* Usually, selection/elimination is the first activity in the process of map design and map compilation. It serves to decide which features will be depicted on the map and how in detail they will be portrayed. This process is totally dominated by the scale and the purpose of the map.

- *Classification:* The goal of classification is to order and group features by their attribute values. Some qualitative characters of features may be lost in this process. The low level classes are merged or regrouped into higher level classes. This technique is largely employed in the generalization of thematic maps.
- *Simplification:* Simplification is a general term. It means the reduction of complexity. Any process which works for this purpose can be considered as simplification. In a focused meaning, it implies the simplification of graphic symbols, such as simplifying the shape of line features. Using simplification techniques, cartographers can retain and enhance the important characters of features and eliminate unwanted details.
- *Amalgamation:* This is a process that aggregates some small individual elements into a large one, so that the general characteristics of features can be retained and enhanced by eliminating some trivial ones.
- *Collapse:* A collapse technique is employed to reduce the dimensions of the symbols. As the scale decreases, some features cannot be portrayed in their original symbols. In this case, the lower dimension symbols will be used for the representation. For example, a lake collapses into a line (or a point) on the map.
- *Exaggeration:* With the decrease of the map scale, some features (or some characteristics of the features) become too small (or too trivial) to be identified. In order to retain the important ones, we will deliberately increase their spatial extent in order to capture their real-world essence. This process is termed exaggeration.

- *Refinement:* In many cases, some features need to be re-aligned because they are either too numerous or too detailed to show to scale. The refinement process allows us to maintain and enhance the general spatial pattern of the features, and to get rid of the trivialities.
- *Displacement:* When the map scale is reduced, the portrayal of some features may be in conflict with each other, or their spatial relationships can no longer be identified. In order to show their spatial relationships correctly and clearly, some features need to be shifted from their true locations. This process is termed displacement.

The techniques described above provide the normal means for us to carry out the generalization processes. Due to the complexity and the diversity of the problems in generalization practice, it is difficult to exhaust the list of methods. In practice, some additional methods may be adopted to deal with the specific problems.

2. 1. 2 The nature of generalization

In conventional cartography, the information is processed and maintained in an analog form. The storage and the display of information are tied together. They cannot be separated at any time. Due to this property, a generalization activity will affect the storage and the display of the information at the same time. This is so liable to cause confusion that quite a few people consider the visual pattern as the overall purpose of the generalization.

As described in section 2. 1. 1, the generalization activities involved in the process of map compilation can be divided into two stages: content processing and representation arrangement. Content processing is the model-oriented aspect of generalization. It is not simply to reduce the map content to the scale, but is a more intellectual process that closely relates to the mapping purpose and intended readers. The orientation of content processing is to reduce the resolution of geographic information for enhancing its general character, and create an appropriate model of reality corresponding to the target map scale. Content processing may be divided into quantitative generalization and qualitative generalization (Ratajski, 1967). Quantitative generalization consists of a gradual reduction in quantity of the features. Qualitative generalization will result in discarding some kinds of features, removing some classes or regrouping them into more abstract classifications. Selection/elimination, classification and some additional techniques are usually employed for content processing.

The representation arrangement is the visualization-oriented aspect of generalization. It aims to arrange a legible and appropriate representation of the map contents for communication purposes. Perhaps due to the strong visualization view in conventional cartography, the model-oriented aspect of generalization has been neglected to a certain degree in the literature. Actually, the model-oriented consideration is critical to the practicability and the scientific value of maps, especially to thematic maps. It is an important aspect of conventional map generalization.

Generalization is a complex process. In this process, the problems should be

considered in an integrative way and dealt with in a comprehensive manner. Though in a large body of literature, the processes of generalization are often analytically decomposed into a series of operational procedures, we should be aware that these parameters are interrelated and not entirely separable (Brassel and Weibel 1988). Successful generalization processing is not the simple composition of some separated techniques. It should involve an understanding of the map purposes, the feature properties, and the geographic characteristics of the mapped area. It also should involve the knowledge about when the generalization needs to be carried out and how it is to be carried out. Therefore, generalization is a highly creative and subjective task. In a manual process, its execution requires expertise. A knowledgeable and experienced cartographer can apply the right methods in the proper situations to deal with the appropriate problems. Since human beings can consider the problems holistically, the generalization process in conventional cartography is a 'batch approach'.

2.2 Generalization in a GIS environment

GIS is a complex domain. Many disciplines are involved in the development and the application of GIS technology. As systems which handle geographic data, GIS has moved far beyond the paradigm of traditional paper map series. Generalization in a GIS context has also acquired a broader meaning than that involved in conventional cartography.

A GIS can be viewed from different aspects. Three views of a GIS, i.e., the map

view, the database view and the spatial analysis view are described by Maguire (1991). The map view focuses on the visualization aspect of the GIS; the database view emphasizes the spatial database design and implementation; the spatial analysis view stresses the system analysis and the data modeling. The three views are not independent, but interrelated or even overlapping each other. Viewing a GIS in such an analytic way is helpful for us to explore the purposes, functions and characteristics of generalization processes in a digital environment.

2. 2. 1 The characteristics of generalization in a GIS context

In a GIS context, the geo-spatial data are maintained in a digital form. Unlike analog maps, where the display and the storage of information are integrated with each other, in a digital context a strong separation is made between the information content of the system and the presentation of this information in graphic form (Morehouse 1995). Though the visual view is the most basic and efficient way for the communication between GIS and the end-users, the database can stand alone and can be implemented separately from the graphic display to a certain degree. This does not mean that the data storage and its display in a GIS are completely independent of each other (actually, no spatial database is designed in such a style which stands alone without visual display function). It only means that the relationships between data storage and display are no longer one to one, but one to more, i.e., for one data set, we can arrange many different displays by using different symbols, different methods (such as different layer

compositions, different map extents etc.) or different scales. Due to this character, the functions of generalization in a digital context can be separated to a certain degree: some generalization operations focus on the database and spatial analysis aspect, and others emphasize the visualization considerations.

In addition, the generalization operations in a digital world are implemented by computers. Since computers, in practice, only work by following a sequence of predetermined algorithms, the generalization processes are treated individually instead of in an holistic manner. To achieve a successful generalization, the entire process must be a simulation of the cartographer's visual analysis, decision-making, and actions (Lee, 1995). Therefore, the principles and the methods used for generalization in the digital world are similar to those used in conventional cartography. The geographic and cartographic knowledge required in the generalization processes must be formalized so that the computers can understand and follow it. This task is perhaps the most difficult challenge in automated generalization research.

2.2.2 The needs and purposes

The requirement for generalization, as for other GIS processes, is not only a technique issue. Some other factors, such as economic considerations, are also concerned. Those factors have been discussed by Muller (1991, 1995) and Spiess (1995). Here in this thesis, the discussion only focuses on the technique component.

Corresponding to the three aspects of the GIS, generalization can also be viewed

from those three aspects. Viewed from the aspect of database implementation, the generalization process can reduce the volume of data by eliminating the coordinates which are unnecessarily detailed for the visualization communication at the target scale and the analysis applications for the specific purposes. Through data reduction, the volume of the data sets can be reduced so that the memory space in the computer system can be greatly saved. This will not only benefit the GIS implementation (such as database maintenance, data access, and data processing), but also benefit the data transmission, especially the transmission through the Internet.

Viewed from the aspect of spatial analysis, some generalization processes can be used to remove unwanted information, reduce the resolution of spatial data (both in spatial and in semantic terms), and universalize the contents. This allows users to get a high level view of the phenomenon. Thus, the users can concentrate on the essential information and avoid the conceptual “noise” in their applications. This process can be considered as establishing a more abstract model of reality based on the existing one. In other words, it is the model-oriented generalization (in some literature, it is simply termed model generalization). In an operational perspective, this process can be considered as the extraction of a subset of data from the original set, or considered as the process of database to database transformation. From this idea, the objective of those generalization processes is to derive a new database which has different spatial / thematic / temporal resolution from the existing database (Peng 1997). Therefore, this process is also termed database generalization in some literature. More discussion about

database generalization will be given in the following section.

Viewed from the aspect of visualization, the task of generalization is to simplify the geometric characteristics (both in shape and in distribution pattern) of the images and arrange a proper representation. This allows the users to get an easy-to-read presentation, so as to avoid geometric “noise”. This process may be called view-oriented generalization or simply view generalization. It will be further discussed in section 2. 2. 4.

2. 2. 3 Database generalization (model-oriented generalization)

As discussed previously, the purpose of database generalization is to transform an existing database into a new one that possesses a lower resolution. From this idea, any process that aims to universalize the information contents can be considered as database generalization. Many techniques can be used for this purpose, of which the most commons are perhaps *selection* and *classification*. The *selection* operation serves to extract a subset of objects from the original object collection. The *classification* process aims to create a set of new groups of objects, in which only the general characteristics of the objects are maintained. Besides *selection* and *classification*, the technique *amalgamation* (for merging the same type of objects or combining different types of objects), *collapse* (for changing the spatial and attribute characters of the objects) and some others are also often involved in the generalization processes.

From the above discussion we can find that the purpose of database

generalization is quite similar to the content processing in conventional cartography. Both of these aim to deal with the issues of abstraction of reality. The abstraction process is, actually, the process of making decisions about which features should be retained in the new database and which ones should be eliminated. The decisions are made based on the evaluation of importance of features: the important features are retained; the less important ones will be eliminated or changed in some ways. To conduct the evaluation, the information which depicts the characteristics of the features must first be acquired. This process, i.e., the information acquisition or data enrichment, is the key step in database generalization.

The information required in the evaluation can be decomposed into two parts: the spatial characteristics of the features and their thematic attributes. Some spatial characteristics can be obtained by performing a series of spatial analyses. For example, through calculating and comparing the length of the rivers and the angles between river sections at junctions, the main streams of the river system can be determined; by conducting network analyses, the roads which are important in terms of topology can be selected from a road system. Richardson (1996) and Richardson *et al.* (1996) discussed methods and experiments in the automatic classification of hydrographic features, transportation network and land-cover.

The thematic attributes usually come from the database. They provide us with information about the qualitative characters of the features, such as the classification of the road system, the population of the cities, etc. The attribute data may be different

from one database to another (different term name, different meaning, different classification and so on). In order to deal with the data which come from different databases, an interface must be designed for extracting the common information from those data.

To perform database generalization, an appropriate data model which supports database abstraction is required. The data model is the backbone of a generalization system. It dominates the mechanism of the system and the process of the data abstraction. Oosterom *et al.* (1995) and Oosterom(1989, 1991) described a data model. The model is based on a hierarchical storage structure and designed for multi-scale GIS. The emphasis of the model is put on the selection and simplification techniques. The selection process is supported by a Reactive-tree, in which the object entries are stored at different levels according to their importance values. The selection operation, in this case, becomes the simple retrieval operation. Therefore, the generalization processing can be accomplished on-the-fly, and the users can get a very quick response.

The introduction of the object-oriented designing method provides a potential perspective to the geographic data modeling and generalization. As pointed out by Thomson and Richardson: 'the powerful data modeling techniques, based on an object-oriented approach, permit the thematic and spatial generalization of geographic data' (1996:446).

In an object-oriented environment, the object is the basic unit for operation. An object is the digital representation of a real world phenomenon (entity). It cannot be

subdivided into other objects of the same kind. A group of objects with similar behaviors is defined as a class of objects. Several related classes can be aggregated into a higher level class without losing their main inheritances.

The object-oriented approach emphasizes the definition of classes for objects, relationships between classes, and possible operations on classes and their members (Mark 1991). It supports data abstraction and character inheritance. These properties greatly facilitate geo-data modeling and the performance of generalization.

In the process of object-oriented design, the definition of classes is a key task. A class is an abstraction of the real world entities, and an object is the concrete representation of the class. Different approaches can be used to define the class (or object), depending on the purpose of the modeling and the characteristics of the geographic features. An object can be the representation of either a simple entity or a complex entity. Tang *et al.* (1996) described a feature-based model, in which the definition of the objects is based on simple geographic entities, such as a road, a river etc. The attributes (both the spatial attributes and the thematic attributes) and the relationships (spatial relationships and non-spatial relationships) are directly attached to the objects.

Buttenfield (1995) discussed an object-oriented multi-scale data model. The model is based upon the Digital Line Graph (DLG-E) data model developed at the US Geological Survey. It extends the DLG-E model to fit generalization uses. In this model, the object is the representation of a complex entity, such as an industrial

complex that may be composed of buildings, warehouses, cooling pond, transportation paths (roads, railways or water channels) and so on. The rules and the procedures (methods) which are used for generalization operations are bound to the objects. Hence the generalization processing can be directly performed on each object.

A generalization system, named GenSystem, has been developed at the Canada Centre for Remote Sensing (CCRS). GenSystem is designed for use in the abstraction process. It will be described in section 2.2.5.

2. 2. 4 View generalization

Visualization is a very important component of a GIS. Whether viewed from the spatial analysis aspect or from the map aspect, visual presentation is indispensable for a GIS. In the spatial analysis view, visual display is the most efficient medium for the communication between the system and users. As an interface, it serves information input and output. The input utility allows the users to send information to the system. The input can be performed by carrying out a series of interactive operations, such as spatial searching, locating, retrieval and so on. The output utility displays the results to the users.

Viewed from the mapping perspective, the graphic presentations are the end products of the map making processes. For this purpose, a cartographic database usually needs to be attached, in which various types of cartographic processes can be performed.

The need for view generalization comes from both the interface use and the mapping applications. The purposes of generalization for these are different in some respects, and the methods used may be different as well. For interface use, the purpose of generalization is to reduce the geometric complexity of the display for pursuing efficient communication and obtaining a legible representation of the result. Today, most GIS packages offer some generalization utilities for these purposes, such as line simplification and smoothing. For the purpose of map-publishing, all of the cartographic requirements must be taken into consideration. Those requirements include symbol design, graphic simplification, location and spatial relationship arrangement etc. This process is more like the generalization for the purpose of representation arrangement in conventional cartography. It is important to distinguish these two kinds of generalization (i.e., the generalization for interface use and the generalization for mapping purpose) because view generalization is often considered as purely cartographic in some literature.

View generalization is also a complex process. In this process, not only the locations, shapes and patterns of the features, but also the spatial relationships between the features should be processed. Research on view generalization has been undertaken for a long period of time. Before the middle of 1980s, research in generalization mainly focused on the simplification and smoothing of linear features. This is perhaps because eighty percent of cartographic objects are perceived to be lines (in fact, many of them are polygons) (Muller *et al.* 1995). Lots of algorithms have been developed for this

purpose. The most popular one is perhaps the Douglas-Peucker algorithm (Douglas *et al.* 1973). The Douglas-Peucker algorithm uses the deviation from a centerline as the criterion for identifying critical points and allows their identification in a global view. This algorithm and some others are also discussed by McMaster (1989a and 1987).

Moreover, the techniques, such as aggregation, exaggeration, elimination, displacement and some others which are used in conventional cartography are often involved in the process of view generalization.

2. 2. 5 The integrated view of generalization

In computational algorithm design, a strategy, termed divide-and-conquer, is widely used to deal with the complex problems. A complex problem can be analytically decomposed into several sub-problems by removing some constraints. Those sub-problem are easier to solve. We can get the solution of the original problem by solving those relaxation problems. This approach is also adopted to deal with complex problems in digital generalization. A good example is that a generalization process is analytically decomposed into different views and a series of sub-processes. The adoption of such an approach is helpful for us to explore the characteristics and define the issues of generalization, and then, to solve the problems step by step. Thus, a comprehensive task in conventional map generalization becomes a series of relatively simple routines in a digital context, so that computers can process them. However, we should be aware that those different aspects of generalization are not completely

independent from each other. In fact, the generalization processes corresponding to the different views are interrelated, or even overlap each other: generalization processes for storage-reduction may include content reduction and graphical simplification — database generalization and view generalization; database generalization is normally the first step (or pre-process) of generalization for visualization purposes.

As pointed out previously, generalization is a comprehensive task. In practice, both the motivations and the objectives of generalization are often multiplex instead of 'pure'. The expected result of generalization should be a proper data set that meets the requirements in all respects. For example, if a data set is generalized at target scale 1:1M, the resolution (both in spatial and in attribute terms) of the information content will be reduced to a specified level. At the same time, the geometric complexity of the representation and the volume of the data should match the target scale as well. In the implementation of generalization processing, the considerations for different views are often closely related to each other. For example, when we carry out abstraction processing of the data, the visual representation is normally used as a measure for assessing the result. Though we can also use other measures for this purpose, the visual measure is no doubt the most intuitive and convenient one. Therefore, the information contents of data should by no means be completely separated from their visual representations in the digital world.

In the meantime, generalization techniques are usually two-fold or even multi-fold in function. A technique may be used for different purposes. Different procedures

of the same techniques may lead to different results. For example, if using the Douglas algorithm and 3-point average smoothing method to perform line simplification, we can have two procedures: the Douglas method followed by the smoothing process, or the smoothing process followed by the Douglas method. By using the first procedure for getting 50% point reduction, the shape of the line is greatly simplified. If the second procedure is used for getting the same rate of point reduction, the new line is very similar to the original one in shape. Obviously, the first procedure emphasizes shape simplification, so it is suitable to be used for view generalization. The second procedure emphasizes the reduction of the coordinates. It is more useful in data volume reduction. This example indicates that the processes for different purposes are often related to each other. Sometimes, the techniques involved in two different processes may be exactly same. Through elaborately designing the procedure, we can strengthen the functions of the techniques in this aspect and suppress the effects in another.

The integrated view of generalization does not hinder us from solving the problems by using the divide-and-conquer method. Conversely, it allows us to pay attention to the relation between different generalization views and generalization techniques instead of isolating one from another. This point is significant for dealing with complex problems.

2. 2. 6 GenSystem

GenSystem is a database generalization system developed at CCRS. The system

provides techniques to pre-process spatial data, code with attribute information, automatically structure, display, and tag the information by exploiting topologic and geometric considerations (Richardson 1996). The architecture of GenSystem is illustrated in Figure 2. 1.

Figure 2. 1. The architecture of GenSystem (from K. Kushigbor)

GenSystem has been developed in an ARC/INFO environment. Object-oriented methods are adopted in the data abstraction processes. The system provides a set of generalization tools to users. Through a graphic interface, the users can choose the

appropriate method depending upon the purpose of the generalization and the properties of the features, appoint the target scale, and interact with the system.

A spatial database can be linked to the GenSystem through the Metadata System. The Metadata System is a file management system in which the information about the data (such as the path of the data file's location and the format of the data) and certain attributes (such as the classification of features) are included. Some data conversion utilities have been performed so that the data from some other databases can be processed by GenSystem.

In this chapter, the concept and the issues of generalization were discussed. The techniques and methods used in generalization were investigated. In the next chapter, the mechanism of the data access across the Internet will be explored. The considerations involved in carrying out generalization in an Internet environment will be discussed.

CHAPTER 3: SPATIAL DATA ACCESS IN AN INTERNET ENVIRONMENT

The Internet is a fast growing area of computer application and information science. The powerful utility of the Internet in information sharing and transmission has made it popular in business, research, education, and our daily life. More and more people use this tool to search for, transfer, and publish information. Among the Internet applications, the World Wide Web (WWW) is by far the fastest growing segment. Together with multimedia Web browsers, the WWW provides users with a powerful, easy-to-use interface to the Internet. Through the WWW, users can access many of the basic Internet utilities (such as FTP, Telnet, Usenet etc.), as well as a variety of external applications on the Internet, such as databases, GISs, and application software packages. Today, the WWW has become the largest, most comprehensive and most widely used electronic information system in the world.

In this chapter, the components of the Internet that are relevant to spatial data handling will be summarized and explored. The exploration will first focus on the WWW applications. The properties of the generalization process in a WWW context will then be discussed.

3. 1 The components of the Internet

3. 1. 1 The Internet and computer network

The term “Internet” (spelled with capital “I”) derives from “Internetwork” and refers to a worldwide collection of computers and networks that have agreed to interconnect and communicate using a common protocol suite (Carrol, 1996). The first internetwork was built nearly 27 years ago as a project of the US Department of Defense. Building such an internetwork was mainly for the military uses in data delivery. Its goal is to create a way for widely separated computers to transfer information and data, and to make data communications as robust and reliable as possible. The network, initially, was called the Advanced Research Project Agency network, or ARPAnet. Later, it was renamed the Internet.

Technically, the Internet is not a network of computers -- it is a network of networks (Stout, 1996). Local area networks (LANs) throughout the world are linked together by telephone lines, cables, communication satellites and other media to form a huge worldwide network.

Network is a technology used in computer communication. Through a network architecture, users can reach any machine on the network, and so use computing resources that are not confined to a single location. The benefit of this approach is that the network can offer huge data resources to users and a cost-effective mechanism for application processes.

So far, a variety of computer networks have been developed and established for different uses at different locations of the world. Using a set of communication rules, those systems (both computers and computer networks) which come from different suppliers can communicate with each other, so that they can be integrated together to form a super network -- Internet. Those rules concerned with computer communication are termed protocols. Both the inter-operability and the ability of many different types of computers to exchange data are the great strength of the Internet and the technology on which it is based (Lane *et al.*, 1993).

3.1.2 Protocols

Protocol is a key element in the Internet. It is responsible for making LANs understand each other in their communication. Any computer that wants to communicate on the Internet must “speak” a language, i.e., TCP/IP. TCP/IP stands for Transmission Control Protocol / Internet Protocol. It is not a single protocol, but consists of a series of protocols (usually called the Internet suite of protocols). This suite includes TCP, IP, TCP/IP applications and many low level protocols used for the network inter-communication. Those protocols interact with each other to provide services. They are collectively known as TCP/IP. TCP/IP is a layered protocol suite. Figure 3.1 illustrates its structure.

The function of each layer of the protocols can be summarized as follows:

Network inter-communication protocol: This is the lowest level protocol.

Figure 3. 1 TCP/IP protocol suite layers (modified from Elbert *et al.* 1994)

At this level, TCP/IP communicates with other network protocols such as Novell, Appletalk, etc.

Internet Protocol (IP): IP takes care of addressing issues. In the Internet, each host (computer) has a unique address which uniquely identifies this host to all others. The utility of IP is similar to an envelope in traditional post office systems. According to this address, the data intended for the system can be delivered to the proper host.

Transmission Control Protocol (TCP): In the Internet, the data sent across IP networks are broken up into pieces (called packets) for transmission. The size of each packet is usually less than 1500 characters long. TCP is responsible for breaking the data into pieces and numbering each piece. Those pieces of data are transmitted neither necessarily in order nor necessarily taking the same route. On the receiving side,

TCP collects all those data pieces, and puts them in the proper order so that the user can receive the entire data file.

TCP/IP applications: At this layer, there is a set of protocols for processing and application purposes. Those protocols govern the operations of actual end-use applications. They serve for the communication between hosts on the network. The applications include not only the protocol elements that support specific types of information transfer, but features and facilities which support the end uses. The following are the most widely used application protocols in the Internet.

- **Telnet:** It is the Internet remote login application. Users can use it to login into other hosts on the Internet. Once the connection with a remote host is created, the user can access the services as his/her keyboard is connected directly to that computer.
- **FTP (File Transfer Protocol):** The responsibility of FTP is to move files from one computer to another through the Internet. Regardless of the kind of computers and the type of operating systems used in those computers, the files can be transferred between them as long as both computers can “talk” to the FTP and have access to the Internet.
- **SMTP (Simple Mail Transfer Protocol):** Its job is to transmit electronic mail over the Internet. Today, it is frequently used in our work and daily life for communicating with remote hosts.
- **Internet Gopher:** Internet Gopher is a TCP/IP Process/Application Protocol. It is

used in a software tool named Gopher to connect a variety of textual data archives on the Internet and display them as a series of menu items. Internet Gopher protocol asks for a client/server methodology for the searching and retrieval of information. It represents the information as a highly structured hierarchy (a multi-level menu). The client/server model will be discussed in next section.

- HTTP (HyperText Transfer Protocol): HTTP is the protocol for moving hypertext files across the Internet. A HyperText file is a text file (usually written in the language HTML) in which some links are embedded in words or phrases. Those links connect to other documents and allow users to select. The selection operation will cause another document to be retrieved and displayed. HTTP offers a non-linear, structureless presentation. Following the links, users can move the connection to any host that the links point to. HTTP also asks for a client/server model for data transmission. It is the most important protocol used in the World Wide Web. As TCP/IP is the language of Internet, HTTP is the backbone of the World Wide Web.

In addition to the above protocols, there are many other application protocols, such as Usenet, Archie, WAIS etc., which are relatively less used.

3. 1. 3 The World Wide Web (WWW)

The World Wide Web (WWW or simply the “Web”) is a collection of protocols and standards used to access the data stored on computers all over the world which are connected to the Internet. The WWW defines and unifies the “languages” used to

retrieve the data, and provides an entry point to the Internet.

The WWW is the newest information service of the Internet. It can speak the languages of HTML, Internet Gopher, FTP, Telnet, Usenet, and WAIS. However, HTTP serves as the primary protocol for information retrieval via the Web.

The WWW is primarily defined by three standards: HTTP, URLs (Uniform Resource Locators), and HTML (HyperText Markup Language). These standards are used by the WWW to locate, access and display information over the Internet. URL is the standard way to give the address of any resource on the Internet. HTML is a coding language used to create HyperText documents.

The WWW uses a client/server model to transfer information. The client/server model includes two basic parts: one is the server, which has some particular software (in WWW domain, it is called Web server or HTTP server) and stores the data that will be transferred; the other part is the client, which provides users with a user-friendly interface to the server's data. The server's data on the WWW (also called Web documents) may be graphic, audio, or video as well as plain text. The client/server mechanism works in the following way (modified from Spainhour *et al.*, 1996):

Firstly, the client initiates a transaction:

- (1) The client first establishes a connection to the server through a URL.
- (2) Using an HTTP command, the client then sends a document request to the server for specifying a particular document to be retrieved.

The server responds:

(1) The server sends a response, which contains some status codes, to the client.

(2) If the client's request is successful, the required data are sent. These data may be a copy of a file, or the response from a CGI program (see section 3.2.4: The concept of Common Gateway Interface).

In WWW terminology, the client is called a browser. A Web browser can interpret and display HyperText documents. It provides an interface to users. At present, several powerful browsers are available, such as Netscape Navigator (developed by Netscape Communications Corp.), Internet Explorer (by Microsoft Corp.), and Mosaic (by the National Center for Supercomputing Application). Those user-friendly browsers provide users with a wide range of facilities to access the Internet resources.

3.2 The access and manipulation of geo-data across the Internet

3.2.1 Why Internet?

The development of GIS technology provides a powerful tool for producing, processing and maintaining geographic data. A huge amount of spatial data has been made available all over the world in the last decades. Data publishing and delivery, now, has become one of the key constraints to the efficient use of data. The Internet has been proved to be a powerful tool for dealing with this problem. It can facilitate both data suppliers and users in several respects.

- Geographic data are usually communicated visually and digitally. The Internet (the

WWW application in particular) provides a great environment for distributing visual information in a readable and reusable format.

- The need for geographic data is rapidly increasing. It becomes more and more difficult to meet the demands for various spatial data sets by a centralized model of data production and distribution. Today, the decentralized and communication-focused model is becoming a trend. The Hyper-Link mechanism of the WWW and the distribution of geo-data on the WWW appear to be a powerful solution to disseminate various data to a large number of users in a global setting. A good example is an on-line information system named Alexandria Digital Library (ADL) (University of California, Santa Barbara, 1997). ADL provides a Hyper-Link mechanism to users. It allows users to access distributed geo-referenced databases worldwide.
- The powerful searching utility of the WWW helps users in data resource searching, locating and information exchange. It can minimize duplication of effort in the collection of expensive spatial data and foster cooperative data collection activity.
- In addition, data publishing and data exchange call for an effort to standardize the data formats. The use of WWW will promote the process of geo-data format standardization.

3. 2. 2 Static access to geo-data

A variety of systems and applications for geo-data access based on the WWW

can be found. Their purposes and the properties are various, so are the techniques used to construct the applications. Luciano Fortunati (1996) summarized the current architectural solutions for data distribution into three levels: at the *low level*, the data on the entire geographical area and with various themes are placed in one or more files. The server does not need a GIS to operate; at the *medium level*, the data on the geo-data server are suitably structured and made available on the WWW, but the geo-data server does not use a GIS to manage the data though a GIS is indispensable for organizing the data; and at the *high level*: the geo-data server uses a GIS to support data management and access.

In practice, quite a number of applications do not ask for a GIS. It is not always necessary to bring a GIS into an application. The architecture of an application is mainly determined by the purpose and the characteristics of the application. Therefore, the division which is based on whether a GIS is involved may not clearly describe the nature of WWW applications. In this exploration, the applications are sorted according to the mechanism of data access. From this idea, the methods of spatial data access can be summarized into two types: the static method and the dynamic method.

Static access means that the data (spatial or non-spatial) for dissemination are already placed on the Web server. They are maintained more like an archive of data files. They are made available to users through a simple Web client/server mechanism. The operation for accessing data is no more than to locate the data by following their URLs. In this process, there is no new information to be produced or introduced. What

they put in is what you can get.

This is the early stage of WWW application and is also the basic use of the WWW in data distribution. At this stage, the user can only reach the existing data on the Web server. The geo-spatial data exist in a constant and “dumb” state. Usually, they are stored in a raster structure (i.e., image file, mostly in GIF or JPEG format) for display. Their URLs are embedded into HTML documents. A map can be located and then displayed by following a corresponding link. Sometimes, the data publishers also provide a data copy mechanism to users. In this case, the user can directly transfer the expected data to his/her computer instead of displaying them on the browser. This can be done by using the Internet tool FTP. Today, this model is widely used to publish data. An example of this model is China Dimension (established by the Consortium for International Earth Science Information Network, Saginaw, Michigan, <http://sedac.ciesin.org/china/>). This is an on-line spatial data service. In this application, a set of links (index) is provided for locating and displaying the selected maps. At the same time, an FTP archive is provided to users for transferring the data sets. Using FTP utility, users can get the copy of each map on the map server.

3. 2. 3 Dynamic access to geo-data

With the advent and application of some new techniques, such as the Common Gateway Interface (CGI), Java and ActiveX (which will be discussed in the following sections), a Web server can link to some specific software packages, a relational database

or a GIS. This allows users to execute a program or access a database in real-time, so that the information can be dynamically accessed. In a dynamic process of data access, new information (or data sets) can be introduced or produced in real-time at the user's request. Those new information and data sets are not originally on the Web server or may not even exist before the user's request is issued.

In a client/server context, any process which is based on the Web server can be considered as a Web server application. A Web server application can be either an executable program package for a particular purpose, or a database. Usually, a program package is linked to the Web server through CGI (see the next section for details). A database (or a GIS) is linked to the Web server through either CGI or a specific application server which can directly communicate with the Web server. Hageli *et al.* (1996) described an application, in which an ARC/INFO server is linked to the Web server through CGI. The Internet Map Server (IMS) for ArcView (developed by ESRI Inc.) takes the other way: IMS is made compatible with both the Web server and HTML so that ArcView can be linked to a Web server through IMS, and its functionality can be accessed.

A variety of Web-based mapping products that support dynamic data access have been developed. Some products developed by the GIS leaders are reviewed by W. F. Limp (1997) and S. D. Johnson (1997). Those products provide interfaces for creating maps or accessing databases. In some systems, such as MapObjects IMS, ArcView IMS (both developed by ESRI), MapInfo ProServer (developed by MapInfo

Corp.), new maps can be dynamically created from the existing data. Those dynamically created maps are linked to their specific databases so that the user can get the report of the map elements through querying the maps. Some systems, such as Autodesk MapGuide 2.5 (developed by Autodesk Inc.), use pre-composed maps. Using these packages, the users cannot dynamically create a new map, but they can access a live database. Thus, the attribute information about the map elements can be dynamically accessed.

3. 2. 4 The concept of Common Gateway Interface (CGI)

CGI is an interface between the Web server and the other resources of the server's host computer. Using CGI, Web clients can execute the programs on a Web server and receive their outputs. Today, CGI has become a widely used standard that provides a mechanism for Web servers and executable programs to communicate with each other (Greenstein, 1996). As we know, a plain HTML document is static. It can only provide users with constant information. A CGI mechanism allows the Web server to execute an external application in real-time while passing user-specific data to the application. The application then processes those data and the server passes the application's response back to the client.

CGI is language-independent. A CGI application can be written in any language (such as C/C++, Perl, Unix Shell, Visual Basic), as long as it can be executed on the Web server computer. Normally, CGI applications reside in a special directory on the server

host computer, mostly in */cgi-bin*. Thus, when CGI applications are invoked, the Web server knows to execute them rather than just display them to the browser.

In the Web client/server mechanism, CGI applications are executed in the following way:

The user makes a request to the server to run an application through a HyperLink in which the URL of the application and an HTTP method are embedded. The HTTP method invokes the CGI application and runs it. After the application is executed, the Web server will report to the client with some status and header information, followed by the result.

Most commonly, a CGI application is invoked through an HTML form. An HTML form is used as an end-use interface which allows the user to interact with a Web server. An HTML form usually consists of a series of check boxes, buttons, or text areas. Users can make their requests by selecting buttons, or input extensive-text data on text areas. The CGI applications then will be executed according to the user's input. The data (text, user's selection, or parameters) input from a form are called form data. Form data are sent to the server in a stream of name-value pairs. Those name-value pairs are URL encoded, in which the pairs are separated by the character `&`; the spaces are changed into pluses; some characters are encoded into hexadecimal. In order to read the encoded form data, CGI applications have to provide some way to decode them, so that the format of the data becomes legal to the application programs. The procedure of CGI implementation can be denoted as in Figure 3. 2.

A CGI application can be a program package for some particular purpose (such as a software package for map generalization), a database, or a GIS. For linking a GIS to the Web server, a specific application server is usually required. The application server is responsible for communication between the Web server and the GIS.

Figure 3. 2 A CGI process

3. 2. 5 Java approach

Java, developed by Sun Microsystems Inc., is an object-oriented programming language. It has become popular in WWW society over the past few years. Unlike other object-oriented languages, Java is platform independent. This advantage is perhaps the main reason that makes Java so popular.

A Java program can be an applet or an application or both. Java applets are specially used in an Internet environment. An applet is downloaded over the WWW and executed by a Web client (browser). In order to view or execute a Java applet, the browser must be Java-capable. A Java application is a general program written in Java (just like a program written in any other programming language). It does not require a

browser to run. Therefore, it can be used for any programming purpose.

At present, a common use of Java in the WWW is to create graphic interfaces for displaying spatial data and interacting between users and Web applications. Some GIS vendors take this approach to bring their GIS to the Internet. For example, ArcView Internet Map Server uses ready-to-use Java applets to help users publish data quickly. Today, the most popular Web browser, Netscape Navigator, has become Java compatible. Internet Explorer also supports Java plug-in components.

The Java approach is quite different from a common CGI's interactive process. CGI scripts are host-based. They require server processing for every program piece launched. Java puts the local process mechanism to work. When the user's request is sent to the Web server, the Java applet is downloaded to the local computer. The user interacts with the applet locally, and the server simply sends data to the client system.

Java applets can provide complex capabilities, such as spatial retrieval, searching, querying, zooming in, zooming out, or even interactive map design, depending on how the applet is designed. Once downloaded, the Java programs run quickly (downloading the whole applet may take time!) because the processes are carried out in the local computer. It greatly reduces network traffic since a large volume of data will not have to move frequently across the Internet. At the same time, it reduces the processing burden on the server. Those advantages are significant, especially in graphic-intensive applications.

Java creates new channels for GIS information to enter the Internet. A Java

applet provides a front-end graphic user interface to accessing the geo-spatial data. Java-based new browsers can access and view not only raster data, but also vector data. A GIS can be linked to the Web server either through CGI associated with a specific GIS server or through a Java server. E. J. Strand (1997 b) gives an example in which the GIS communicates to the Web server by using the CGI method. DDView 3.0 (Sovik, 1997) uses a Java server at the host machine for the whole processing. The Java client/server interface makes input and output fully interactive. In this method, CGI is not needed.

3.2.6 ActiveX approach

Another software component technology widely used in Web applications is ActiveX. ActiveX was developed at Microsoft Corp. It originates from another Microsoft technology called Object Linking and Embedding (OLE), which is an object based software architecture. Microsoft retooled its OLE controls to be network enabled and renamed them ActiveX controls (Strand, 1997 b).

OLE is a Windows facility that allows users to use several Windows applications to produce a single document (called a compound document). In OLE, the original data become an object within the application that uses the data. An object can be any data displayed in a compound document, such as text, graphs, charts, parts of spread sheets or pictures. The object contains the original data in its native format. When an object is inserted in another document, it keeps an association with the application that creates it.

ActiveX allows developers to deploy applications based on objects. The objects can be easily replaced and upgraded. With Java, when you download a Java applet, you download several platform-independent object-oriented classes. An ActiveX component, by contrast, is a self-contained binary file, and the components must be compiled differently for every platform (Carr,1997). The ActiveX components can also be designed to be automatically downloaded on the user's system when the user accesses an ActiveX application.

Intergraph Corp. used the ActiveX method to develop its GeoMedia Web Map software for Web servers and browsers. GeoMedia Web Map allows the user to retrieve data from Intergraph's GIS. The graphic data are converted to Computer Graphics Metafile (CGM) format to serve data transmission across the Internet.

ESRI Inc. used the ActiveX approach in its product MapObjects Internet Map Server. MapObjects is a collection of mapping and GIS components for application developers. It is built upon the OLE 2.0 standard, and includes over 35 programmable OLE objects in it. MapObjects is not for end users, but for the people who want to develop applications. It provides some popular programming environments, such as Visual Basic, Visual C++, Microsoft Access, to developers. Developers can use them to create customized mapping dynamically, and develop various GIS applications.

3.3 Spatial data display

Geo-spatial data can be disseminated through the Internet in different ways. If

the data are transferred by using FTP utility, there is no need to worry about the structure and the format of the data. The FTP utility enables data to be transferred between dissimilar computers. However, if the data are accessed and displayed across the WWW, the structure and format of the data must be taken into consideration. At the early stage of the WWW application, only raster data can be accessed and displayed through Web clients. The most popular format used in this stage is GIF. Today, raster image (GIF and JPEG) is still the most popular data model used in the WWW.

With the adoption of new technologies in the WWW, vector data can be accessed and viewed through Web browsers. The raster method and the vector method are based on different techniques. Each method has its own advantages and drawbacks. Which method should be adopted in an application mainly depends on the properties of the application.

3.3.1 The raster method

In the raster method, spatial data are stored in graphic images. Graphic images are most commonly used in two ways in the WWW: data display and image mapping. For data display purposes, the URLs of image files are embedded into an HTML document. The images can be directly retrieved from the server and displayed on the browser by following the links. Image mapping is a special term used in the WWW context. It means that the image is used as a complete interface. In this use, the image is divided into several regions which link to different pages or data files. Clicking on an

area of the image sends the coordinates of the mouse click to a map file that contains the coordinates of the rectangle, circle or polygons that define the various “clickable” regions of the image. By checking against the map file, Web control knows where the user clicked, and thus, where they wanted to go. The appropriate page or data file is then retrieved for them.

The image mapping method is usually used to create a geographic index on a Web page. A geographic index is more intuitive than the text index system for users to retrieve the geographically distributed data. It is a common method in Web page design that the region-related data are organized as individual files. Those files can be put on the same page or reside on separate page sites. An image map is designed and used as an interface for users locating those data files or pages. Clicking on a region on the image map, the user can access the data file that is linked to this region, or get to a related page site.

The data access in the raster method can be either static or dynamic. As discussed previously, the static data access is a process in which the data placed on the Web server are directly retrieved and displayed. The dynamic data access, however, needs some special technology, such as CGI (discussed in section 3.2.3), to be adopted. In the raster method, a GIS (or database) is usually involved for the dynamic access purpose. With the support of GIS, some GIS-like spatial operations, such as selecting, locating, querying, zooming in, zooming out etc., can be performed across the WWW. In a dynamic process, a GIS data server receives requests from the Web browser, then

retrieves the geo-spatial data, creates a raster image map and sends it back to the Web browser for display (Strand, 1996 b). As discussed above, an image can be made into an active map and link to a database. Thus, clicking the mouse on the map serves as a GIS query operation.

The raster method has some advantages. The most significant one is that the data structure is simple, hence, it is easy to handle. This may explain why the raster method is widely adopted in WWW applications. Some spatial operations, such as windowing-in and windowing-out, can be performed more efficiently by using raster data. This allows users to get a quicker response. Another advantage of the raster method is that no special software is required on the client side for data display. Any browser which is HTML compatible can directly display a GIF image map.

Raster data usually take up a large volume of memory. Large data volume means a longer transfer time and a slower response from the Web. This is a big disadvantage of the raster method. Another drawback of it is that the interaction between the user and the map is not efficient. When the user clicks the mouse on a GIF or JPEG map on the client side, the software only knows the mouse's x, y coordinates. These coordinates are used to match with the data stored in a server's GIS for identifying the map element at this location. Once the map element is identified, the information about the element can be accessed. For example, when a user clicks the mouse on a road, an (x, y) pair will be picked up and sent to the server. The server with GIS capabilities determines the corresponding map element (the road) according to the value of the (x, y) pair. Then, the

attribute information of the road, such as the name, type, grade, length etc., is retrieved and sent back to the client.

In this process, the user interacts with the server for each access performance. The data need to move back and forth over the network. This will slow down the response. At the same time, the x, y coordinates are the only information the server gets from the map (at client side). Hence, they are the only information to be used to identify the map element by the server application. If the graphics are intensive on a map, any click that is not exactly on the right map element could lead to an incorrect match (Limp, 1997).

3.3.2 The vector method

The plain HTML interface does not support vector files. Vector data access and display through the WWW become possible only after some new techniques (Java, ActiveX) are adopted in the WWW,

The vector method is different from the raster method in mechanism. The vector method is object-oriented. The map elements on a vector map, such as points, lines and polygons, are considered as individual graphic objects. They are not “dumb”, but associated with attribute information. In order to display a vector map and query graphic objects on the Web browser, some special software (termed plug-in) is required on the client computer. Plug-ins must be installed locally before the vector data are accessed.

Vector maps can take different ways for transmission over the Internet. W. F. Limp (1997) discussed some leading Web-based mapping products which use different methods for data access. One way is to use a pre-composed map for the access of spatial components of data. In this method, the map is prepared before it is put on the Web. The characteristics of the map are pre-defined, hence the user cannot create a new map in real-time, but interact with the existing map in several complex, pre-defined ways. The results, in this case, are fully controlled by the developers. The map can be linked to an on-line database, so that the user can carry out dynamic queries. Autodesk Inc. adopts this approach in their product MapGuide 2.5.

The other way is to use existing data directly when needed. This allows users to create a map on-the-fly, and query a database dynamically. This method is used in GeoMedia Web Map, in which the geographic data must be prepared in the Modular GIS Environment (MGE) format for supporting the Web application.

Vector data have several advantages over raster data for the use in WWW applications. Usually, vector data are less voluminous than raster data. In the vector method, geographic instances can be constructed as graphic objects. The software knows the points, lines, or polygons indicated by the mouse rather than only the x, y coordinates. With the help of some plug-in tool, the queried information can be reported locally, without having to go back to the server for processing. Therefore, a faster interaction with the user can be expected. In addition, vector data can be scaled down or up to fit the display area without losing resolution. Complex geographic features can be

represented more precisely.

A disadvantage of vector data is that the data structure is more complicated than raster structure. In the vector method, the spatial data usually need to be processed before they can be used in Web applications. In data access, the techniques required for dealing with vector data are more complicated. Another disadvantage is probably the use of plug-ins. A plug-in may be a Java, ActiveX downloaded applet, or other specific packages. Java applets do not need to be specially downloaded and installed by the user. They will be automatically downloaded to the local machine when they are accessed. Some applications (such as DDView 3.0) use the Java server and client technology for data access. They allow users to access the vector data by using a Java-capable browser. No extra work needs to be done by the users. Some systems provide specific plug-in utilities to users. Users have to download and install the plug-ins before they access the data. If the user does not have the permission to install a package in the local computer system, he/she will be not able to access the data. Moreover, downloading a plug-in applet takes considerable time, especially the more powerful ones: a powerful applet application often means a large package.

The use of plug-in has two sides. Just as discussed above, it may create extra work for the user, and sometimes, even become a constraint in data access. On the other hand, it makes interaction between users and the Web more efficient. With a client-side plug-in application, some GIS-like operations, such as object selection, panning, zooming in and out, can be done locally. The response the user gets will be

much faster than that in the host based processes.

In sum, the raster method and vector method each has advantages and disadvantages. One method may be good in some aspects, but not in others. No one method can suit all situations perfectly. Generally speaking, if a highly interactive process or database query is required in the application, the vector method may work better. Inversely, if the purpose of an application is mainly for data display, transmission or location-oriented operations, the raster method may be the better choice.

3. 4 Data format conversion

Data conversion is a big problem in geo-spatial data access and integration. With the application of GIS technology, a large amount of geographic data has been developed and used in the geomatics industry. Those data come from different producers, and are offered in a variety of formats. Since different formats have different specifications, they are incompatible with each other. To use data which are not supported by the local system, the user has to convert them into the format that is legal in his/her system. Data conversion is a difficult and time consuming task. It makes up a large part of the cost in GIS implementation.

In order to facilitate the data application and make the system capable of fitting different data formats, most GISs provide a set of translators for converting foreign geo-spatial data formats into their own formats. Though this method is helpful in overcoming the data format barrier to a certain degree, its applicability is limited. On the

one hand, a limited number of translators can hardly fit the large variety of data formats. On the other hand, a large set of translators will increase the cost of a GIS, and make the system cumbersome. In addition, the conversion between different data formats often results in a loss of significant information.

A solution to this problem is standardization. Standardization has two objectives: one is to minimize the number of geo-spatial data formats; the other is to facilitate information exchange. Several standards have been established, in which the DIgital Geographic information Exchange STandard (DIGEST), the Spatial Archive Interchange Format (SAIF), the Spatial Data Transfer Standard (SDTS), ISO, and TC/211 are some of the important ones.

Standardization is an important measure towards uniforming data format, facilitating data exchange, and overcoming the geo-spatial data barrier. However, the standardization effort alone will not solve the problems. Today, a variety of GIS are in use. The system architectures of those GIS and the data models used are different from one to another. Most GIS developers have established their own proprietary data formats for speeding up the processes and operations in their particular systems. Those formats will remain in the market because they have been widely used in the GIS industry. It is very unlikely the industry will move to a single standard. At least a half-dozen significant standards can be expected (Clement *et al.*, 1997).

A new technique, called the Open Geospatial Datastore Interface (OGDI) provides another solution for dealing with this problem. OGDI is an application

programming interface (API) that resides between a GIS software package (the application) and various geodata products, to provide standardized geo-spatial access method to the latter (LAS Inc., 1997 a).

OGDI was developed by the Department of National Defence of Canada (DND), in partnership with Logiciels et Applications Scientifiques Inc. (LAS Inc., 1997 a). Its purpose is to provide a transparent access to various geo-spatial data formats. Using OGDI, different spatial data formats can be read and displayed without any translation or conversion. If users like, they can drop the data they read into any format that their systems can support. OGDI also provides tools for adjusting coordinate systems and map projections.

OGDI mainly consists of two parts: a C language API, and a set of drivers. The API is a dynamically-loadable library, used to access OGDI facilities; drivers are a set of dynamically-loadable libraries. They are used to process API's requests and to access various geo-spatial data formats, one for each format. When an API issues a request, a corresponding driver will receive it, then, the driver retrieves information from the source data, translates it into a transient data model and returns the result to the application.

OGDI uses a client/server architecture to disseminate spatial data. The driver of OGDI can be accessed either locally or remotely. For remote access, a program called glttd (geographic locator transfer protocol daemon) and a network driver are required. They are used to create a connection between the application and remote drivers through

a TCP/IP network. Therefore, OGD I can be implemented over the Internet.

Today, OGD I drivers are available for several data formats, such as DIGEST, VRF, VMAP, DNC, ADRG, DTED. Drivers for three proprietary formats, GRASS, ARC/INFO and PAMAP, are also available (Clement *et al.* 1997, and LAS Inc., 1996). A GIS, named GRASSLAND, has brought OGD I into the system, and used it to handle different data formats.

3.5 Generalization across the Internet

While the application of the WWW in geo-spatial data dissemination is advancing rapidly, the generalization of geographic data across the Internet remains a challenge. Just like the situation in a GIS environment, the implementation of generalization over the Internet is an appealing application. Using an automated generalization mechanism, a series of data sets at smaller spatial scales can be extracted from the source data. If such an application can become operable over the Internet, both data producers and users will benefit. Firstly, the producers do not need to maintain a series of databases at different spatial resolution to supply data. The users can avoid downloading and using a large data set with redundant information. It also no longer becomes necessary for the users to have generalization tools with their GIS.

Carrying out spatial data generalization across the Internet has become possible with the advancement of WWW applications. In principle, there is not much difference between the implementation of generalization application and the implementation of

other dynamic WWW applications. But data generalization is usually more complicated. Some specific characteristics need special attention.

3. 5. 1 Considerations in generalization applications

Generalization application is a dynamic process. In this process, a new data set will be extracted from the source data as the result of a generalization process. The requirement for data transmission in generalizations application is slightly different from most Web applications, where the responses of the server are sent back to the browser for display or for interactive uses. In generalization applications, the result of generalization is not necessarily displayed on the browser. It can be directly sent back to the user's computer (local computer) by using either FTP or a download approach. This will save lots of data preparation work. Of course, it can also be sent back to the browser for display, so that the user knows whether the result is satisfactory or not. To do this, the data need to be further processed, e.g. converted into GIF format, prepared for display, etc. This is time consuming. Actually, after receiving the data, the user can display the data using his/her local GIS. This will be more economical and more efficient.

In generalization applications, the geographic data involved usually have a large volume. This means a relatively long time is required for data transmission over networks. Moreover, the generalization process also needs some time to execute. Time is a very important consideration in Internet applications. Users who have to wait for a

long time may lose interest. Of course, the generalization application is mainly for the users who need data for serious applications. A certain level of patience from the users can be assumed for the generalization processes.

In order to save running time caused by network use and to facilitate data applications, the data organization should allow users to have flexibility so as to avoid dealing with an unnecessary large package. For example, a data set (coverage) in a GIS may be organized to cover a very large area for some specific uses. Sometimes, a whole province or even a whole country region is involved in a single coverage (such as a road layer coverage of a country). In Internet applications, this is not a good choice. The user who only needs a data set covering a small region has to deal with the whole package of data. Handling a large data set in which a great part of the data package is unnecessary to the user will waste the user's time and energy. Though we can use some GIS utilities (such as clipping, masking) to make a proper data package for generalization, those GIS operations themselves are time consuming in implementation.

A good choice is to divide the data package into tiles (the definition of term tile will be given in Chapter 4), just like map sheets in conventional cartography. Thus, users can freely choose the tiles they want, then generalize those relatively small tiles. If required, they can join those generalized tiles into a large coverage on their local GIS.

3.5.2 The problems introduced in generalization implementation

As discussed above, the data sets should be organized at a proper size for

facilitating the use and the transmission over the Internet. While this gives flexibility to the users in their applications, it can introduce inconsistency among the data packages during the generalization processes.

In a generalization application, if more than one tile is involved, those tiles will be processed separately and sent back to the user through the network. The problem is that those generalized tiles may be not compatible with each other spatially. For example, if a river goes across two adjacent tiles, the river on one tile may be selected because it extends a long distance, but it may be ignored on the next tile since only a short part of it lies on the tile. When those two tiles are joined together, an error will occur -- the river is truncated.

In a GIS environment, this problem can be solved through interactive processing (either pre-processing or post-processing). In an Internet environment, this approach is inapplicable. Interactive data processing across the Internet is too expensive in time. Therefore, a mechanism which can automate this process should be developed. This allows users to carry out generalization without raising inconsistency.

In this research, a mechanism that is used for dealing with the inconsistency introduced in generalization will be developed. This will be discussed in Chapter 4.

3. 5. 3 A model for bringing a generalization system into the Internet

To bring a generalization system (such as GenSystem) into the Internet, the simplest approach is to make the system into a CGI application. Thus, the system can

be invoked through CGI to carry out generalization, then the result will be sent back to the client.

In order to run the generalization system, an interface is required for users to interact with the system. GenSystem is developed in an ARC/INFO environment. Its interface is written in ARC/INFO AML. To bring it into the Internet, this interface should be replaced with a HTML form interface, or a Java applet.

The interface should allow users to perform operations, such as selecting the data tiles that the user wants, choosing the proper generalization tool, determining the target scale, selecting the generalization method according to the property of the application, and inputting some parameters which are required in generalization processes.

GenSystem can process data which are in both ARC/INFO format and Oracle format. In order to communicate with ARC/INFO GIS and an Oracle database, an ARC/INFO server and an Oracle server need to be involved. The model of generalization implementation can be illustrated as Figure 3.3.

In Figure 3.3, a consistent processor is involved in the model. It communicates with GenSystem and to process the generalization results so as to enable the generalized data sets to be consistent with each other. The development of this mechanism will be described in Chapter 4.

Figure 3.3 A model for bringing GenSystem into the Internet.

CHAPTER 4: THE DESIGN OF A MECHANISM FOR CONSISTENT GENERALIZATION

In this chapter, a mechanism for consistent generalization will be developed. The goal of the mechanism is to eliminate the inconsistency arising from generalization processes. Firstly, the requirement for such a mechanism and the methodology are discussed. Then the development of the mechanism and its implementation will be described in detail.

In order to facilitate the representation and to give a precise meaning, some terms used in this chapter are defined as follows:

- **Geographic entity (entity):** A real world phenomenon that is not subdivided into phenomena of the same kind, e.g., a road, a river etc.
- **Map object (object):** A digital representation of a geographic entity or part of an entity, e.g., an arc in a spatial database which represents a real world river. A geographic entity can be represented as one or more map objects.
- **Tile:** A data set maintained in a spatial database. It is the basic independent unit for a GIS to manage the spatial data and for the users to retrieve the data.
- **Map boundary:** The digital representation of the map border line (neatline).
- **Node:** The junction of two or more arcs, or the end point of an arc. An arc starts from and ends at a node.

- **Borderline node:** The node which lies on the map boundary. It is the intersection of an arc and the map boundary.

4.1 The need for the mechanism

In a spatial database, the geographic features are usually organized as layers. A data layer consists of a set of logically related geographic features and their attributes (Aronoff,1989). The layer may be organized thematically, e.g., a road layer, an hydrology layer etc. It can also be organized according to the spatial property of the features, e.g., the point, line and area features are stored in separate layers. Each layer includes several files that contain the spatial and the attribute information of the layer. Those data layers can be processed and retrieved individually, or put together to form a digital map.

When a GIS must handle a large amount of spatial data, the spatial database is usually divided into tiles. For example, in an ARC/INFO GIS, the database may be divided into a set of small coverages; in MapInfo, the term “table” is adopted to represent the same thing. As an independent data set, the tile is the basic unit which can be retrieved from and processed in a spatial database. The size and the partition of tiles are not fixed. They may be different from one database to another, depending on the uses of the database and the restrictions of the software. A tile can either include a large volume of data and cover a large region, or be a small data set that covers a small area.

The partition of tiles usually accords with the conventional map sheets, since most geographic data in GIS at present come from the existing maps.

In practice, the demands for data are not always in accord with the existing data tiles. Users often need some data sets that extend across more than one tile. Figure 4. 1 illustrates the case where the required data extends across four tiles. In this case, the involved tiles may be used or processed one by one, or if required, be joined together to

Figure 4. 1 The required data extends across four tiles.

form a single tile (coverage). In a spatial database, all tiles should be well pre-processed and be made compatible with each other both in spatial terms and in attributes. Thus, the adjacent tiles can be joined together without creating any inconsistency (sometimes, a projection transformation is required).

In a generalization application, if more than one tile is involved, there are two

methods to carry out the process. One is to join those tiles into a single tile, then generalize this larger coverage. The second method is to generalize each tile individually, then combine the results.

Using the first method, we can get a consistent result since all tiles have been joined together before they are generalized, and the generalization process will act on a single coverage. The drawback is that the system has to deal with a large data set. In the second method, the tiles are generalized separately. While a tile is generalized, the features are evaluated only to the extent of the currently processed tile. Whether an object can be selected depends totally on its importance in this tile. The integrity and the continuity of the corresponding geographic entity are rarely be taken into consideration. If a geographic entity extends across two tiles, the map object of one part of the entity may be selected in this tile, but the object of the other part may well be ignored in the next tile during the generalization process. Figure 4.2 illustrates the potential problems. Figure 4.2 (a) shows that the sections of rivers denoted by dotted lines are eliminated in the generalization process because they are considered less important in the tiles they lie on. Figure 4.2 b illustrates that serious problems can occur when two tiles are joined together.

For the users whose GIS are equipped with a generalization system, the first method is perhaps the better choice because they can directly interact with a GIS to carry out the pre-processes, e.g., adjusting the position of features that extend across map sheet boundary by using the edge matching utility of the GIS; joining the tiles

together and making a coverage which has the correct topology. This approach guarantees that the user can get a consistent result.

(a)

(b)

Figure 4. 2 The potential problems introduced in generalization process

In an Internet context, the situation is quite different from that in a directly operable GIS environment. The users can not directly interact with the GIS in which the data are maintained. The limited capability of interaction and the security considerations make it impossible for the outside client users to carry out such complex pre-processing on the server's GIS. In addition, some other constraints also need to be taken into

account (these considerations were discussed in section 3.5), e.g., in order to facilitate the flexible use of the data, and pursue an efficient data processing and data transmission through the network, a large data package should be avoided. In this case, the second method becomes a more reasonable choice.

To overcome the problems and ensure that the users get consistent results in the generalization processes through the Internet, a mechanism needs to be developed. The mechanism should allow the users to carry out generalization processes on different occasions without worrying about the consistency issue, e.g., the user can generalize one tile on one day, and generalize an adjacent tile on some other occasion in a reliable and consistent manner.

4.2 The methodology

As discussed previously, the approach of respective generalization can lead to inconsistency between generalized tiles. The reason for the problem is that the generalization processes are isolated from tile to tile. If a geographic entity extends across two tiles, there is no guarantee that once the object of the entity in tile A is selected when tile A is generalized, the object of the entity in tile B will also be selected when tile B is generalized. Obviously, if our attention is only focused on the current processed tile (simply called the current tile), the proper solution for this problem will not be found. Therefore, the key point for the mechanism design is that the mechanism should allow not only the current tile, but also the neighboring tiles to be considered in

the generalization process.

In this research, a generalization track file system will be developed for such a purpose. The main body of the generalization track file system is a set of generalization track (abbreviated to GNT) files, and two adjacent-relationship reference files.

A GNT file can be described as a relational table in which the generalization history information for the objects that extend across the map boundaries is recorded. The objects in question are defined by the borderline nodes, i.e., the intersections of the map objects and the map boundaries. Each tile (i.e., a coverage) has a GNT file. Those GNT files provide the references to the GenSystem. When a tile is generalized, GenSystem can get its neighbouring tiles' generalization history information from their GNT files. According to this information, a series of adjustment actions will be carried out on the current tile. The result of these actions is that the generalized tiles become compatible with each other. An example of the GNT file is illustrated in Table 4.1.

The meaning of each item (column) of a GNT file is described as follows:

- **Node#:** the serial number of the borderline node. Each record of a GNT file corresponds to a borderline node. All of the borderline nodes on a tile are reported in the GNT file of the tile.
- **Node-ID** the identification of the borderline node. In this item, a specific integer value is assigned to each borderline node. This value uniquely identifies the node.

The borderline nodes lie on different sides of the map boundaries. To refer to each side of the map boundaries conveniently, they are named as follows: the up

Node#	Node-ID	Distance	Obj-ID	S1-1	S1-2	...	S2-1	S2-2	...	S3-1	...
1	1001	2.37	102	1	1		1	1		1	
2	1002	4.69	86	0	0	:	0	0	:	0	:
3	1003	5.21	21	1	1	:	1	1	:	0	:
:	:	:	:	:	:	:	:	:	:	:	:
46	2001	1.22	362	1	1		1	0		0	
47	2002	3.45	229	0	1		0	0		0	
:	:	:	:	:	:		:	:		:	
82	3001	2.33	144	1	1		1	1		1	
83	3002	4.97	287	1	1		0	0		0	
:	:	:	:	:	:		:	:		:	
138	4001	3.02	25	0	0		0	0		0	
139	4002	7.29	31	1	1		1	1		1	
:	:	:	:	:	:		:	:		:	
:	:	:	:	:	:		:	:		:	

Table 4. 1 An example of the GNT file

side (north) boundary is named N; the right side (east) boundary is named E; the down side (south) and the left side (west) boundary are named S and W respectively.

The borderline nodes are encoded as follows: The nodes that lie on the boundary N are numbered from 1001 to 1999; those ones that lie on the boundary E are numbered from 2001 to 2999; the ones that lie on the boundary S and W are numbered from 3001 to 3999 and 4001 to 4999 respectively.

The borderline nodes that lie on the same side of the boundaries are sorted according to a specific item. In this example, they are sorted according to the values of the item Distance (see Figure 4.3).

- **Distance:** in this column, the distance for each borderline node to the origin tic point is recorded. A tic point is defined, in the GNT file system, as the intersection

Figure 4.3 The nodes on the same side of the map boundaries are sorted according to the Distance values.

of two map boundaries (they are usually the parallel and the meridian respectively). There are a total of four tics on each tile which are located on the four corners of the tile and used for registration purposes. The four tics are numbered in an order such as that illustrated in Figure 4.4 a.

For the nodes on boundary N, the origin point is defined as Tic1. Thus, the value of item Distance for each node (its identification value is between 1001 and 1999) indicates the distance between this node and Tic1. For the nodes on boundary E, the origin point is Tic3. Tic4 is the origin point for the nodes both on S and on W. Figure 4.4 b illustrates above definitions.

Item Distance is used to create the matchings between the borderline nodes on two adjacent tiles. There are some other measures which can be used for this purpose (see sections 4.4.2 and 4.4.3). The distance matching method here is taken

as an example to describe the mechanism of the GNT file.

Figure 4. 4 The tics and the origin points

- **Obj-ID:** the item Obj-ID includes the identifications of those map objects that start from or end at the borderline nodes. Each object has its own Obj-ID so that it can be uniquely identified. Usually, such a kind of item (the item name may be different) is included in some files of each coverage in a database, such as the item Cover-ID in the PAT file of an ARC/INFO coverage. It is used to identify the objects. Using this item, each object can be selected individually for editing or retrieval purposes. It also provides a convenience for the users to link external data files to the coverage. When the borderline nodes are created, the value of Obj-ID for each corresponding object can be obtained from some file of the coverage.

The item Obj-ID here in this mechanism is used as a key. Using this key, the borderline nodes in the GNT files can be linked up with the corresponding objects in the database. When a node is examined, the corresponding object can be identified. Thus, we can adjust the status of the object (selected or eliminated) according to the information coming from the neighbour tiles in the generalization process.

- **S*-*:** the columns S1-1, S1-2 ... specify the target scales that the users can get and the algorithms used in the generalization processes. In the practice of automated generalization, a data set usually does not have to be infinitely generalized. If the target scale is too far removed from the original scale of the data, the result will become unusable. GenSystem is designed to provide several option target scales to the users. For each kind of geographic feature, several algorithms are designed to deal with the generalization processes. Different algorithms emphasize different properties of the feature. At the same map scale, those algorithms may produce different results. Therefore, a column should be created for each algorithm at each target scale. In character string S*-*, the first number denotes the target scale, e.g., S1-* indicates the first target scale; S2-* indicates the second target scale. The second number denotes the algorithm used in the generalization process, e.g., S*-1 indicates the algorithm 1; S*-2 indicates the algorithm 2.

In S*-* columns, if an object is selected in the generalization at this scale, the number 1 is put in the corresponding record. If the object is eliminated, the number 0 is put in. Thus, these items reflect the generalization results of the features that

extend across the map boundaries at each specific method and map scale.

The other part of the GNT file system is the adjacent-relationship reference files. They are used to determine the adjacent relationship between tiles and the neighbour relationship between the sides of the tiles in question. After these relationships are determined, the GenSystem can find the right sides of the right tiles that are related to the current processed tile.

While a tile is generalized, its GNT file is checked record by record. All nodes recorded in the GNT file will be examined sequentially. When a node (a record) is examined, its S*-* value is checked. If this value is 1 (it means that the corresponding object is selected), the node is released and the next record will be examined. If the value is 0, it means that the object is eliminated at this scale. In this case, the corresponding neighbour tile's GNT file should be checked. Figure 4.5 illustrates the neighbourhood of the sides between adjacent tiles.

According to the value of the item Node-ID and Distance, a partner node in an adjacent tile can be determined (see Figure 4.5, the partnership of nodes between adjacent tiles). Thus, a partnership between the two map objects which represent the different parts of the same geographic entity can be created. In this object pair, if the partner (the object in adjacent tile) is also eliminated at this scale, this node is released and the next record will be examined. If its partner is selected, the object in the current tile will be added to the collection of the selected objects. In other words, this object should be selected in the generalization process. Then, the examination continues until

all records of the current GNT file are examined.

Figure 4.5 Y is an adjacent tile of X; side W of Y is the neighbour of side E of X.

The term matching (a concept of graph algorithm design) can be used to depict how the GNT mechanism enables the generalized tiles to be made compatible with each other.

Suppose X and Y are two adjacent tiles; E and W are the neighbor sides of X and Y respectively (see Figure 4.5); X' and Y' are respectively the generalized maps of X and Y (the generalization method used and the map scale of X' and Y' are S1-1).

Assume there are m nodes (on the side E of X) that have value 1 in the column S1-1 of the file X-GNT. The set of the nodes can be denoted as $X_e\{x_1, x_2, x_3, \dots, x_m\}$. Assume there are n nodes (on the side W of Y) with value 1 in the same column of the

file Y-GNT. The set of the nodes is denoted as $Y_w\{y_1, y_2, y_3, \dots y_n\}$. If map X' and Y' happen to be compatible with each other, the numbers of the nodes which have the same S1-1 value in X-GNT and Y-GNT should be equal, i.e., $m = n$. At the same time, a matching should exist between two node sets, i.e., for each node in X_e , there is one (and only one) partner in Y_w . It can be depicted as: $x_1 - y_1, x_2 - y_2, \dots x_m - y_n$. In this case, we say that the matching is a perfect matching. A perfect matching, in this context, means that all nodes in both X_e and Y_w are saturated. Inversely, if a perfect matching exists between set X_e and set Y_w , map X' and Y' are consistent with each other.

If the matching between X_e and Y_w is not perfect, map X' and Y' will not be consistent with each other. In this case, there are two kinds of possibilities: $m \neq n$; or though $m = n$, some nodes in one set do not have matching nodes in the other set. In either case, some nodes either in X_e or in Y_w or both are not saturated.

Assume X is the current tile. While X is generalized, Y-GNT will be checked. For each node in Y_w , if it is not saturated, its partner node (in X-GNT file) will be retrieved and added in X_e . Finally, all of the nodes in Y_w will become saturated, and the length of X_e will be greater than or equal to the length of Y_w , i.e., $m \geq n$. In this case, we say that the matching is Y_w -complete. A Y_w -complete matching means that all of the map objects on tile Y' which intersect with the map boundary W have their partner objects on tile X' .

When tile Y is currently processed, the above procedure will be carried out

inversely, i.e., all nodes in X_e will become saturated, and an X_e -complete matching will be obtained. It means that all of the map objects which extend across map boundary E on tile X' have their partner objects on tile Y' . Integrating above two procedures, a perfect matching between X_e and Y_w will be obtained, i.e., after both maps are generalized, they will be consistent with each other.

From the discussion above, it can be established that two core tasks are involved in building up the system: one is the determination of adjacent relationships both between tiles and between sides of the tiles; the other is the establishment of GNT files. They will be discussed in section 4.3 and 4.4 respectively.

4.3 The adjacent-relationship reference files

The task of the adjacent-relationship reference files is to maintain the adjacent relationships. The determination of adjacent relationships is the first step in establishing the GNT file system. It is the prerequisite for the implementation of a GNT file system. As mentioned previously, two kinds of relationships need to be determined: the adjacent relationship between tiles (T-T) and the neighbour relationship between sides of the tiles (S-S). The T-T relationship is specific for each database. Conversely, the S-S relationship is universal for every tile. The S-S relationship can be determined by a set of rules.

4.3.1 The definition of the T-T relationship

For an arbitrary tile X in a tile set, its neighbours (adjacent tiles) are defined as the tiles that share a geographic boundary with X. In a spatial database, the geographic boundary shared may be represented as a straight line, an arc, or only a point. A tile can have at most eight neighbours (see Figure 4.6).

J45	J46	J47	J48 ...
I45	I46	I47	I48 ...
H45	H46	H47	H48 ...
G45	G46	G47	G48 ...
...	...	...	...

Figure 4.6 The adjacent relationship among tiles

The eight neighbours of a tile are respectively named as north-west (NW) neighbour, north (N) neighbour, north-east (NE), east (E), south-east (SE), south (S), south-west (SW), and west (W) neighbour. The adjacent relationships among tiles are illustrated in Figure 4.6. In Figure 4.6, tile J45, for example, has three neighbours: J46, I46 and I45; tile I46 has eight neighbours: J45, J46, J47, I47, H47, H46, H45 and I45. To tile I46, tile J45 is its north-west (NW) neighbour; J46 is its north (N) neighbour; I45 is its west (W) neighbour. When we deal with the consistent issue of tile I46, all of its eight neighbour tiles should be taken into consideration.

4.3.2 The creation of the T-T relationship file

Based upon the definition given in section 4.3.1, a T-T relationship reference file can be created for each specific database. A T-T relationship reference file (simply called T-T file) can be depicted as a relational table, in which the neighbour tiles for each tile are recorded. A T-T file is illustrated in Table 4.2, in which the item (column) Tile includes all of the tiles involved in the database. For each tile, its NW neighbour, N neighbour, W neighbour are recorded in the column NW, N, W respectively.

No.	Tile	NW	N	NE	E	SE	S	SW	W
1	J45	0	0	0	J46	I46	I45	0	0
2	J46	0	0	0	J47	I47	I46	I45	J45
3	J47	0	0	0	J48	I48	I47	I46	J46
:	:	:	:	:	:	:	:	:	:
:	I46	J45	J46	J47	I47	H47	H46	H45	I45
:	:	:	:	:	:	:	:	:	:

Table 4.2 An example of the T-T relationship reference file

A T-T file can be created automatically or manually. If all tiles in a GIS are represented by using the geographic coordinate system (the system of longitude and latitude coordinates), or they are all projected in the same Cartesian coordinate system, the tiles can be joined together in some order without any gaps. In this case, the adjacent relationships among them can be created by using an automated approach. The algorithm for automatically creating the reference file is described as follows:

Suppose there are n tiles in a database. They are denoted as $X_1, X_2, \dots, X_i, \dots, X_n$.

The four tics of tile X_i are denoted as T_{i-1} , T_{i-2} , T_{i-3} and T_{i-4} respectively.

Algorithm:

```

begin
  for ( $i = 1; i \leq n; i++$ )
  {
     $k = 0$ ;
    read  $T_{i-1}$ ;
    read  $T_{i-2}$ ;
    read  $T_{i-3}$ ;
    read  $T_{i-4}$ ;
    for ( $j = 1; j \leq n; j++$ )
    {
      if  $k \geq 8$ 
        break; (leaping out the for-loop of  $j$ )
      read  $T_{j-1}$ ;
      read  $T_{j-2}$ ;
      read  $T_{j-3}$ ;
      read  $T_{j-4}$ ;
      if ( $T_{i-1} = T_{j-3}$ ) (i.e.  $T_{i-1}$  coincides with  $T_{j-3}$ , the same below)
      {
        output:  $X_j$  is the NW neighbour of  $X_i$ ;
         $k = k+1$ ;
        continue;
      }
      if ( $T_{i-1} = T_{j-4}$  and  $T_{i-2} = T_{j-3}$ )
      {
        output:  $X_j$  is the N neighbour of  $X_i$ ;
         $k = k+1$ ;
        continue;
      }

      if ( $T_{i-1} = T_{j-2}$  and  $T_{i-4} = T_{j-3}$ )
      {
        output:  $X_j$  is the W neighbour of  $X_i$ ;
         $k = k+1$ ;
        continue;
      }
    }
  }

```

```

    }
    if ( $T_{i-2} = T_{j-4}$ )
    {
        output:  $X_j$  is the NE neighbour of  $X_i$ ;
         $k = k+1$ ;
        continue;
    }
    if ( $T_{i-2} = T_{j-1}$  and  $T_{i-3} = T_{j-4}$ )
    {
        output:  $X_j$  is the E neighbour of  $X_i$ ;
         $k = k+1$ ;
        continue;
    }
    if ( $T_{i-3} = T_{j-1}$ )
    {
        output:  $X_j$  is the SE neighbour of  $X_i$ ;
         $k = k+1$ ;
        continue;
    }
    if ( $T_{i-3} = T_{j-2}$  and  $T_{i-4} = T_{j-1}$ )
    {
        output:  $X_j$  is the S neighbour of  $X_i$ ;
         $k = k+1$ ;
        continue;
    }
    if ( $T_{i-4} = T_{j-2}$ )
    {
        output:  $X_j$  is the SW neighbour of  $X_i$ ;
         $k = k+1$ ;
        continue;
    }
}
end

```

If the tiles in the GIS are projected in different Cartesian coordinate systems, the T-T file cannot be created in an automated manner. For example, in Canada, all topographic maps at scales of 1:250,000 and larger are drawn on the UTM projection

(Nicholson and Sebert, 1981). The whole country is divided into several zones which are 6° of longitude wide and run from Pole to Pole. Each zone is projected individually. For the tiles which fall in the same zone, the adjacent relationship between them can be created using the algorithm described above. For the tiles which are located near the edges of the zones, some of their geographical neighbours may fall in adjacent zones. Without some projection transformation, they cannot be joined together to form a seamless data set. In this case, the adjacent relationship between them cannot be automatically established. In some countries (e.g., United States and China), the Lambert Conformal Conic Projection is adopted as the official projection for some small scale topographic map series (such as 1:1,000,000 topographic map series). In order to obtain a more accurate projection (less distortion), usually each map sheet is projected individually. Thus, the geographically adjacent tiles cannot be directly joined together. In this case, we can create the relationship reference manually according to the index of the map series. If the number of tiles is not large, or the coordinate systems of the tiles are diverse, the second method may be more efficient.

4. 3. 3 The definition and the determination of the S-S relationship

While a tile is processed, every side of its neighbour tiles does not need to be checked. Only those sides which are related to (or neighbour with) the current tile will be checked. In order to identify the right sides which are involved in the processing of the current tile, we need to further define the neighbour relationship between the sides of

the adjacent tiles, i.e., the S-S relationship.

Before defining the S-S relationship, a particular case needs to be considered: when a map object extends across the tic point, it intersects with two map boundaries at the same time (see Figure 4.7 a). In this case, the node is encoded as follows:

if the node coincides with (or falls in a given buffer of) Tic 1, its Node-ID is 9001;

if the node coincides with (or falls in a given buffer of) Tic 2, its Node-ID is 9002;

if the node coincides with (or falls in a given buffer of) Tic 3, its Node-ID is 9003;

if the node coincides with (or falls in a given buffer of) Tic 4, its Node-ID is 9004;

Figure 4.7 b illustrates the encoding rules for the nodes in question.

(a) The map object extends across the tic point.

(b) The rules for encoding the nodes that coincide with the tic points.

Figure 4.7 The rules for a particular case: the nodes coincide with tic points.

Based on the definitions and rules described above, the neighbour sides of a given tile can be precisely determined. Suppose N, E, S and W are the four sides of an

arbitrary tile X; Tic1, Tic2, Tic3 and Tic4 are the four tics of X. Their neighbour sides (including tics) are respectively defined as follows (Table 4.3):

Side and tic of X	Neighbour side and tic	Neighbour tile which includes the neighbour side and tic
N (1001-1999)	S (3001-3999)	N neighbour tile of X
E (2001-2999)	W (4001-4999)	E neighbour tile of X
S (3001-3999)	N (1001-1999)	S neighbour tile of X
W (4001-4999)	E (2001-2999)	W neighbour tile of X
Tic1 (9001)	Tic3 (9003) Tic2 (9002) Tic4 (9004)	NW neighbour tile of X W neighbour tile of X N neighbour tile of X
Tic2 (9002)	Tic4 (9004) Tic3 (9003) Tic1 (9001)	NE neighbour tile of X N neighbour tile of X E neighbour tile of X
Tic3 (9003)	Tic1 (9001) Tic4 (9004) Tic2 (9002)	SE neighbour tile of X E neighbour tile of X S neighbour tile of X
Tic4 (9004)	Tic2 (9002) Tic1 (9001) Tic3 (9003)	SW neighbour tile of X S neighbour tile of X W neighbour tile of X

Table 4.3 the definition of neighbour sides and tics

In an S-S relationship file, only the relative neighbour relationship between sides and tics are defined, no specific tile is concerned. Therefore, the definition is universal. It defines the S-S relationship between any adjacent tiles. In other words, it is not specified in the database.

4. 4 Generalization track (GNT) files

The structure of a GNT file has been illustrated in Table 4.1. From Table 4.1, we can find that the information involved in a GNT file may be divided into two parts. One part is the information which is used to create the matchings between map objects of adjacent tiles. This information is maintained in the columns from Node-ID to Obj-ID. The other part is the information about the generalization history at each given target scale, which is maintained in the item S1-1, S1-2 ...

The matchings between map objects are established through the creation of the matchings between their borderline nodes. The matchings between borderline nodes can be created by using either their spatial locations or the attribute data. In a GIS, each object has an exact spatial location. This is the common characteristic of all GIS. The attribute data, on the other hand, are specific to the database. They may be different from database to database. In attribute data, only those items which are able to identify the geographic entity (not the map object) can be used to create the matchings. Some databases may include such kinds of items, but some others may not. Therefore, the spatial matching between two objects is more precise and more unique. The method of matching creation which is based on the spatial property of the map objects is more universal to any GIS.

In this research, the spatial matching is used as the key matching because the goal of the research is to develop a general solution for the problems. The attribute matching,

if it exists, is used as a complementary matching to verify the spatial matching.

According to the properties of the application in this research, the spatial matching can be created through three approaches: index matching, coordinate matching and distance matching. The adoption of an approach in a specific application depends on the conditions of the data, e.g., the projection of the data sets, the quality and the accuracy of the data.

4. 4. 1 The creation of the borderline nodes

Features which extend across the map boundaries will intersect the map boundaries at some points. These intersection points, i.e., borderline nodes, are recorded in the GNT file. They play the key role in creating the spatial matching between map objects.

In area feature layers, the map boundaries (some or all) are always involved as long as any polygon on the tile extends across the map boundary. In this case, the map boundaries are part of the polygon boundaries and are used to close the outside polygon. In these layers, the borderline nodes, i.e., the intersection points of polygon lines and the map boundaries, exist already. They can be read directly from the node file of the coverage. Hence, they do not need to be further discussed.

There are usually no map boundaries associated with the linear feature layers. In this case, the borderline nodes need to be created through computation.

Theoretically, if the database is perfectly correct, the features which extend

across the map border lines will exactly end on the map boundaries. In other words, the end (or the start) points of those map objects will lie on the map boundaries. If so, the creation of the borderline nodes will become simple: we just need to read the nodes from the node file of the coverage and identify those that coincide with the map boundaries -- the borderline nodes we want to create.

However, in practice, those features often do not exactly reach the boundary, or sometimes, some of them may over shoot. In either case, the end points of the objects are not right on the map boundaries. Thus, the computer cannot identify the map objects which extend across two tiles. In this case, the borderline nodes have to be re-created in some way.

A direct approach is to find the intersection of each object line segment and the each map boundary segment, then test whether the intersect point is on both line segments. If the point is on both segments, it is a borderline node. Otherwise, it should be discarded.

This computation of line intersection point is an essential part of many vector mode systems, and lies at the basis of all clipping algorithms (Clarke, 1990). Usually, it needs a large amount of computation work. Though the algorithm can be optimized, a big computation job is still unavoidable.

In this application, the end points of the objects which extend across the map boundaries exist already. As discussed, these points should, theoretically, be the borderline nodes. Proceeding from finding the relationship between these points and the

map boundaries, an alternative algorithm can be developed. In this algorithm, the distance between the point and the map boundary segment is computed first, then the distance is evaluated. If the distance is less than a given threshold, and if the closest point along the boundary line segment to the test point is on the segment in question, the point is considered as the borderline node. Otherwise, the point should be discarded. The procedures of the algorithm are described as follows:

Suppose u is a line determined by two adjacent vertices of the map boundary $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ (the line segment between point P_1 and P_2 is denoted as P_1P_2). $P_0(x_0, y_0)$ is an end point (node) of a map object. Passing through point P_0 , a line v can be drawn, which is perpendicular to line u , and intersects u at $C(x_c, y_c)$. Point C is the closest point along u to point P_0 (see Figure 4. 8).

Figure 4. 8 Passing through point P_0 , draw a line v , which is perpendicular to line u , and intersects u at $C(x_c, y_c)$.

Our task is to determine point $C(x_c, y_c)$, compute the distance from P_0 to C , and

if the distance is less than the given threshold, test whether C is on the segment P_1P_2 or not.

Since both P_1 and P_2 are on line u , the equation of line u can be created as:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \quad 1a$$

Arranging it, we obtain:

$$y = \left(\frac{y_2 - y_1}{x_2 - x_1} \right) (x - x_1) + y_1 \quad 1b$$

Let m_1 be the slope of line u , then

$$m_1 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right)$$

Let $A = x_2 - x_1$, $B = y_2 - y_1$, and substitute for $x_2 - x_1$, $y_2 - y_1$, equation 1b becomes:

$$y = \left(\frac{B}{A} \right) (x - x_1) + y_1 \quad 1c$$

Where $m_1 = \frac{B}{A}$

Let m_2 be the slope of line v . $\therefore v \perp u$

$$\therefore m_2 = \frac{-1}{m_1} = \frac{-A}{B}$$

Similarly, we can obtain the equation of line v :

$$y = -\left(\frac{A}{B} \right) (x - x_0) + y_0 \quad 2c$$

$\therefore$ point $C(x_c, y_c)$ lies on both u and v ,

$$\therefore y_c = \left(\frac{B}{A}\right)(x_c - x_1) + y_1 \quad 1d$$

$$\text{and } y_c = -\left(\frac{A}{B}\right)(x_c - x_0) + y_0 \quad 2d$$

By subtracting equation 2d from 1d,

$$\left(\frac{B}{A}\right)(x_c - x_1) + y_1 + \left(\frac{A}{B}\right)(x_c - x_0) - y_0 = 0 \quad 3a$$

Arranging equation 3a, we obtain:

$$x_c = \frac{A^2 x_0 + AB(y_0 - y_1) + B^2 x_1}{A^2 + B^2} \quad 4a$$

Similarly, by solving simultaneous equations 1d and 2d for y_c , we can obtain:

$$y_c = \frac{A^2 y_1 + AB(x_0 - x_1) + B^2 y_0}{A^2 + B^2} \quad 4b$$

Where $A = x_2 - x_1$; $B = y_2 - y_1$.

Formula 4a and 4b give the coordinates of point $C(x_c, y_c)$. In these two formulas, $A^2 + B^2$ is the squared distance between point P_1 and P_2 . Since the length of the segment $P_1 P_2$ is not equal to 0 (i.e., $A^2 + B^2 \neq 0$), formula 4a and 4b will always produce values, no matter what the slope of the line u is.

Now, the distance from $P_0(x_0, y_0)$ to line u can be determined by computing the distance between $P_0(x_0, y_0)$ and $C(x_c, y_c)$:

$$d = \sqrt{(x_c - x_0)^2 + (y_c - y_0)^2} \quad 5a$$

If $d = 0$, it means point P_0 is on line u (not necessary to be on the segment $P_1_P_2$). As discussed previously, if the database is perfectly accurate, the end point of an object which extends across two tiles should lie right on the map boundary, i.e., if the object extends across segment $P_1_P_2$, we should have: $d = 0$. Due to some small differences in the coordinate values, the end point may deviate from the map boundary, i.e., $d \neq 0$. In order to identify this kind of node, a tolerance needs to be used. The determination of the tolerance depends on the map unit and the accuracy of the database.

Let r be the tolerance. For a point $P_0(x_0, y_0)$, if $d > r$, point $C(x_c, y_c)$ is not considered as a borderline node, and it should be discarded.

If $d \leq r$, point $C(x_c, y_c)$ is considered as the candidate of the borderline node. It is maintained for a further test.

Though point $C(x_c, y_c)$ is on the line u , it may not lie on the segment in question. Figure 4.9 illustrates three possible situations.

In Figure 4.9 (a) and (b), point C lies on the segment (inside or on the end). In these cases, point C is a borderline node. In Figure 4.9 (c), point C is not on the segment, but on the extension of the segment beyond its end points. It is impossible for the node to be the borderline node. Therefore, the point should be discarded.

In order to determine if point C is on the segment $P_1_P_2$, some tests need to be done. A method for testing this is described by Clarke, K. C. (1990). To test for this,

(a) Point C lies on segment P_1P_2 .

(b) Point C coincides with vertex P_2 .

(c) Point C lies on the extension of P_1P_2 .

Figure 4.9 Three possible situations of the intersection point

we merely need to test to see if C lies between x_1 and x_2 , or C lies between y_1 and y_2 . It can be depicted as following algorithm:

Let $\min\{x_1, x_2\} = x_1$, if $x_1 < x_2$;
 $\min\{x_1, x_2\} = x_2$, if $x_2 < x_1$;
 $\max\{x_1, x_2\} = x_1$, if $x_1 > x_2$;
 $\max\{x_1, x_2\} = x_2$, if $x_2 > x_1$;

For $\min\{y_1, y_2\}$ and $\max\{y_1, y_2\}$, we give the similar definitions.

Algorithm:

begin

for each borderline node candidate $C(x_c, y_c)$

if $(x_c \leq \max\{x_1, x_2\}) \ \&\& \ (x_c \geq \min\{x_1, x_2\})$ or $(y_c \leq \max\{y_1, y_2\})$
 $\&\& \ (y_c \geq \min\{y_1, y_2\})$

output: point C is a borderline node;

else

discarding point C
end

If point $C(x_c, y_c)$ is the borderline node, point $P_0(x_0, y_0)$ is substituted by point $C(x_c, y_c)$. Node C and the Obj-ID of the corresponding map object are reported in the GNT file.

In this process, the intersection point C needs to be computed for each node on a tile and every segment of the map boundary. This is a quite big job in which a number of unnecessary computations are involved. The algorithm can be optimized so that neither all of the nodes on a coverage nor every segment of the map boundary is involved in the computation.

Usually, the map boundaries are smooth curves. Sometimes, some boundaries are even straight line (e.g., in the Lambert Conic Projection, the meridians are depicted as straight lines; in the Mercator Projection, both meridians and parallels are projected into straight lines). If the map boundary between two tics is a straight line, the whole boundary is a single segment and two tics will be the end points of the segment. In this case, only one segment for each boundary is involved in the computation.

If the boundary is a curve and consists of more vertices, we can first determine the vertex that is closest to the node. This can be done by using the point-to-point distance algorithm (formula 5a) to calculate the distance from the node to each vertex of the boundary, then test each distance to find the closest vertex to the node. Two segments which are linked immediately by the closest vertex should be involved in the

computation because one of them must be the closest segment to the node.

For the nodes on a tile, we do not need to put them all in the queue for the computation. We can just select those ones which are close to the corresponding map boundary for the computation and ignore the rest. The following is the process for selecting the close nodes.

Let $\min\{y_1, y_2 \dots y_k\}(N)$ be the smallest value of all y coordinates of the vertices on boundary N (including Tic1 and Tic2).

Let $\min\{x_1, x_2 \dots x_L\}(E)$ be the smallest value of all x coordinates of the vertices on boundary E (including Tic2 and Tic3).

Let $\max\{y_1, y_2 \dots y_m\}(S)$ be the largest value of all y coordinates of the vertices on boundary S (including Tic3 and Tic4).

Let $\max\{x_1, x_2 \dots x_n\}(W)$ be the largest value of all x coordinates of the vertices on boundary W (including Tic4 and Tic1).

For boundary N , if the y coordinate of a node meets the condition:

$$y \geq \min\{y_1, y_2 \dots y_k\}(N) - r$$

the node should be selected for the computation. Where, r is the given threshold, which has the same value as the r mentioned previously.

For boundary E , if the x coordinate of a node meets:

$$x \geq \min\{x_1, x_2 \dots x_L\}(E) - r$$

the node should be selected for computation.

For boundary S, if the y coordinate of a node meets:

$$y \leq \max\{y_1, y_2 \dots y_m\}(S) + r$$

the node should be selected for computation.

For boundary W, if the x coordinate of a node meets:

$$x \leq \max\{x_1, x_2 \dots x_n\}(W) + r$$

the node should be selected for computation.

4. 4. 2 The index matching method

Using borderline nodes, we can create the matching between map objects on adjacent tiles. Three approaches can be used to create the matching, of which the index method is perhaps the most simple.

In the index matching method, the created borderline nodes except those ones which coincide with the tics are sorted into a specific sequence. The nodes which lie on boundary N and S are sorted according to their x coordinates and given the Node-ID value 1001-1999, 3001-3999 (the value increasing with the x coordinates) respectively. The nodes which lie on boundary E and W are sorted according to their y coordinates and given the Node-ID value 2001-2999, 4001-4999 respectively. The borderline nodes which coincide with (or fall in a buffer of) the tics are given the Node-ID value 9001, 9002, 9003 and 9004 respectively (see Figure 4.7 b). The Node-ID value of each node will be used as the index number of the node for creating the matching. The structure of the GNT file for index matching method is shown in Table 4.4.

Node#	Node-ID	x	y	Obj-ID	S1-1	S1-2	
1	1001	:	:	28	1	1	:
2	1002	:	:	73	1	0	:
3	1003			14	0	0	:
:	:			:	:	:	
47	2001			132	1	1	
48	2002			129	0	0	
:	:			:	:	:	
127	9003			97	1	1	

Table 4. 4 The structure of the GNT file for index matching method

If the original data are perfectly correct in the database, i.e., the adjacent tiles are compatible with each other, for each node in a GNT file we can find one (and only one) partner node in the corresponding neighbour GNT file. Based on the correspondence relationships (S-S relationship) given in Table 4.3, the matching between two nodes in this method is created according to their index number (Node-ID value). The matchings are illustrated in Figure 4.10.

Figure 4. 10 For each node in a GNT file, a partner node can be found according to their index numbers in the corresponding neighbour GNT file.

More attention should be paid to the nodes which have the index number from 9001 to 9004. Those nodes coincide with the tics. Their partners may be reported in different neighbour GNT files from other nodes, depending on the specific situation (this will be discussed in section 4.5). From Figure 4.10, we can find that the node with index number 9002 in X-GNT has its partner in the GNT file of the NE neighbour of X (index number 9004), instead of the N neighbour of X (index number 9002).

In this method, the compatibility between original data tiles plays a key role in creating the matching. The spatial locations of the nodes are only used to sort the nodes for creating the proper order of the nodes. They do not directly affect the creation of the matching. Therefore, this method can be used for the tiles which are in the different coordinate systems, or in the different projections.

The limitation of this method is that the original data must be perfectly compatible (not necessarily accurate). If there are some inconsistencies in original data tiles, serious chaos will be introduced into the matchings. One solution for dealing with this problem is to employ the attribute data to verify the created matching or to help create the correct matching between two map objects. If there is not such an attribute item in the database, an alternative method should be adopted.

4. 4. 3 The coordinate matching method

The coordinate matching method uses absolute spatial locations (coordinates) of the nodes to create the partnership between borderline nodes. If the tiles in a database

are projected in the same coordinate system, we can use this method to create the matchings.

The structure of the GNT file for coordinate matching is the same as that for the index matching (see Table 4.4), and the nodes are sorted in the same way as well, i.e., the nodes on side N and S of boundaries are sorted according to their x coordinates; the nodes on side E and W are sorted according to the y coordinates. The partner relationship between the nodes in two GNT files is created by matching their coordinates. If two nodes have the same (x, y) coordinates, a matching (partnership) exists between them. As discussed previously, due to some small differences in the exact coordinate values at points, two partner nodes may not exactly coincide in location. In order to overcome this problem, a buffer needs to be created for each node in the process of creating matching. The tolerance r can be used as the radius of the buffer. Thus, the node in the neighbour GNT file which falls in this buffer is considered as the partner of the node in question. Figure 4.11 illustrates the GNT files in the coordinate matching method and the partnership between the nodes.

In Figure 4.11, the node 3003 in the GNT file of the N neighbour of X has no partner in X-GNT file. This means that the original data tiles in the database may not be compatible with each other.

The creation of the matching between two borderline nodes which coincide with the tics (if they exist) does not need to compare their coordinates, because the partnership between them is precise. Like the index matching method, the matching for

Figure 4.11 The GNT files and the partnership between the nodes in the coordinate matching method ($r = 0.5$).

those nodes can be directly created according to their Node-ID values.

The advantage of this method is that the creation of matchings is based on the exact spatial locations of the nodes. Therefore, if some inconsistencies exist in original data, they do not adversely affect the matchings (see figure 4.11). In addition, the matching can be created without the assistance of the attribute data by using this method. Of course, the attribute data are still useful. If some attribute items are available and can be used to verify the matchings, some problems arising from the inaccuracy of the original data may be remedied.

The limitation of the use of this method is that all tiles must be in the same coordinate system (the unique coordinate system ensures that all tiles can be joined together without gaps). At the same time, the database must be accurate enough, or the edge matching process has to be well done before the data are generalized.

4.4.4 The distance matching method

As an example, the structure of the GNT file for the distance matching method is illustrated in Table 4.1. In this method, the distances from the borderline nodes (except the ones that lie on the tics) to the corresponding origins are calculated and recorded.

The definition of the origins is given in section 4.2, and is illustrated in Figure 4.4

(b). In order to get a convenient reference, the regulation is listed as follows:

The side of boundaries:

N
E
S
W

Origins:

Tic1
Tic3
Tic4
Tic4

The distance from each borderline node on the east side of a tile to the origin (Tic3) is illustrated in Figure 4.12.

Figure 4.12 The distance from each borderline node on the east side of a tile to the origin

The nodes which lie on the same side of the boundaries are sorted according to the distance values in the column Distance. Thus, with the increase of the Node-ID

value, the distance value gets larger. The matchings between nodes on adjacent tiles are created according to their distance values in respective GNT files (see Figure 4.13). A tolerance is still necessary for ignoring the small differences of the distance values in creating the matchings.

Figure 4.13 The matchings are created according to the distances values in respective GNT files.

In the distance method, the matchings are created by using the relative spatial distance (between the nodes and the origin tics), instead of absolute spatial locations. Therefore, it can overcome some of the drawbacks of both the index and the coordinate method. For the database in which the tiles are not in the same coordinate system (e.g., they are in the different projection zones), we can use this method to create the

matching. If the tiles in a database use different projections, this method still works. In this case, a compensation factor needs to be multiplied by the distance values of the neighbour GNT file. This allows the adjacent side of the neighbour tile to be adjusted in length so that the neighbour sides of two tiles have the same length, and the two matching nodes have the comparable distance values. In addition, if the original tiles are not particularly consistent with each other, we still can create the correct matchings by using this method.

For the nodes which coincide with the tics, the matchings are created by using the same procedure as that used in the index method, and no distance comparison is needed for them.

4. 4. 5 Generalization history information

In column S1-1, S1-2 ..., the information about the generalization track is reported for each generalization method at each designed target scale. It provides us with a reference about which features are selected and which ones are eliminated in the generalization processes. This information can be obtained by running GenSystem to carry out the generalization processing on each tile at each target scale before the data are published on the Internet. In a GIS environment, this work can be done in a batch process.

Through the common item Obj-ID in both a GNT file and the corresponding coverage file (such as the PAT file or AAT file in an ARC/INFO coverage), the GNT file

can be linked up with the coverage file. Thus, the information about each object which includes the borderline node can be obtained from the database. While a coverage is generalized by using GenSystem, the map objects which are selected for representation at this scale are marked in a coverage file. This information is transferred to the GNT file through the linkage between them. For each record in a GNT file, if the corresponding object is marked (i.e., the object is selected) at this scale, we put the number 1 into the corresponding S*-* column. If it is not marked, a number 0 is put into this position. This process will be repeated for each method and target scale until the GNT file is completed.

4.5 The method of implementing the GNT system

The function of the GNT system is to adjust the generalization results according to the information coming from the neighbour tiles, and ensure that the generalized tiles are compatible with each other no matter what time each of them is generalized and what the generalization order is.

The GNT file system is established before the data are published on the Internet. The establishment of a GNT file can be automated, just like the creation of the topology of a coverage in a GIS. It is a part of the preparation work for the data to be published on the Internet.

Similar to the strategies for linking models to a GIS (discussed by Batty and Xie, 1994), the GNT system can be strongly coupled on the GenSystem, i.e., it is considered

as an element of the GenSystem and is embedded within it; or it can be loosely coupled with GenSystem, i.e., it is developed as a module and is linked to the GenSystem based on the importation and output of common data.

When users implement GenSystem to carry out a generalization process, they do not need to run the GNT system specially. The GNT module will be invoked by GenSystem automatically (or the GNT module itself is included in GenSystem). Users just need to run GenSystem to carry out the generalization processes. The generalized maps will be compatible with each other and with the ones that will be generalized in some other occasion.

The mechanism of the implementation of the GNT system will be discussed next. The distance matching method is taken as the example for the description.

Suppose I46 (see Figure 4.6) is the current generalized tile; there are m nodes on the boundary N of I46 and the Node-ID values of them are $n_1, n_2, n_3 \dots n_m$ ($n_1 = 1001, n_2 = 1002 \dots$); the corresponding distance values are $d_1, d_2, d_3 \dots d_m$. Consider the generalization process using method 1 at scale 1 (the generalization information is maintained in item S1-1):

- (1) Opening the I46-GNT file and examining the records in it one by one.

For node n_i ($i = 1, 2, 3 \dots m-1$), If the S1-1 value is 1, the node passes and the operation turns to the next record to examine n_{i+1} .

If the S1-1 value is 0, the neighbour GNT files need to be checked.

The neighbour GNT files are defined in the F-F reference file (Table 4.2) and the

neighbour sides are defined in the S-S reference file (Table 4.3). For the boundary N of I46, the neighbour side is boundary S of J46. Therefore, the matching should be created between the nodes which have Node-ID values 1001-1999 in I46-GNT and the nodes which have Node-ID values 3001-3999 in J46-GNT.

(2) Creating the matching.

Suppose there are k nodes on the boundary S of J46; the Node-ID values of them are $s_1, s_2, s_3 \dots s_k$ ($s_1 = 3001, s_2 = 3002 \dots$); the distance values are respectively $ds_1, ds_2, ds_3 \dots ds_k$; the given tolerance is r . Now, for each node n_i (on N of I46), we want to find its partner on the S of J46.

Algorithm:

```

begin
  for (i = 1; i ≤ m; i++)
  {
    First = 1;
    Last = k;
    Found = False;
    Lower =  $d_i - r$ ;
    Upper =  $d_i + r$ ;
    j = i;

    while ((First ≤ Last) and (Found == False))
    {
      if (Lower ≤  $ds_j$  ≤ Upper)
        Found = True;
      else if ( $ds_j$  ≤ Lower)
      {
        First = j + 1;
        j = First;
      }
      else if ( $ds_j$  ≥ upper)

```

```

        {
            Last = j - 1;
            j = Last;
        }
    }

    if (Found == True)
        output: node  $s_j$  is the partner of node  $n_i$ ;
    else
        output: no partner for node  $n_i$  is found;
    }
end

```

This algorithm guarantees the identification of the partner for node n_i if the partner exists in the given buffer of n_i . In this algorithm, the searching depth (i.e., the number of nodes which are searched in the process of finding the partner for a current node) is determined by the condition of the source data. If the source data sets are perfectly consistent with each other, the value of m should be equal to the value of k (i.e., $m = k$) and node s_i should be the partner of node n_i . In this case, the searching depth is 1. Inversely, if the source data sets are poor in consistency, a relatively larger search depth will result.

Some attribute data (if they exist in the database) may be used to verify the matching. Those attribute items must be able to identify the geographic entity (not a map object). For example, a river that extends across two map tiles is represented as two map objects in the database. If an item exists, in which the attribute value shared by two map objects can uniquely identify this river (such as the name, or the geographic code of the river), this item can be used to verify or correct the matching.

If no partner can be found in the search described above, there are two kinds of possibilities: the source data sets in the database are not consistent with each other and the partner does not exist; or some edge matching problem exists in the data. Figure 4.14 shows a case where the spatial search approach may fail to create the matching between two map objects. In this case, the attribute information should be used to find the partnership.

(a) A section of the river coincides with the map boundary

(b) The section coinciding with the boundary is lost in the database

Figure 4. 14 A case where the spatial searching approach may fail to create the matching

(3) Checking and adjusting the result of generalization.

If the S1-1 value of node s_1 in J46-GNT is 0 as well, it means that two tiles are compatible with each other in the representation of this entity. The operation will turn to the next record.

If the S1-1 value of node s_1 is 1, the corresponding map object of the geographic entity in I46 should be added in the collection of selected map features. Thus the two tiles become compatible with each other at this entity. Then, the operation turns to the

next record. The above process will continue until all nodes except the ones that have the Node-ID values from 9001 to 9004 are examined.

(4) Examining the nodes having Node-ID 9001-9004.

If $n_m = 9001$, its partner node may be in the different GNT file from other nodes. According to the T-T and S-S reference files, the possible neighbours are I45-GNT, J45-GNT, and J46-GNT. Figure 4. 15 illustrates the three possible situations.

Figure 4. 15 Three possible situations for the feature which passes over the Tic1 of tile I46.

If the feature extends across I46 and J45 (Figure 4.15 a), the J45-GNT file should be checked and the Node-ID value of the partner should be 9003.

If the feature extends across I46 and I45 (Figure 4.15 b), the I45-GNT file should be checked and the Node-ID value of the partner should be 9002.

If the feature extends across I46 and J46 (Figure 4.15 c), the J46-GNT file should be checked and the Node-ID value of the partner should be 9004.

These three possible neighbour files should be checked in sequence until the partner is found. The process of checking and the adjusting the generalization result is the same as that in the step (3).

4.6 The analysis of the GNT file system

The GNT file system designed above provides a general mechanism for dealing with the inconsistency in generalization processes. Based on this design, more utilities may be developed. For example, the discussion given above mainly focused on the linear features. We can also use this mechanism for consistent generalization of some area features. In this case, the specific arcs instead of borderline nodes will be used to create the matching between two polygons on adjacent tiles. The arcs in question should be the polygon boundaries and at the same time, the map boundaries. Those arcs can be defined by using their borderline nodes and the topological relationships existing in the coverage. The adjustment (or correction) actions may include re-selecting the map objects that were eliminated in the generalization; changing the methods used in the generalization process and so on. The problems involved in the generalization of area features are more complicated. The techniques used in the generalization are diverse. Therefore, it is impossible for this mechanism to fit all cases.

Adding the GNT mechanism in the generalization processes will lead to an increase of the computational time. The time increase can be divided into two parts: the preparation time (for establishing the GNT system) and the implementation time.

The preparation time is spent for establishing the GNT system (off-line). The establishment of the system includes creating the T-T reference file and the S-S reference file for a database, creating borderline nodes and capturing the generalization track information for each method at each target scale. A software package can be developed for creating the GNT system so that the system can be created in or mainly in an automated manner.

The creation of the T-T file and the S-S file is quite easy. It can be done either automatically or manually. The calculations involved in the creation of borderline nodes are no more than that involved in a GIS clipping (or windowing) operation. The capture of the generalization track information is a relatively time consuming task because the generalization process has to be carried out (by running the GenSystem) in each method at each target scale. Fortunately, the operation is not complex so that it can be done by using a batch approach.

The critical factor for data processing through the Internet is the on-line running time. In order to get an outline of the implementation time required for the GNT system, we can conduct a worst case analysis. The distance matching method is still taken as an example for the purpose of this analysis.

Suppose there is a total of n borderline nodes in a GNT file. According to the discussion in section 4.5, the procedures involved in the implementation of the GNT system include: checking the S^{*-} value of each record; if the value is 0, creating the matching between this node and its partner on the corresponding neighbour tile; then

adjusting the result of the generalization if its partner's S^{*-} value is 1.

The worst case for the implementation time is that n objects concerned are all eliminated in the generalization process so that all of them require creating the matching with their partners and all of them need to be adjusted.

For the examination of n nodes, the critical operation is comparison. We need n comparisons in the current GNT file and n comparisons in the neighbour GNT files. For the adjustment of the features, a total of n objects needs to be adjusted (re-selected). In the GenSystem, this means that n objects need to be remarked in a coverage file. The above operations are efficient in terms of computational time.

In the process of creating matchings, the number of operations is determined by the searching depth to a large degree. The critical operation in this process is still comparison. If the data sets are of high quality, we will only need a small searching depth. The perfect situation is that the searching depth for every node is 1. In this case, for each node in the current GNT file, only one node in its neighbour GNT file needs to be compared. We need a total of n comparisons. If the original data sets are not very consistent with each other, a larger searching depth will be required. For example, if the average searching depth is 3, we need $3n$ comparisons. In either case, the total number of the comparisons in this process grows linearly. This can be represented as $O(n)$. Of course, the above analysis is based on the worst case. In a practical implementation, the number of operations required will be much smaller than this.

In order to accelerate the implementation of the GNT system, we can take an

alternative strategy. Firstly, we create a meta file for the database to indicate the implementation status of the GNT system. In this file, the records are created for each tile, one record for each generalization method at each target scale. At the beginning, a value 0 is assigned to each record. After the GNT system is implemented and a tile is generalized by using some method at some scale (e.g., S1-1), the value in the corresponding record becomes 1.

Secondly, when a tile is generalized, the meta file is checked first. If it shows that the GNT system has never been implemented in method 1 at scale 1 (i.e., the value of the record for S1-1 in the meta file is 0), the implementation of the GNT system is the same as that described previously, but the S1-1 value of the objects (in the GNT file) which have been adjusted is updated into 2. When the GNT system is implemented a second time (or after the second time), we just need to check the current GNT file, find out the objects whose S1-1 value is 2, and adjust (re-select) those objects. The neighbour GNT files no longer need to be checked, and the process of matching creation is omitted. Thus, from the second implementation, we only need n operations for examining the nodes and n operations for adjusting the features (the worst case). This will greatly reduce the implementation time of the GNT system. The drawback of this method is that after the first implementation, the generalization processes become separated again. If any changes take place on a tile, its GNT file and all of its neighbours' GNT files have to be renewed.

In general, the advantages of the GNT file system can be summarized into two

points: the users can get consistent results by using this system in the generalization processes because the generalization information of the neighbour tiles is considered during the adjustment processes; the operations required in the on-line implementation are quite limited. Therefore the implementation will be efficient.

One disadvantage is that a relatively big job needs to be done in the preparation stage, i.e., the establishment of the GNT file system. The design of the system is based on the static division of the tiles. This is mainly for the consideration of accelerating the on-line processing across the Internet. For the dynamic area's generalization, e.g., the generalization of an arbitrary window and its neighbour windows (arbitrary as well), a different strategy should be adopted.

As mentioned previously, the discussion in this chapter mainly focuses on line features. For the generalization of area features, more research is needed to deal with the problems involved.

The GNT system developed in this chapter is at the stage of a logical design. The system, including the formulas and the algorithms used to create and implement the system, needs to be tested. This test will be described in Chapter 5.

CHAPTER 5: TEST AND IMPLEMENTATION

In Chapter 4, a GNT file system was designed for the purpose of consistent generalization. A set of formulas and algorithms for creating and implementing the GNT file system was derived, developed or described. In this chapter, most of these will be tested, and the GNT system will be implemented. The test is in two parts: the establishment of the GNT file, and the implementation of the GNT file system.

5.1 The testing environment and methodology

The test will be carried out in the ARC/INFO GIS environment instead of an Internet context. The ARC/INFO GIS is selected because the interface and some tools of GenSystem are developed in the ARC/INFO environment, and the data sets are generalized and output as ARC/INFO coverages. Thus, the generalization results from GenSystem can be directly used in the GNT system. Moreover, the AML utility and the INFO module of ARC/INFO provide simple programming tools. This allows us to do programming and to organize the processes so that the algorithms developed in Chapter 4 can be implemented and the complex applications can be performed automatically. Besides ARC/INFO, the C programming language is used in the test to create GNT files.

The data sets used for the test are two hydrography coverages in an area of

south part of Quebec (from CCRS). The scale of the data sets is 1:20,000. The two coverages (tiles) are spatially adjacent with each other. They were produced by clipping a large coverage at some arbitrary location.

In this test, the distance matching method is taken as the example for the implementation. Two GNT files which include the item Distance (see Table 4. 1 in Chapter 4) will be created, one for each tile. They will be used to perform the adjustment of the generalized data tiles for achieving consistent results. The test will be done at the target scale 1:50,000 by using the Basic Formula method (one of the options in GenSystem for generalizing hydrography features). The method is the same for other target scales and options.

The corresponding work in this test will be done by following the procedures described in section 4. 4 and section 4. 5.

5. 2 The creation of the GNT file

Two source hydrography coverages are named Hydro1 and Hydro2 respectively, and are illustrated in Figure 5. 1. As mentioned previously, the scale of the source data is 1:20,000. This means that the spatial and attribute resolution of the information content is at the scale of 1:20,000, not that of the presentation displayed in Figure 5. 1. As discussed in Chapter 2, in a digital environment, the relationships between data content and display are no longer one to one. One data set can have many different presentations. In Figure 5. 1, the images are arranged to fit the page and to give a proper

view. Their scale is arbitrary. In this chapter, if we talk about the scale of the data, it means the scale of the information content, rather than the scale of the display. For the scale of the presentation, we use the term view scale.

The GNT files which will be created for tile Hydro1 and Hydro2 are named Hyd1-GNT and Hyd2-GNT respectively. According to the definition of the adjacent relationships (T-T relationship and S-S relationship) described in section 4.3, Hydro1 is the N neighbour of Hydro2, and Hydro2 is the S neighbour of Hydro1. The adjacent sides of the boundaries between two tiles are the S side of Hydro1 and the N side of Hydro2. Thus, in the two GNT files, we only need to create the borderline nodes which lie on these two sides respectively.

In the distance matching method, an origin point needs to be defined for the borderline nodes on each map boundary. According to the definition (section 4.2), the origin point for side N is Tic1, and the origin point for side S is Tic4. The coordinates of these tics and some other tics that are concerned in this test are listed as follows:

Hydro1			Hydro2		
Tic	x	y	Tic	x	y
Tic3:	304232.5	5108366.5	Tic1:	285456.0	5108366.0
Tic4:	285457.4	5108367.6	Tic2:	304232.0	5108365.5

To create the borderline nodes, the nodes of all arcs on each coverage need to be evaluated. In order to reduce calculation, a set of nodes that are close to the corresponding sides of the map boundaries can be selected for the computation. The

nodes that are close to side S on Hydro1 and the nodes that are close to side N on Hydro2 are listed in Table 5. 1.

Hydro1			Hydro2		
Node#	x	y	Node#	x	y
1	285477.0	5108475.0	1	285618.9	5108366.5
2	285618.9	5108366.5	2	285701.1	5108289.0
3	285733.0	5108367.0	3	285733.0	5108367.0
4	287140.0	5108589.0	4	286331.0	5108253.0
5	287240.0	5108367.0	5	287240.0	5108367.0
6	289348.0	5108555.0	6	289386.3	5108367.0
7	289379.0	5108599.0	7	289398.0	5108367.0
8	289386.2	5108367.0	8	289424.0	5108330.0
9	289398.4	5108367.0	9	289919.0	5108367.0
10	289919.0	5108367.0	10	291850.5	5108367.0
11	291850.5	5108367.0	11	291997.3	5108366.5
12	291997.2	5108367.0	12	296300.1	5108366.5
13	296300.1	5108367.0	13	297981.0	5108366.5
14	297981.0	5108367.0	14	299732.0	5108366.5
15	299732.0	5108367.0	15	300889.8	5108366.5
16	300889.0	5108366.5	16	301962.5	5108366.5
17	301962.5	5108366.5	17	303203.4	5108366.5
18	303203.4	5108366.5	18	303522.2	5108366.5
19	303522.2	5108366.5			

(a) The nodes which are close to the side S of Hydro1.

(b) The nodes which are close to the side N of Hydro2.

Table 5. 1 The nodes that are selected for the computation on both coverages.

The nodes listed in Table 5. 1 are involved in the computation for creating the borderline nodes. Each node is evaluated according to the distance between this node and the corresponding map boundary segment. In this test, the map boundaries are straight lines. Each boundary is a single line segment, and two tics are the end points of the segment. Therefore, only one segment for each side is involved in the computation. Through computing the closest point of each node along the segment and evaluating the

distance, a set of candidates of borderline nodes is obtained. Those candidates are then further examined to check if they are on the corresponding segments. Only those that lie on the segments are considered as the borderline nodes. The formulas and algorithms used in the computation and examination were described in section 4. 4. 1. The borderline nodes created are listed in Table 5. 2.

Hydro1			Hydro2		
Node#	x	y	Node#	x	y
1	285618.91	5108367.50	1	285618.91	5108366.00
2	285733.00	5108367.50	2	285733.00	5108366.00
3	287240.00	5108367.50	3	287240.00	5108366.00
4	289386.19	5108367.00	4	289386.31	5108366.00
5	289398.41	5108367.00	5	289398.00	5108366.00
6	289919.00	5108367.00	6	289919.00	5108366.00
7	291850.50	5108367.00	7	291850.50	5108366.00
8	291997.19	5108367.00	8	291997.31	5108366.00
9	296300.09	5108366.50	9	296300.09	5108365.50
10	297981.00	5108366.50	10	297981.00	5108365.50
11	299732.00	5108366.50	11	299732.00	5108365.50
12	300889.00	5108366.50	12	300889.81	5108365.50
13	301962.50	5108366.50	13	301962.50	5108365.50
14	303203.41	5108366.50	14	303203.41	5108365.50
15	303522.19	5108366.00	15	303522.19	5108365.50

(a) The borderline nodes on the side S of Hydro1.

(b) The borderline nodes on the side N of Hydro2.

Table 5. 2 The borderline nodes on both coverages.

After the borderline nodes are created, the distance between each borderline node and the corresponding origin tic is calculated. This distance will be used to perform the matching between borderline nodes on two adjacent tiles. The processes described above (the creation of the borderline node and the computation of the distance) have

been programmed in the C language so that those processes become automated (see Appendix A).

A threshold with value 10 (m) is used in the node evaluation. This means that the map objects whose end points fall within a 10 meter distance from the map boundary on the ground will be considered as objects which extend across the map boundary. On a 1:20,000 scale view, 10 meters on the ground is represented as 0.5 millimeter. Usually, this is enough to overcome the problem of small differences in location. In addition to the unit of the coordinates and the scale of the source-data, the accuracy of the data should be considered as a factor for determining the value of the threshold. Generally speaking, if the data are not accurate, a relatively larger threshold should be used.

In order to link the borderline nodes in a GNT file with the corresponding map objects in the coverage, the identifications (IDs) of the objects need to be recorded in the GNT file. In an ARC/INFO coverage, the attribute information about arcs is stored in an AAT (Arc Attribute Table) file. In the AAT files of the source coverages (i.e., Hydro1.aat and Hydro2.aat), an item can be found whose values can uniquely identify each arc. Those values are put in a newly created item named Obj-ID, and used as the map object ID. From this item, the IDs of the map objects which extend across the map boundary can be selected. Those IDs are put in the corresponding GNT file, also in the item Obj-ID. Thus, using item Obj-ID as a key, we can link a GNT file with its coverage.

The last step is to implement GenSystem to generalize the tiles Hydrol and Hydro2, and produce two new tiles at the scale of 1:50,000. Two new tiles are named Gen-hyd1 and Gen-hyd2 respectively and are illustrated in Figure 5. 2. From Figure 5. 2, we can find that some inconsistencies are introduced into the maps during the generalization process.

Coverage Gen-hyd1 and Gen-hyd2 provide us with the generalization information (i.e., which map objects have been selected to represent on the new map). Thus, all information for creating the GNT file has been obtained. The GNT files, Hyd1-GNT and Hyd2-GNT, are created and illustrated in Table 5. 3.

Node#	Node-ID	Distance	Obj-ID	S50k
1	3001	161.50	689	0
2	3002	275.59	682	0
3	3003	1782.59	683	1
4	3004	3928.78	684	1
5	3005	3941.00	685	1
6	3006	4461.59	686	0
7	3007	6393.09	687	1
8	3008	6539.78	688	1
9	3009	10842.69	690	0
10	3010	12523.59	691	0
11	3011	14274.59	692	1
12	3012	15431.59	693	0
13	3013	16505.09	694	1
14	3014	17746.00	695	0
15	3015	18064.78	696	0

(a) Hyd1-GNT

Node#	Node-ID	Distance	Obj-ID	S50k
1	1001	162.91	3	1
2	1002	277.00	2	0
3	1003	1784.00	10	0
4	1004	3930.31	7	1
5	1005	3942.00	5	1
6	1006	4463.00	38	0
7	1007	6394.50	6	1
8	1008	6541.31	4	1
9	1009	10844.09	41	1
10	1010	12525.00	113	1
11	1011	14276.00	19	0
12	1012	15433.81	18	1
13	1013	16506.50	53	0
14	1014	17747.41	12	0
15	1015	18066.19	11	1

(b) Hyd2-GNT

Table 5.3 The GNT files of tile Hydrol and Hydro2.

5.3 The implementation of the GNT file system

In order to mark the map objects that need to be re-selected, an item is added to the AAT file of the source coverage. In Hydrol.aat and Hydro2.aat, an item named Status is added. When item Status is created, its value at each record is 0. During the implementation of the GNT system, if an object needs to be adjusted, the Status value of this object will be replaced with the number 1.

The method of implementation of the GNT system was described in detail in section 4.5. This process consists of three steps: examining each record in the current GNT file, matching between borderline nodes if necessary and marking the map object, and adjusting the generalization result. In this test, the processes of the first two steps

are carried out in the INFO module. Using the INFO programming utility, the processes of these two steps are programmed (see Appendix B). The adjustment of the result (the third step) is carried out by using AML to group the operations into a batch process. The procedures are described as follows:

1. **Examining the current GNT file:** Firstly, tile Hydrol is considered as the current tile. After it is generalized (i.e., coverage Gen-hyd1 is produced), Hyd1-GNT is examined. In the INFO module, Hyd1-GNT is selected and examined record by record. If the S50k value of the record is 1, nothing needs to be done and next record is selected to examine. If the value is 0, (e.g., the record with Node-ID 3001), the matching needs to be performed.

2. **Matching and marking the map object:** Also in the INFO module, Hyd2-GNT is opened. The record which Node-ID value is 1001 (i.e., 3001-2000) is first selected for examination. If the difference between its Distance value and the Distance value of node 3001 in Hyd1-GNT is not larger than the given tolerance (10 meters), the matching is made between them, and node 1001 (in Hyd2-GNT) is the partner of node 3001 (in Hyd1-GNT). If the difference is larger than the given tolerance, node 1001 is discarded and the searching continues.

Once the matching is performed between two nodes, the S50k value of the partner (in Hyd2-GNT) is checked. If this value is 0, Hyd1-GNT is re-opened and the next record is selected for examination. If this value is 1, the map object identified by the Obj-ID value of this record in Hyd1-GNT should be adjusted (i.e., re-selected).

Using Obj-ID as the key, the record of this map object in Hydro1.aat can be identified.

This object is marked by putting the number 1 in its Status item.

3. Adjusting the generalization result: After all records in Hyd1-GNT are examined, i.e., all map objects which need to be adjusted are marked in Hydro1.aat, the adjustment process can be carried out. In the Arcedit module, all arcs which have been marked can be selected according to their Status value (the arcs that have been marked have Status value 1; the others' Status value remains 0). Those arcs are then put in coverage Gen-hyd1. The result of this process is that the number of map objects in tile Gen-hyd1 which extend across boundary S is not less than that in tile Gen-hyd2 for objects that extend across boundary N. These operations can be grouped into a batch process by using AML:

Adjust.aml

```
arcedit
&setvar name1 = [response 'Enter the source tile name']
mapextent %name1%
edit %name1%
editfeature arc
select status = 1
&setvar name2 = [response 'Enter the generalized tile name']
put %name2%
yes
quit
```

Running this AML program in the Arc environment and entering Hydro1 as the source tile name, Gen-hyd1 as the generalized tile name, we can get the adjusted coverage. This coverage is named Adj-hyd1 and is illustrated at the top of Figure 5. 3.

When tile Hydro2 is treated as the current tile, the procedures of the implementation are similar to those described above, but the current GNT file and the neighbour GNT file used in the process should be transposed, i.e. Hyd2-GNT is the current GNT and Hyd1-GNT is the neighbour GNT. Besides this, when doing the matching, the Node-ID value of the neighbour GNT (Hyd1-GNT) equals the Node-ID value of the current GNT (Hyd2-GNT) plus 2000 (see the program on Appendix B).

The adjusted coverage is named Adj-hyd2. It is illustrated in Figure 5. 3 (bottom). Comparing the two maps illustrated in Figure 5. 3, we find that they have become consistent with each other.

If the source tiles are not consistent with each other, or if some map objects happen to end near the map boundary but do not go across the boundary, the number of borderline nodes on two adjacent sides may not be the same. In order to examine if the algorithm works in this case, let us delete node 6 from Table 5. 1 (b) and at the same time, deleted the corresponding arc on tile Hydro2. Thus tile Hydro1 and Hydro2 are no longer compatible with each other. The new GNT file of new Hydro2 is illustrated in Table 5. 4, and the new coverage (i.e., new Hydro2) is illustrated in Figure 5.4 (top).

Repeating the implementation process described above, the new results are achieved. The results are illustrated in Figure 5. 4 (middle and bottom). Comparing the coverages at middle and bottom of Figure 5. 4 with those in Figure 5. 3, we can find that both are the same. This indicates that the algorithm works in this situation and the matchings created in the process are correct. The matchings can be correctly created in

this case because the algorithm allows the search to be done in different directions (up or down, depending on the situation) until the partner node is found (if it exists).

Node#	Node-ID	Distance	Obj-ID	S50k
1	1001	162.91	3	1
2	1002	277.00	2	0
3	1003	1784.00	10	0
4	1004	3930.31	7	1
5	1005	3942.00	5	1
6	1006	6394.50	6	1
7	1007	6541.31	4	1
8	1008	10844.09	41	1
9	1009	12525.00	113	1
10	1010	14276.00	19	0
11	1011	15433.81	18	1
12	1012	16506.50	53	0
13	1013	17747.41	12	0
14	1014	18066.19	11	1

Table 5. 4 The new GNT file of new Hydro2

5. 4 Discussion

The test described in this chapter covers the main process of the mechanism designed in Chapter 4. It includes the creation of the GNT file (creating borderline nodes, calculating distance, capturing generalization information etc.) and the implementation of the system (examining borderline nodes, creating matchings, marking map objects and adjusting the generalization results). The formulas derived in Chapter 4 are programmed and tested, and most algorithms are implemented. By using the GNT mechanism, inconsistencies introduced in the process of generalization are eliminated. The expected result is achieved.

The adjacent relationship mechanism (i.e., creating the T-T relationship and S-S relationship, and automatically looking up adjacent tiles and the neighbour sides) is not covered. On the one hand, the procedures of this process are precise and relatively simple. They do not need to be tested. On the other hand, the implementation of this process needs a number of data sets to be involved. The collection of proper data sets is a difficult task and beyond the scope of this research.

The index matching method and the coordinate matching method are not implemented either. The implementation of these two methods is similar to that described in this chapter.

In this test, only two tiles are involved. Therefore, it is impossible for the cases involved in the test to cover all variations of the problems which might be encountered. The purpose of this test is to verify the concept of the design, and to test the formulas and the algorithms included in the design. From this perspective, the expected goal has been achieved.

The implementation of the system has been carried out in the ARC/INFO environment. Due to the limitations of ARC/INFO in terms of programming, this process cannot be completely automated. To develop a practical GNT system and enable it fully to communicate with GenSystem, a more powerful language, such as C, C++ or Java should be used. This is beyond the scope of this thesis, but the logical design and approach have been developed and implementation should encounter no serious problems.

Figure 5.1: Source data tiles: Hydro1 (top), Hydro2 (bottom).

Figure 5.2: Generalized data tiles (about 40% reduction)

Top: Gen-hyd1

Bottom: Gen-hyd2

Figure 5.3: Adjusted data tiles: Adj-hyd1 (top), Adj-hyd2 (bottom).

Figure 5.4: New source tile and new adjusted tiles
Top: new Hydro2
Middle: new Adj-hyd1
Bottom: new Adj-hyd2

CHAPTER 6 SUMMARY AND CONCLUSION

In the previous five chapters, several important aspects of spatial data generalization across the Internet have been explored and studied. These include:

- the characteristics, purposes and issues of generalization both in conventional cartography and in a digital environment.
- the access, dissemination and consistent generalization of spatial data in an Internet context.
- the development, analysis and testing of a mechanism (named the GNT system) for the purpose of consistent generalization of spatial data.

As pointed out in Chapter 1, the purpose of this research is not to develop a whole generalization application for the Internet, but to focus on the study of the issues for geo-spatial data handling. First, the concept of generalization was discussed (Chapter 2). Then, the process of spatial data access, the method and the problems of spatial data generalization in an Internet environment were explored (Chapter 3). After this, the research focused on the development of the GNT system (Chapter 4). The system is developed for carrying out consistent generalization across the Internet. This is the core contribution of the research.

Generalization is a central concept in spatial data processing. In conventional cartography, the task of generalization is to establish a proper model of reality for the

specific purpose and to make the representation legible. In a digital environment, the purposes of generalization can be viewed from three aspects: data volume reduction; spatial resolution reduction of the information content (model-oriented generalization or database generalization); and geometric characteristic simplification for visual purposes (view-oriented generalization).

Database generalization aims to deal with the issues of abstraction of reality. It is usually a pre-process for view generalization. The task of view generalization is to make the presentation of the information contents legible. In a digital context, the processes of database generalization and the processes of view generalization are more independent from each other than those in traditional cartography. However, they should not be separated completely. The procedures for different purposes are often related to each other. Based on this point, an integrated view of generalization was discussed (section 2. 2. 5).

The Internet (the WWW utility of it in particular) has become a popular medium for data access and dissemination. The components of the Internet which are relevant to spatial data handling were summarized in section 3.1. These include the computer network, protocols, and the client and server.

A variety of Web applications which are concerned with spatial data can be found on the Internet. Various techniques and methods have been used in these applications. In this study, the applications are sorted into two types according to the mechanism of data access: the static method and the dynamic method. Static access of

data is the first stage of the WWW application. In static access, the data sets are made available to the users by simply locating and accessing the data through the URLs of the data sets. There is no new information being produced or introduced in this process. In a dynamic mechanism, the user can execute a program or access a database in real-time so that the information can be dynamically accessed. The realization of dynamic data access requires some new techniques to be introduced, such as CGI, Java or ActiveX.

Spatial data can be displayed through a Web browser in either the raster mode or the vector mode. In the raster mode, geo-spatial data are stored and transferred in graphic images (mostly in GIF format). Today, a raster image is the most popular data model used in the WWW. Any browser, as long as it is HTML compatible, can directly display a GIF image. The raster method is not efficient in either data transmission or interactive operation. The vector method is object-oriented. In a vector mode, the map elements are treated as individual graphic objects which are associated with attribute information. The vector method is efficient in interactive operations, such as spatial query and searching. A specific software package termed a plug-in is usually required for the display of a vector map.

A generalization system can be brought into the Internet through CGI. In this case, the generalization system becomes an application of CGI. A model for bringing a generalization system (named GenSystem) into the Internet was proposed in section 3.5.

Spatial data usually take up a large volume of memory. Large data volume means

that a relatively long time is required in data processing and data transmission. In an Internet context, time is a key factor which must be considered. In order to facilitate data application and to reduce the time of data access, a large data package should be avoided. A large coverage may be divided into smaller tiles.

The problem is that after being generalized, those tiles may no longer be compatible with each other. The reason for the resulting inconsistency is that the generalization processes are isolated from tile to tile. In order to overcome this problem, the relationships between the map objects which go across the map boundaries on adjacent tiles must be considered in the process of generalization. Two possible approaches allow us to consider these relationships in generalization. One approach is to evaluate not only the map objects which extend across the map boundaries on the current processed tile, but also their partner objects on the neighbouring tiles. The other approach is to create a generalization track file system in advance. When a tile is generalized, the generalization information of its neighbour tiles can be referred through running the system. The first approach does not need much preparation work, but takes a longer time to run because the evaluation is carried out on-line. The second approach needs more preparation work, but it is efficient in terms of on-line running time. Since time is a very important constraint in Internet applications, the second approach is employed in this research. Based on this idea, a GNT system was developed (Chapter 4) for the purpose of consistent generalization. Actually, most applications in an Internet context take a similar strategy, i.e., spending more time on the preparations

for speeding up the on-line data access.

In the GNT system, the adjacent relationships, i.e., the adjacent relationship between tiles (T-T) and the neighbour relationship between sides of the tiles (S-S) are defined by using both a T-T relationship file and an S-S relationship file. The generalization track (GNT) file is designed and used for creating the matchings between the related map objects and maintaining the generalization information. The spatial matching is used as the key matching in this mechanism since the spatial property of the map objects is more universal in a digital context. Different methods are designed for creating the matching. They can be chosen in use according to the condition of the source data. The procedures of implementation of the GNT system were described in section 4. 5.

A test was conducted and described in Chapter 5. Two tiles were used to create and implement the GNT system. Through implementing the GNT system to adjust the generalization results, the inconsistencies which are introduced in the generalization processes are eliminated. A consistent result was achieved.

From the review of the research, it can be concluded that the research objectives set out in Chapter 1 have been achieved. However, consistent generalization across the Internet is a complex process. There are still issues that need to be investigated. Two major issues are described as follows:

1. **Developing an application interface and a set of drivers for the GNT system.** The problems that may be involved and the techniques used in generalization

are diverse. To enable the GNT system to deal with different kinds of problems, a set of drivers needs to be developed, one driver for dealing with one kind of problems. At the same time, an application interface which is used to access GNT system facilities remains to be developed. When a request for dealing with a specific problem is issued (e.g., issued by GenSystem), a corresponding driver will receive it through the interface. This driver will carry out the process to deal with the particular problem. This approach was adopted in OGDI.

2. The implementation of the GNT system in a distributed GIS context.

The GNT system is designed based on the concept of centric GIS. In order to deal with the data which are produced in different GIS, maintained in different computers, and stored in different formats over the Internet, the open GIS technique (such as OGDI) needs to be introduced, and a mechanism (e.g., a metadata system) needs to be created for the implementation of the GNT system. This is an interesting challenge but to achieve this goal, a great deal of further research is required.

BIBLIOGRAPHY

Aronoff, Stanley, 1989, *Geographic Information Systems: A management perspective*, WDL Publications, Ottawa, Canada.

Barber, Cynthia, Robert Cromley and Robert Andrie, 1995, Evaluating alternative line simplification strategies for multiple representations of cartographic lines, *Cartography and Geographic Information Systems*, Vol. 22, No. 4, 276-290.

Batty, Michael and Yichun Xie, 1994, Modeling inside GIS: Part 1. Model structures, exploratory spatial data analysis and aggregation, *Int. J. Geographical Information Systems*, Vol. 8, No. 3, 291-307.

Brassel, Kurt E. and Robert Weibel, 1988, A review and conceptual framework of automated map generalization, *Int. J. Geographical Information Systems*, Vol. 2, No. 3, 229-244.

Bundy, Geraint L., Chris B. Jones and Edmund Furse, 1995, Holistic generalization of large-scale cartographic data, in Muller, J.C., J.P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 106-119.

Buttenfield, Barbara P., 1995, Object-oriented map generalization: modeling and cartographic considerations, in Muller, J.C., J.P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 91-105.

Carr, David F., 1997, Object wars, *Internet World*, February, 66-70.

Carroll, Michael L., 1996, *CyberStrategies, How to Build An Internet-Based Information System*, ITP, New York.

Carroll, Jim and Rick Broadhead, 1994, *Canadian Internet Handbook 1995 Edition*, Prentice Hall Canada Inc., Scarborough, Ontario.

Clarke, Keith C., 1990, *Analytical and Computer Cartography*, Prentice Hall, Englewood Cliffs, New Jersey.

Cola, Lee De, 1994, Simulating and mapping spatial complexity using multi-scale techniques, *Int. J. Geographical Information Systems*, Vol. 8, No. 5, 411-427.

- Cromley, Robert G., 1992, *Digital Cartography*, Prentice Hall, Englewood Cliffs, New Jersey.
- Dong, Changcheng, Paul Luker, Philippa Berry and Hongji Yang, 1995, Analysis and modeling of spatial objects to implement OOP for spatial data generalization in GIS, *International Conference on Object Oriented Information Systems*, Springer, London, 189-199.
- Douglas, David H. and Thomas K. Peucker, 1973, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, *The Canadian Cartographer*, Vol. 10, No. 2, 112-122.
- Douglas, David H., 1990, It makes me so CROSS, in Peuquet, Donna J. and Duane F. Marble (eds.) *Introductory Reading in Geographic Information Systems*, Taylor & Francis, London, 303-307
- Elbert, Bruce & Bobby Martyna, 1994, *Client/Server Computing Architecture, Applications, and Distributed Systems Management*, Artech House, Norwood, MA.
- Flynn, Peter, 1995, *The World Wide Web Handbook, An HTML Guide for Users, Authors and Publishers*, International Thomson Computer Press, London.
- Fortunati, Luciano, 1996, Geo-data server on WWW, *JEC-GI Proceedings*, Vol. 1, 96-99.
- Goodchild, Michael F., 1995, Geographic information systems and geographic research, in Pickles, John, (eds.) *Ground Truth*, The Guildford Press, New York, 31-50.
- Green, David R, 1997, Cartography and the Internet, *The Cartographic Journal*, Vol. 34, No. 1, 23-27.
- Greenstein, Dave, 1996, API trumps CGI, *Internet World*, May, 32-33.
- Grunreich, Dietmar, 1995, development of computer-assisted generalization on the basis of cartographic model theory, in Muller, J.C., J.P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 47-55.
- Guptill, S. C., K. J. Boyko, M. A. Domaratz, R. G. Fegeas, H. J. Rossmeissl, and E. L. Usery, 1990, An enhanced digital line graph design: a feature-based data model for digital spatial data base that represent geographic phenomenon, *U. S. Geological Survey Circular 1048*.

Hageli, M, R. Bosch and J. Wey, 1996, Connecting WWW, GIS and RDBMS, *JEC-GI Proceedings, Vol. 1*, 108-111.

Handley, Mark & Jon Crowcroft, 1995, *The World Wide Web, Beneath the Surf*, UCL Press Limited, London.

Heisser, Michael, Georg Vickus and Johannes Schoppmeyer, 1995, Rule-orientated definition of the small area 'selection' and 'combination' steps of the generalization procedure, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 148-160.

International Cartographic Association (ICA) 1973, *Multilingual Dictionary of technical Terms in Cartography*, Verlag, Franze Steiner GMBH.

Jackson, M. Drew and Timothy L. Taylor, 1996, *The Internet Handbook for Canadian Lawyers*, Carswell, Scarborough, Ontario.

Joao, Elsa Maria, 1995, The importance of quantifying the effects of generalization, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 183-193.

Johnson, Shelby D., 1997, GeoMedia Web Map, *GIS World, July*, 68-70.

Kilpelainen, Tiina and Tapani Sarjakoski, 1995, Incremental generalization for multiple representations of geographical objects, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 209-218.

King, Nelson, 1998, Java's database connection, *Internet World, January*, 79-92.

Krol, Ed and Paula Ferguson, 1995, *The Whole Internet for Windows 95: User's Guide & Catalog*, O'Reilly & Associates Inc., Sebastopol, CA.

Krol, Ed, 1992, *The Whole Internet: User's Guide & Catalog*, O'Reilly & Associates Inc., Sebastopol, CA.

Lane, Elizabeth and Craig Summerhill, 1993, *Internet Primer for Information Professionals: A Basic Guide to Internet Networking technology*, Meckler Ltd. London.

Lee, Dan, 1995, Experiment on formalizing the generalization process, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 219-234.

- Li, Deren, Jianya Gong and Pengde Li, 1994, An integrated object-oriented data model for GIS, *The Canadian Conference on GIS, Proceedings, Vol. 1*, 701-711.
- Li, Zhilin and Stan Openshaw, 1993, A natural principle for the objective generalization of digital maps, *Cartography and Geographic Information Systems, Vol. 20, No. 1*, 19-29.
- Li, Zhilin and Stan Openshaw, 1992, Algorithms for automated line generalization based on a natural principle of objective generalization, *Int. J. Geographical Information Systems, Vol. 6, No. 5*, 373-389.
- Limp, W. Fredrick, 1997, Weave maps across the Web, *GIS world, September*, 46-55.
- Lomarcán, Diana L., 1995, Network: the basics, *Computers in Libraries, Vol. 15, No. 3*, 19-24.
- Mackaness, William A., 1994, Knowledge of the synergy of generalization operators in automated map design, *The Canadian Conference on GIS, Proceedings, Vol. 1*, 525-536.
- Mackaness, William A. and M. Kate Beard, 1993, Use of graph theory to support map generalization, *Cartography and Geographic Information Systems, Vol. 20, No. 4*, 210-221.
- Maguire, D. J., 1991, An overview and definition of GIS, in Maguire, David J., Michael F. Goodchild and David W. Rhind, (eds.) *Geographical Information Systems, Principles and Applications, Vol. 1*, New York: Longman Scientific & Technical, 9-20.
- Mark, David M., 1991, Object modeling and phenomenon-based generalization, in Buttenfield, Barbara. P. and Robert. B. McMaster, (eds.) *Map Generalization: Making Rules for Knowledge Representation*, London: Longman, 103-118.
- McCormack, J. E., M. N. Gahegan, S. A. Roberts, J. Hogg and B. S. Hoyle, 1993, Feature-based derivation of drainage networks, *Int. J. Geographical Information Systems, Vol. 7, No. 3*, 263-279.
- McMaster, Robert B., 1995, Knowledge acquisition for cartographic generalization: experimental methods, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 161-179.
- McMaster, Robert B. and K. Stuart Shea, 1992, *Generalization in Digital Cartography*, Association of American Geographers, Washington D. C.

McMaster, Robert B., 1989a, Introduction to 'numerical generalization in cartography', *Cartographica*, Vol. 26, No. 1, 1-6.

McMaster, Robert B., 1989b, The integration of simplification and smoothing algorithms in line generalization, *Cartographica*, Vol. 26, No. 1, 101-121.

McMaster, Robert B. and K. Stuart Shea, 1988, Cartographic generalization in a digital environment: a framework for implementation in a geographic information system, *GIS/LIS' 88, Proceedings*, Vol. 1, San Antonio, 240-249.

McMaster, Robert B., 1987, Automated line generalization, *Cartographica*, Vol. 24, No. 2, 74-111.

Miller, David, 1994, The many faces of the Internet, *Internet World*, October, 34-38.

Molenaar, Martien and Jose A. Martinez Casasnovas, 1996, A formalism for the structural description of vector maps, and its use for multi-scale representations: a hydrographic example, *Cartographica*, Vol. 33, No. 1, 55-63.

Monmonier, Mark S., 1982, *Computer-assisted Cartography, Principles and Prospects*, Prentice Hall, Englewood Cliffs, New Jersey.

Morehouse, Scott, 1995, GIS-based map compilation and generalization, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 21-30..

Muller, J. C., R. Weibel, J. P. Lagrange, and F. Salge, 1995, Generalization: state of the art and issues, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 3-17.

Muller, J. C., 1991, Generalization of spatial databases, in Maguire, David J., Michael F. Goodchild and David W. Rhind, (eds.) *Geographical Information Systems, Principles and Applications*, Vol. 1, New York: Longman Scientific & Technical, 457-475.

Nicholson, Norman L. and L. M. Sebert, 1981, *The Map of Canada*, Dawson, Kent, England and Archon, Connecticut.

Oosterom, Peter Van, 1995, The GAP-tree, an approach to 'on-the-fly' map generalization of an area partitioning, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 120-132.

Oosterom, Peter Van and Vincent Schenkelaars, 1995, The development of an interactive multi-scale GIS, *Int. J. Geographical Information Systems*, Vol. 9, No. 5, 489-507.

Oosterom, Peter Van, 1991, The Reactive-tree: a storage structure for a seamless, scaleless geographic database, *Proceedings of Auto-Carto 10*, 393-407.

Oosterom, Peter Van, 1989, A reactive data structure for geographic information systems, *Proceedings of Auto-Carto 9*, 665-674.

Peng, Wanning, 1997, *Automated Generalization in GIS* (Ph.D thesis), Wageningen Agricultural University and ITC, Netherlands.

Ratajski, Lech, 1967, Phenomenes des points de generalisation, *International Yearbook of Cartography*, 143-151.

Richardson, Dianne E., 1996, Automatic processes in database building and subsequent automatic abstractions, *Cartographica*, Vol. 33, No. 1, Spring, 41-54.

Richardson, Dianne E. and R. C. Thomson, 1996, Integrating thematic, geometric, and topologic information in the generalization of road networks, *Cartographica*, Vol. 33, No. 1, 75-83.

Richardson, Dianne E. and J. C. Muller, 1991, Rule selection for small-scale map generalization, in Battenfield, Barbara P. and Robert B. McMaster, (eds.) *Map Generalization: Making Rules for Knowledge Representation*, New York: Longman Scientific & Technical, 136-149.

Robinson, Gary J., 1995, A hierarchical top-down bottom-up approach to topographic map generalization, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 235-245.

Ruas, A. and J. P. Lagrange, 1995, Data and knowledge modeling for generalization, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 73-90.

Shea, K. Stuart and Robert B. McMaster, 1989, Cartographic generalization in a digital environment: when and how to generalize, *Auto-carto 9*, 56-67.

Smith, T. R., S. Menon, J. L. Star and J. E. Estes, 1987, Requirements and principles for the implementation and construction of large-scale geographic information systems, *Int. J. Geographical Information Systems*, 1, 13-31.

Snyder, John P., 1987, *Map Projection - A Working Manual*, U.S. Geological Survey Professional Paper 1935.

Spainhour, Stephen and Valerie Quercia, 1996, *WebMaster in a Nutshell*, O'Reilly, Sebastopol, CA.

Spiess, E., 1995, The need for generalization in a GIS environment, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 31-46.

St-Laurent, L., 1996, Review of geographic browsers on the Web, *Geomatics Canada: Land Information Network for Canada, LINC Version 1*, 1-29.

Stout, Rick, 1996, *The World Wide Web Complete Reference*, Osborne McGraw-Hill, CA.

Strand, Eric J., 1997 a, Java creates new channels for GIS information, *GIS World*, May, 28-29.

Strand, Eric J., 1997 b, GIS takes two roads to the Internet: ActiveX and Java, *GIS World*, January, 32-34.

Strand, Eric J., 1996 a, Vector maps weave their way across the Web, *GIS World*, December, 32-33.

Strand, Eric J., 1996 b, Spatial data plug into the Internet, *GIS World*, October, 30-32.

Su, Bo, Zhilin Li, Graham Lodwick and Jean-Claude Muller, 1997, Algebraic models for the aggregation of area features based upon morphological operators, *Int. J. Geographical Information Science*, Vol. 11, No. 3, 233-246.

Tang, Agatha Y., Teresa M. Adams and E. Lynn Usery, 1996, A spatial data model design for feature-based geographical information systems, *Int. J. Geographical Information Systems*, Vol. 10, No. 5, 643-659.

Taylor, Peter J. and Ronald J. Johnston, 1995, Geographical Information Systems and geography, in Pickles, John, (eds.) *Ground Truth*, Guildford Press, New York, 51-67.

Thomas, Brian J., 1997, *The Internet for Scientists and Engineers*, SPIE Optical Engineering Press, Bellingham, Washington.

Thomson, Robert C. and Dianne E. Richardson, 1996, Generalization, temporal GIS and map revision, *Geomatica*, Vol. 50, No. 4, 441-448

Weibel, Robert, 1995 a, Three essential building blocks for automated generalization, in Muller, J. C., J. P. Lagrange, and R. Weibel, (eds.) *GIS and Generalization: Methodology and Practice*, London: Taylor & Francis, 56-69.

Weibel, Robert, 1995 b, Map generalization in the context of digital systems, *Cartography and Geographical Information Systems*, Vol. 22, No. 4, 259-263.

Weibel, Robert, 1991, Amplified intelligence and rule-based systems, in Buttenfield, Barbara. P. and Robert B. McMaster, (eds.) *Map Generalization: Making Rules for Knowledge Representation*, New York: Longman Scientific & Technical, 172-186.

Worboys, Michael F., 1994, Object-oriented approaches to geo-referenced information, *Int. J. Geographical Information Systems*, Vol. 8, No. 4, 385-399.

Worboys, Michael F., Hilary M. Hearnshaw and David J. Maguire, 1990, Object-oriented data modeling for spatial database, *Int. J. Geographical Information Systems*, Vol. 4, No. 4, 369-383.

References on Internet:

Clement, Gilles, Christian Larouche, Denis Gouin and Paul Morin, 1997, Open Geospatial Datastore Interface (OGDI), <http://www.las.com/ogdi/ogdi/ong.htm>.

Environmental Systems Research Institute Inc. (ESRI), 1998, ArcView Internet Map Server, <http://www.esri.com/base/products/arcview/extension/imsext.html>

Environmental Systems Research Institute Inc. (ESRI), 1997, MapObjects Internet Map Server, <http://www.esri.com/base/products/mapobjects/ims/mapobjectsims.html>

LAS Inc., 1997 a, Overview, <http://www.las.com/ogdi/index.html>

LAS Inc., 1997 b, OGDI presentation, <http://www.las.com/ogdi/ogdialta.htm>

LAS Inc., 1996, New development in GRASSLAND: OGDI, <http://www.cecer.army.mil/grass/LAS/ogdi>

Netscape Communications Corporation, 1996, The Netscape ONE development environment vision and product roadmap,
<http://www.netscape.com/comprod/one/white-paper.html>

Object Liking and Embedding (OLE), <http://www.fit.unimas.my:8080/~nara/tmx/OLE/others/hand-out.html>

OLE-Object Linking and Embedding, <http://www.swan.ac.uk/ebms/docs/ole.htm>)

Sovik, Nathan, 1997, DDViewer3.0: interactive visualization of demographic data using Java technology, <http://sedac.ceisin.org/codata/ddviewer3.0/>

The Common Gateway Interface, <http://hoohoo.ncsa.uiuc.edu/cgi/primer.html>

University of California, Santa Barbara, 1997, Alexandria Digital Library,
<http://alexandria.sdc.ucsb.edu/>

APPENDIX A: THE C PROGRAM FOR CREATING BORDERLINE NODES

```
/* Name: gnt.c */
/* Utility: for creating borderline nodes */

#include <stdio.h>
#include <math.h>

main()
{
    extern FILE *fopen();
    FILE *fp1, *fp2, *fp3, *fp4;
    double sqrt(double);

    int i, j, k, l;
    char name1[10], name2[10], bname[10], dname[10];
    double dx, dy, dist;
    float x1,x2,y1,y2,A,B,s,minx,miny,maxx,maxy,ticx,ticy, r=10.0;
    float x0[20], y0[20], xc[20], yc[20], bnx[20], bny[20];

    /* reading the end points of the map boundary */
    printf("Enter the map boundary file: ");
    scanf ("%s", name1);
    fp1=fopen(name1,"r");
    if(fp1 == NULL)
    {
        fputs("Error in open\n",stderr);
        exit(1);
    }
    fscanf(fp1, "%f%f%f%f", &x1,&y1,&x2,&y2);

    /* reading the nodes which are closed to the map boundary. */
    printf("Enter the coverage node file: ");
    scanf ("%s", name2);
    fp2=fopen(name2,"r");
    if(fp2==NULL)
    {
        fputs("Error in open\n",stderr);
        exit(1);
    }

    i=0;
    do
    {
        fscanf(fp2,"%f%f", &x0[i], &y0[i]);
        i++;
    }
```

```

} while (x0[i-1]>0);

/* calculating the distance and create the borderline nodes */
k=0;
A=(x2-x1);
B=(y2-y1);
s=(A*A + B*B);

if (x1<x2)
{
    minx=x1;
    maxx=x2;
}
else
{
    minx=x2;
    maxx=x1;
}

if (y1<y2)
{
    miny=y1;
    maxy=y2;
}
else
{
    miny=y2;
    maxy=y1;
}

for(j=0; j<i; j++)
{

    xc[j] = (A*A*x0[j]+A*B*(y0[j]-y1)+B*B*x1)/s;
    yc[j] = (A*A*y1+A*B*(x0[j]-x1)+B*B*y0[j])/s;
    dx = (xc[j]-x0[j]);
    dy = (yc[j]-y0[j]);
    dist = sqrt(dx*dx+dy*dy);
    if(dist <= r)
    {
        if((xc[j]<=maxx)&&(xc[j]>=minx)
           ||(yc[j]<=maxy)&&(yc[j]>=miny))
        {
            bnx[k]=xc[j];
            bny[k]=yc[j];
            k++;
        }
    }
}
}

```

```

printf("Enter the output file of borderline nodes: ");
scanf("%s",bname);

fp3=fopen(bname,"w");
if(fp3==NULL)
{
    fputs("Error in open\n",stderr);
    exit(1);
}

printf("Enter the coordinates of the origin Tic\n");
printf("X: ");
scanf("%f",&ticx);
printf("Y: ");
scanf("%f",&ticy);

printf("Enter the output file of distance: ");
scanf("%s",dname);
fp4=fopen(dname,"w");

for(l=0; l<k; l++)
{
    fprintf(fp3,"%8d%20.2f%14.2f\n",l+1,bnx[l],bny[l]);
    dx=(bnx[l]-ticx);

    dy=(bny[l]-ticy);
    dist=sqrt(dx*dx+dy*dy);
    fprintf(fp4,"%8d%20.2f\n",l+1,dist);
}

fclose(fp1);
fclose(fp2);
fclose(fp3);
fclose(fp4);
}

```

APPENDIX B: THE INFO PROGRAM FOR IMPLEMENTING THE GNT SYSTEM

PROGRAM NAME: MATCHING

07/20/1998

10000 PROGRAM SECTION ONE

10001 REMARK **** THIS PROGRAM IS USED TO CREATE THE
MATCHING BETWEEN GNT FILES ****

10002 REMARK _____

10003 DISPLAY 'ENTER THE CURRENT GNT FILE: '

10004 ACCEPT \$CHR1

10005 DISPLAY 'ENTER THE CURRENT COVERAGE NAME: '

10006 ACCEPT \$CHR2

10007 DISPLAY 'ENTER THE NEIGHBOURING GNT FILE: '

10008 ACCEPT \$CHR3

10009 REMARK _____

10010 CAL \$NUM5 = 1

10011 LABEL A0

10012 IF \$NUM5 > 15

10013 GOTO A20

10014 ENDIF

10015 IF \$CHR1 = 'HYD1-GNT'

10016 SEL HYD1-GNT

10017 GOTO A1

10018 ELSE

10019 IF \$CHR1 = 'HYD2-GNT'

10020 SEL HYD2-GNT

10021 GOTO A1

10022 ENDIF

10023 ENDIF

10024 DISPLAY 'THE GNT FILE NAME IS WRONG!'

10025 GOTO A20

10026 LABEL A1

10027 RESEL NODE# = \$NUM5

10028 IF \$S50K = 1

10029 GOTO A10

10030 ENDIF

10031 CAL \$NUM6 = OBJ-ID

10032 CAL \$NUM7 = DISTANCE - 10

10033 CAL \$NUM8 = DISTANCE + 10

10034 CAL \$NUM9 = NODE-ID

10035 REMARK _____

10036 IF \$NUM9 > 1000 AND \$NUM9 < 3000

10037 CAL \$NUM10 = \$NUM9 + 2000

10038 EL SE

10039 IF \$NUM9 > 3000 AND \$NUM9 < 5000

10040 CAL \$NUM10 = \$NUM9 - 2000


```
10041 ENDIF
10042 ENDIF
10043 CAL $NUM20 = 0
10044 CAL $NUM21 = 5000
10045 MOVE 'FALSE' TO $CHR22

10046 REMARK _____
10047 LABEL A2
10048 IF $NUM20 > $NUM21 OR $CHR22 = 'TURE'
10049   GOTO A10
10050 ENDIF
10051 IF $CHR3 = 'HYD1-GNT'
10052   SEL HYD1-GNT
10053   GOTO A3
10054 ELSE
10055 IF $CHR3 = 'HYD2-GNT'
10056   SEL HYD2-GNT
10057   GOTO A3
10058 ENDIF
10059 ENDIF
10060 DISPLAY 'THE NEIGHBOUR GNT FILE NAME IS WRONG!'
10061 GOTO A20
10062 LABEL A3
10063 RESEL NODE-ID = $NUM10
10064 IF DISTANCE GE $NUM7 AND DISTANCE LE $NUM8
10065   MOVE 'TURE' TO $CHR22
10066 ELSE
10067 IF DISTANCE LT $NUM7
10068   CAL $NUM20 = $NUM10 + 1
10069   CAL $NUM10 = $NUM20
10070   GOTO A2
10071 ELSE
10072 IF DISTANCE GT $NUM8
10073   CAL $NUM21 = $NUM10 - 1
10074   CAL $NUM10 = $NUM21
10075   GOTO A2
10076 ENDIF
10077 ENDIF
10078 ENDIF

10079 IF $S0K = 0
10080 GOTO A10
10081 ENDIF
10082 IF $CHR2 = 'HYDRO1'
10083 SEL HYDRO1.AAT
10084 ELSE

10085 IF $CHR2 = 'HYDRO2'
10086 SEL HYDRO2.AAT
10087 ELSE
10088 IF $CHR2 NE 'HYDRO1' AND $CHR2 NE 'HYDRO2'
10089 DISPLAY 'THE COVERAGE NAME IS WRONG!'
10090 GOTO A20
10091 ENDIF
10092 ENDIF
```

10093 ENDIF
10094 RESEL OBJ-ID = \$NUM6
10095 CAL STATUS = 1

10096 REMARK _____
10097 LABEL A10
10098 CAL \$NUM11 = \$NUM5 + 1
10099 CAL \$NUM5 = \$NUM11
10100 GOTO A0

10101 REMARK _____
10102 LABEL A20
10103 END