Modelling Topographic Information in Sri Lanka within a Multiple Representation Database Management System

K.A.B.S.Rupasinghe

March, 2006
Modelling Topographic Information in Sri Lanka within a Multiple Representation Database Management System

by

K.A.B.S.Rupasinghe

Thesis submitted to the International Institute for Geo-information Science and Earth Observation in partial fulfilment of the requirements for the degree in Master of Science in Geoinformatics.

Degree Assessment Board

Thesis advisors
Dr. J. E. Stoter
Drs. R. A. Knippers

Thesis examiners
Prof. Dr. M-J. Kraak
Prof. Dr. Ir. P. van Oosterom

INTERNATIONAL INSTITUTE FOR GEO-INFORMATION SCIENCE AND EARTH OBSERVATION
ENSCHDE, THE NETHERLANDS
Disclaimer

This document describes work undertaken as part of a programme of study at the International Institute for Geo-information Science and Earth Observation (ITC). All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the institute.
Abstract

Modelling information in a multiple representation database (MRDB) supports consistent management of different representations of the same real world phenomena at different resolutions. This need highly arises in applications where users share the same database for different contexts and cartographic applications. Most of the National Mapping Agencies maintain topographic data sets at different scales separately. No inter database consistency and propagation of updates exists. Updates are done separately for data sets consuming money and time needlessly. The Survey Department of Sri Lanka also maintains data sets at different scales separately without inter database consistency and propagation of updates from large scale data set to small scale data sets.

The MRDB in this research is based on the model generalization which is a preprocessing stage for cartographic generalization to create subsequent cartographic outputs. The MRDB on model generalization establishes links between digital landscape models (DLMs) at different scales and mainly enhances the propagation of updates and data analysis capabilities between different representation levels.

Having the need to maintain the topographic data in a multiple representation database management system (MRDBMS), this research aims at implementing an MRDBMS for building features in a geodatabase, an object-relational database model extended with object-oriented capabilities. The research starts with analysing existing data specifications for topographic information in Sri Lanka and revising the specifications at the scales of 1:10,000 and 1:50,000 to represent these information in an MRDB environment. Then existing data are enriched according to the new specifications to support model generalization to create DLMs at above two scales in an MRDBMS. After enrichment process suitable generalization algorithms are chosen (some new algorithms created and some existing algorithms used). Further, requirements of the MRDBMS model are identified and MRDB model for buildings and road layers are designed. The MRDB model in this research adapts inter-relationship approach where representation at 1:10,000 and 1:50,000 are linked through inter-representation links (i.e. object links between data sets at different resolutions). Out of the two main approaches of creating an MRDBMS: a) creating MRDBMS during automatic generalization of large scale data set (base data set) to create small scale data sets by making inter representation links, and b) creating MRDBMS using already existing data sets at different resolutions through matching objects, the research adapts the first approach since the existing data sets at two scales are not consistent due to collection of data from different sources. In this approach, links between the data sets at two levels

---------------

i
are established during the generalization process itself.

The prototype implementation of MRDBMS in geodatabase starts with creating the logical structure and uploading data into the MRDB model. Then research defines the functionality for update propagation and queries in accordance with the requirements and specifications of MRDBMS. The concept used in this thesis for update propagation is based on the method called incremental generalization where only the features in a particular feature class influenced by updates are generalized using the same generalization rules adapted in the implementation of the MRDBMS. Finally the MRDBMS provides examples of some functionality for propagation of updates and querying between the two representation levels for the building feature class.

**Keywords**

Model generalization, cartographic generalization, multiple representation database, multiple representation database management system, data enrichment, clustering, data partitioning, automatic generalization, resolution, unified modelling language, propagation of updates, data consistency, data conflicts, object relational database, geoprocessing
# Contents

Abstract i  
List of Figures vii  
List of Tables ix  
Acknowledgements xi  

## 1 Introduction

1.1 Motivation and problem statement 1  
1.2 Problem scope 4  
1.3 Research identification 5  
1.3.1 Research objectives 5  
1.3.2 Research questions 7  
1.3.3 Innovation aimed at 8  
1.4 Project set-up 9  
1.4.1 Method adapted 9  
1.5 Structure of the thesis 10  

## 2 Generalization for Topographic Mapping

2.1 Overview 11  
2.2 Introduction 11  
2.3 Graphic and conceptual generalization 16  
2.4 Conceptual architecture for automatic generalization 16  
2.5 Knowledge acquisition for generalization 17  
2.6 Models of generalization 19  
2.6.1 Object generalization 19  
2.6.2 Model generalization 20  
2.6.3 Cartographic generalization 21  
2.7 Generalization operators 21  
2.8 Generalization algorithms 23  
2.9 Orchestration of generalization process 23  
2.10 Generalization quality assessment 23  
2.11 Concluding remarks 24
3 Multiple representation databases

3.1 Overview .................................................. 27
3.2 Introduction ................................................. 27
3.3 Data models .................................................. 30

3.3.1 Relational data model ..................................... 31
3.3.2 object-oriented data model ............................... 31
3.3.3 object-relational data model ................................ 31
3.3.4 Geodatabase model ........................................ 32

3.4 Unified modelling language .................................. 32

3.4.1 Objects ...................................................... 32
3.4.2 Classes ....................................................... 33
3.4.3 Association ................................................... 33
3.4.4 Generalization .............................................. 33
3.4.5 Class diagram .............................................. 34

3.5 Modelling MRDB ............................................ 34
3.6 Creation of MRDB ............................................ 35

3.6.1 Deriving links between objects in corresponding data sets 36
3.7 Multiple representation management system .................. 37
3.8 Automated propagation of updates in MRDB .................. 38
3.9 Concluding remarks ........................................... 41

4 Design of a multiple representation database .................. 43

4.1 Overview ...................................................... 43
4.2 Data source .................................................. 43

4.2.1 Topographic base data model TOPO-10K ................ 44
4.2.2 Topographic data model TOPO-50K ..................... 44
4.2.3 Existing specifications for 1:10K and 1:50K ............... 44
4.2.4 New feature specifications for 1:10K and 1:50K .......... 45

4.3 Data enrichment .............................................. 47
4.4 Design of multiple representation management system ..... 47
4.5 Data model for the MRDB .................................... 49

4.5.1 Object links between representations .................... 50
4.6 Concluding remarks .......................................... 50

5 Implementation of MRDBMS .................................. 51

5.1 Overview ...................................................... 51
5.2 Data and implementation environment ........................ 51
5.3 Data enrichment .............................................. 52

5.3.1 Data enrichment for building features ................... 52
5.3.2 Data enrichment for road and land use features .......... 59

5.4 Generalization process ....................................... 60

5.4.1 Specifications for generalization operations ............... 61
5.4.2 Generalization algorithms ................................ 63
5.4.3 Establishment of links .................................... 69
5.5 Evaluation of generalization ................................ 69
5.6 System architecture of MRDBMS .............................. 72
5.7 Implementing MRDB in ArcGIS 9 ............................. 73
## Contents

5.7.1 Designing the logical model ........................................... 74  
5.7.2 Generating the geodatabase schema .................................... 76  
5.8 Update propagation in the MRDBMS ...................................... 77  
  5.8.1 Specifications for change event ...................................... 78  
  5.8.2 Specifications for insertion of new objects .......................... 81  
  5.8.3 Specifications for deletion of objects ................................ 81  
5.9 Conflicts after updates ...................................................... 82  
5.10 Query functionality in MRDBMS ......................................... 83  
  5.10.1 Attribute queries ...................................................... 84  
  5.10.2 Spatial queries ....................................................... 85  
  5.10.3 Combination of attribute and spatial queries ...................... 85  
5.11 Concluding remarks ....................................................... 85  

6 Conclusions and recommendations ........................................ 87  
  6.1 Discussion ............................................................... 87  
  6.2 Conclusions ............................................................. 87  
  6.3 Recommendations ....................................................... 93  

Bibliography ................................................................. 95  

Appendices ................................................................. 100  

A Specifications for buildings .......................................... 101  
B Specifications for roads ................................................. 103  
C Specifications for land use .............................................. 105  

D Menus for data enrichment ........................................... 107  
  D.1 Adding unique Feature ID ............................................... 107  
  D.2 Classification features into category .................................. 107  
  D.3 Prioritize buildings .................................................... 108  
  D.4 Partition buildings into regions ........................................ 108  
  D.5 Cluster buildings ....................................................... 109  
  D.6 Sub-classification of buildings ........................................ 109  

E Menus for generalization ............................................. 111  
  E.1 Selection ............................................................... 111  
  E.2 Aggregation ............................................................. 111  
  E.3 Typification ............................................................. 112  
  E.4 Simplification ........................................................... 112  
  E.5 Establishing links ....................................................... 113  

F Generalization and MRDB Tools ...................................... 115  

G Terminology ............................................................ 117
# List of Figures

1.1 Characteristics of an MRDB ........................................... 2
1.2 Difference between model generalized database and cartographic database .......................................................... 3
1.3 Alternative geometric representations of the same cross road [45]. ............................................................... 4
1.4 Topographic map production flow ........................................ 5
2.1 Why generalize a map .......................................................... 12
2.2 Ratajski model of generalization [44] .................................... 14
2.3 Agent lifecycle .................................................................. 15
2.4 Relation between graphic and conceptual generalization [63] ................................................................. 16
2.5 Ladder and Star approaches in generalization .......................... 17
2.6 Generalization as a sequence of modelling operations [67] ............................................................... 20
2.7 Generalization operators [28] .................................................. 22
2.8 Generalization of an urban block: non uniqueness of results [26]. .................................................................. 24
3.1 UML static structure elements[25] ........................................ 33
3.2 Example of object classes ........................................................ 35
3.3 Links stored in the MRDB ....................................................... 36
3.4 The linking method of objects ................................................ 37
3.5 Example of complex update [24] ............................................ 40
4.1 Example of existing topographic data sets ........................................ 44
4.2 Classification hierarchy of building and road features ................ 46
4.3 General architecture of a federated database system ................. 48
4.4 General architecture for MRDBMS [52] ................................. 49
4.5 Conceptual data model for building and road features at two different scales. ...................................................... 50
5.1 Clustering buildings with creating buffers. ................................... 56
5.2 Clustered buildings in the source data set ................................. 57
5.3 Sub classification of clustered groups ..................................... 58
5.4 Work flow of the enrichment process .................................... 59
5.5 Example of possible aggregation of two close area objects ........ 60
5.6 The Generalization process ................................................... 61
5.7 Example of an aggregated building subject to eliminate from target data set .................................................... 62
5.8 Results of aggregation ....................................................... 65
## List of Figures

5.9 Analysis of results of aggregation ............................................. 65  
5.10 Analysis of results of aggregation ............................................. 65  
5.11 Typification method ................................................................. 66  
5.12 Results of typified buildings ..................................................... 67  
5.13 Simplifying buildings in ArcGIS .................................................. 68  
5.14 Example of simplified single building ......................................... 68  
5.15 Example of simplified aggregated buildings .................................... 68  
5.16 Results after generalization in different areas .................................. 70  
5.17 Comparison of existing data at two scales ...................................... 71  
5.18 System architecture of the MRDBMS .............................................. 72  
5.19 Creating a geodatabase schema using CASE tools (ESRI) ................... 73  
5.20 The FeatureClass logical model .................................................... 75  
5.21 Coded value domains ................................................................. 76  
5.22 Result of model generalization ..................................................... 77  
5.23 Example of update propagation event: attribute change .................... 79  
5.24 Example of update propagation event: delete ................................... 82  
5.25 Example of querying .................................................................. 84  

D.1 Menu in ArcMap to enrich Feature-ID .......................................... 107  
D.2 Menu in ArcMap to classify features .............................................. 107  
D.3 Menu in ArcMap to prioritize building features ............................... 108  
D.4 Menu in ArcMap to partition buildings into regions ........................... 108  
D.5 Menu in ArcMap to Cluster buildings ............................................ 109  
D.6 Menu in ArcMap to sub-classify buildings ...................................... 109  

E.1 Menu in ArcMap to select and eliminate buildings according to specifications. .......................................................... 111  
E.2 Menu in ArcMap to aggregate buildings ......................................... 111  
E.3 Menu in ArcMap to typify buildings ................................................. 112  
E.4 Menu in ArcMap to simplify buildings ............................................ 112  
E.5 Menu in ArcMap to establish link between building polygons of source data set and target data set .............................................. 113  

F.1 Generalization tools and MRDBMS tools in ArcMap .......................... 115
List of Tables

2.1 Sequence of operations for generalization .......................... 22
4.1 Specifications for building features at 1:10K and 1:50K data sets 45
4.2 Specifications for road features at 1:10K and 1:50K data sets . . 45
5.1 Attribute table of building layer of the source data set ........... 54
5.2 Example of object link tables ........................................... 69
5.3 Tag values used in the model. ............................................ 76
A.1 Specifications for building features at 1:10K and 1:50K data sets 101
B.1 Specifications for road features at 1:10K and 1:50K data sets . . 103
C.1 Specifications for land use features at 1:10K and 1:50K data sets 105
List of Tables

x
Acknowledgements

First and foremost, I want to express my deep and sincere gratitude to my supervisor Dr. Jantien E. Stoter, for her continuing suggestions and contributions throughout the thesis period. From the very beginning, even before my proposal writing, I got the opportunity to get her valuable and interesting comments and ideas on cartographic generalization which was a very much interesting and challenging field to me right from the beginning of my career in creating digital topographic databases in the Survey Department of Sri Lanka. During my research period, I appreciate the way she guided me to adapt a more scientific approach in the research work.

Many thanks should go to Drs. Richard A. Knippers, my second supervisor for his constructive criticisms, suggestions and valuable comments to the thesis. His support, guidance and encouragement were extremely useful for the successful completion of the research.

I am also indebted to Dr.Ir.R.A.de By, geo-information processing department for laying me a sound foundation on database management systems during the early part of my study. My deep gratitude should also go to Dr.C.P.J.M. van Elzakker, who paved me the way to develop my research skills during my study period.

This acknowledgement would not be complete without mentioning Mr. W.H.Bakker, geo-information processing department who laid down me the basic foundation on Python programming language which is a new object oriented programming language immensely useful in the field of geo-information processing.

I would also like to express my thanks to Mr. G.C.Hurneman, the programme director of geoinformatics (GFM) for his fatherly support during the whole period of study at ITC. My sincere thanks should also convey to Dr.V.A.Tolpekin, the student advisor of GFM-2 and the coordinator of the thesis modules for his valuable help not only in educational matters but also in personnel matters in my career of studies at ITC. Special thanks should also go to my friend, batch mate Miss. Zhou Yuhong for exchanging her valuable ideas and comments on my research. I am grateful to my GFM-2 colleagues who provided me a peaceful social and academic environment during my study period and stay in the Netherlands.

I remain forever indebted to the government of the Netherlands for the financial support given to me to read the degree and to the Survey Department of Sri Lanka which nominated me to follow this degree.

It is worthwhile to mention that this research would not have reached a successful completion without the support and encouragement of many individuals who have not been mentioned by names.
Finally, my deep gratitude and appreciation goes to my parents for their unwavering support, patience and love throughout my life. My deepest gratitude is reserved for my loving wife Ujitha, who stood with me giving all the support and encouragement for my work throughout the period of my study. To my family, I owe all this achievement.

Brian Rupasinghe
Chapter 1

Introduction

1.1 Motivation and problem statement

There exist a large number of various applications that utilize digital geographic information ranging from traditional applications in military and/or forestry to novel applications such as location based services. The new applications create new demand in geographical information [31]. Traditionally, small and medium scale geographical information was produced by National Mapping Organizations (NMOs). With the advancement of Geographic Information Systems (GIS) there has been an increase in the number of organizations that produce geographical information. This new trend has already been recognized in the European Commission initiative Inspire (Infrastructure for Spatial Information in Europe) [33]. Interoperability nowadays bridges the gap between different information systems providing a chance to find complementary information about the same or related facts in various sources which, have been independently developed by these organizations. However semantic interoperability is not easy to achieve, as related knowledge of information is most likely described in different semantic definitions, varying levels of detail, different application purposes and different data structures. Reconciling this heterogeneity to create a fully integrated database is known to be very hard and currently an unresolved problem. A simple way is to identify related knowledge and to provide a mechanism to materialize the relationships between different representations of the same phenomena. Hence support of multiple representations in database management systems (DBMS) provides a necessary step to interoperability [46] since it provides basis for harmonization of data sets of partner mapping agencies and other geoinformation providers.

An multiple representation database (MRDB) can be described as a spatial database, which allows us to represent same real world phenomenon at different levels of precision, accuracy and resolution [55] with different semantics and geometry. In an MRDB, different views on the same physical objects or phenomena can be stored and linked. Multiple representation is attributed from different views of the world, different applications, as well as different resolutions which lead to differences in the objects in terms of geometry and semantics. There are two main features that characterize an MRDB in a multi
1.1. Motivation and problem statement

Figure 1.1: Characteristics of an MRDB: Store multiple representations (left) link corresponding objects (right) [62].

scale context (see figure 1.1): (a) different levels of detail (LoDs) are stored in one database or several databases, and (b) the objects in the different levels are linked in MRDB. The first feature is similar to traditional map series of the NMOs: maps of different scales exist separately, only explicitly linked by the common location. In the second feature, individual objects are implicitly linked to each other in such a way that each object identifies its corresponding objects in other representations [59]. This is a specific feature of an MRDBMS.

There are several reasons for introducing an MRDB. It allows for a multiscale analysis of the data: information in one resolution can be analysed with information in another resolution: a topographic data set of lower resolution containing only settlement areas is queried concerning the buildings at higher resolution data set in that area; a situation where topographic data is linked with cadastre data [62]. A major reason for NMOs to investigate and implement MRDB is the inability to maintain a database where geometry information is kept at the most precise scale, and all geometries at less precise scales are automatically computed through cartographic generalization. Cartographic generalization is still a long, time consuming and costly process. Existing automated generalization software is still not capable of performing cartographic generalization fully automatically [54]; [46]; [57]. A major advantage of an MRDB, compared to maintaining the different scales separately is the possibility of propagating updates between the scales where the actual information is updated in the most detailed (high resolution) data set in order to propagate updates through links to other low resolution data sets [62] and to keep consistency through several databases. MRDBs can also be used to support where geographical information at various resolutions have to be distributed in real-time over the internet or mobile devices in order to support efficient zoom-in, zoom-out and browsing functionality [31].

Links between different levels of data sets in MRDB can be created in model generalization phase by creating new layers. In this process, it is possible to maintain links between objects in different data sets. In Model Generalization phase cartographic generalization is not considered which is used to enhance the graphical display. Model generalization can be seen as a pre-
processing stage in a geographic database for subsequent cartographic generalization, which uses reduction, enlargement, exaggeration, displacement and other modifications of the graphic symbolisms on a map to increase effectiveness of visual communication. The MRDB which is created in model generalization environment is based on data reduction as the scale decreases, is composed of real-world abstractions which do not have overlapping conflicts, since they have defined geographic extent and they do not have to compete for the space for their representations [20]. Figure 1.2 clearly distinguishes between model generalized database and cartographic database (at one level). However cartographic databases at different scales can also be represented in an MRDBMS if the data between different representations are consistent. Figure 1.3 shows three possible representations for the same real world road intersection shown on the photograph. The example is derived from the three databases that the French national mapping agency (IGN) currently maintains for its map production in order to give a better idea of multiple representation of same real world object. Representation 2 is the less detailed (sufficient for 1:50,000 to 1:100,000 cartography) and depicts the crossroad as a node, where roads are geometrically represented as lines. Representation 3 can support car navigation applications (where the scale varies between 1:10,000 in urban areas and 1:50,000 in countryside areas). It still represents the geometry of roads as lines but uses a finer granularity that more precisely defines the traffic paths in the crossroad as a complex system of lines. Finally, representation 1 is used for topographic maps at a more detailed scale (1:5000 to 1:10,000). It shows the geometry as a surface and requires high-resolution data acquisition. In this figure, the three geometric representations refer to the same real-world crossroad, irrespective of the fact that one representation conveys more detail
than the others. This approach can be used to maintain multi-representation semantics (i.e. the knowledge of what corresponds to what) whether the correspondence between instances materializing the same reality is a 1:1 mapping, a 1: n mapping or an n: m mapping. The left-hand side of figure 1.3 shows how the representations may be geometrically superimposed in an integrated visualization [45].

1.2 Problem scope

The Survey Department of Sri Lanka, being the sole NMO in Sri Lanka, maintains typically one database per scale independently from each other, at different scale levels: 1:250,000, 1:50,000 and 1:10,000 (still under production-50% complete) for the entire country and 1:5,000 or 1:2,000 in selected urban areas. These databases are in Arc Info coverage format produced and converted from CAD format in which the data is originally compiled. Although geographic objects in these databases have Geographic Feature Code (GFCODE) (which is a classification code), objects cannot be uniquely identified as in an object-oriented database. Another problem of the data sets is that data at the scales of 1: 2,000, 1:5,000 and 1: 10,000 have been collected using photogrammetric technique while some data at 1: 50, 000 have been generalized using manual methods from one inch sheets at the scale of 1: 63,360, produced by plane table survey technique in addition to photogrammetric technique and data at 1: 250,000 have been generalized from one inch sheets by using manual methods. Therefore there is no inter data consistency at the scales of 1:10,000, 1:50,000 and 1: 250,000, due to the reason that data has been collected and compiled using different sources. In addition to the data inconsistency itself, since these databases are maintained independently, there is no interrelationship among databases, hence with no update propagation form larger scale database to smaller scale databases and no inter-database consistency. Updates of these databases are done separately at different update cycles (5-7 years period) lead-
ing to data redundancy, inconsistencies, increase of data collection cost and waste of time. Figure 1.4 shows the existing topographic production flow in the Survey Department of Sri Lanka.

<table>
<thead>
<tr>
<th>DATA SETS</th>
<th>MAPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPO - 2K</td>
<td>MAP - 2K</td>
</tr>
<tr>
<td>TOPO - 5K</td>
<td>MAP - 5K</td>
</tr>
<tr>
<td>TOPO - 10K</td>
<td>MAP - 10K</td>
</tr>
<tr>
<td>TOPO - 50K</td>
<td>MAP - 50K</td>
</tr>
<tr>
<td>TOPO - 250K</td>
<td>MAP - 63360</td>
</tr>
</tbody>
</table>

K = 1000

- Digital data sets derived from Digital Photogrammetry
- Digital data sets derived from Digital Photogrammetry and digitizing 1:10K analogue maps
- Digital data sets derived from digitizing analogue maps
- Analogue maps compiled from Aerial Photogrammetry by analogue plotting
- Analogue maps compiled from Aerial Photogrammetry by analogue plotting and by manual generalization of one inch maps
- Analogue map produced by manual generalization of one inch maps
- One inch maps produced by Plane table techniques

**Figure 1.4: Topographic map production flow in the Survey Department of Sri Lanka.**

### 1.3 Research identification

#### 1.3.1 Research objectives

Based on the above discussion, the overall objective of this proposal is to enhance current functionality provided by data management software (DBMS or GIS) to support more flexible representation schemes, so that users may easily manage information using multiple representations. In this research proposal, building and road layers at the scales of 1:10,000 and 1:50,000 are only dealt with since they are subject to changes and hence updating more often
than natural features such as land use and hydrography. Since the data sets at the scales of 1:10,000 and 1:50,000 are inconsistent as mentioned earlier in chapter 1, creating a new level (1:50,000) for buildings and road layers from existing large scale data set (1:10,000) by automatic model generalization is the first step that needs to be performed [59]; [31]. Second step is to make object links between two data sets. Finally the third step is to define specifications and implement MRDB functionality for propagation of updates such as update attributes, delete, insert and modify objects and functionality for querying between representation levels. Multiple representations can also refer to as several representations of the same real world phenomena at the same scale that have been captured in different moments in time for different purposes. However, time dimension is not going to be considered in this research.

The specific objectives are to:

- analyse and evaluate the existing situation of topographic databases in the Survey Department of Sri Lanka,
- define new specifications for buildings and road classes for 1:10,000 and 1:50,000 data sets,
- enrich the existing data sets with auxiliary information to simplify generalization process,
- define generalization specifications to go from 1:10,000 level to 1:50,000 level,
- find out suitable model generalization algorithms to create 1:50,000 level through automated generalization of the large scale data set at the scale of 1:10,000,
- model an MRDB by designing a conceptual model, using object-oriented modelling language called Unified Modelling Language (UML) for buildings and roads feature classes at two representation levels: 1:10,000 and 1:50,000,
- specify requirements of the MRDBMS,
- implement generalization algorithms in the MRDBMS including the linkage of objects,
- implement MRDBMS converting conceptual model into a logical model,
- define specifications for MRDBMS in terms of functionality such as insertion, modification, deletion and querying, and
- implement update propagation and query functionality in the MRDBMS.
1.3.2 Research questions

The following research questions are addressed to fulfill the above specific objectives.

Specific objective 1: Analysing the existing situation in topographic databases in the Survey Department of Sri Lanka.

- How are data sets at different scales maintained, e.g. based on full automation of a basic core data set, in a multiple representation environment or independently from each other?

- What architecture and data format is used to produce and maintain data sets and what software is used to make products?

- Where is generalization applied in the production process? Interactively?

Specific objective 2: To define specifications for buildings and roads feature classes for both 1:10,000 and 1:50,000 data sets and for the MRDBMS.

- What are the existing specifications for building and road classes at both scales?

- What are the requirements for the respective topographic database system?

- How to define new specifications at both scales for building and road classes with hierarchical relationships of objects at these two scales in accordance with the requirements and for the MRDBMS?

Specific objective 3: To enrich the existing base data sets in order to facilitate model generalization.

- What auxiliary information (spatial and thematic) is used during data enrichment to support the generalization process?

- What are the existing functionality available in the software that can be used to enrich the existing base data sets at the scale of 1:10,000 for two feature classes?

- How data enrichment is performed interactively or automatically?

Specific objective 4: To find out suitable model generalization algorithms to create 1: 50,000 level through automated generalization of the large scale data set at the scale of 1:10,000.

- What are the appropriate generalization operations and algorithms for this research?

- How generalization assessment is done by comparing the generalized data with original data?
1.3. Research identification

Specific objective 5: Modeling of an MRDB by designing a conceptual model, using object-oriented programming language called Unified Modelling Language (UML) for buildings and roads feature classes for two representation levels: 1 : 10,000 and 1 : 50,000.

- What are the modeling concepts available to create and manage multiple representations of geographical data?
- How can these concepts be applied most optimally to the Sri Lankan case?
- How geographical information at different levels can be related to each other?

Specific objective 6: To implement Multiple Database Management System (MRDBMS).

- Which database platform and software is used to create prototype system for MRDBMS?
- How object links are implemented at different levels?
- How conceptual model is converted into logical model?
- How to implement feature classes and tables for defined levels of representation?
- How can update propagation be implemented with insert/update/delete functions?
- How to implement other functionality in an MRDBMS such as querying different representations?

1.3.3 Innovation aimed at

With an MRDBMS, several potentials can be distinguished in general: a) automatic propagation of updates from the large scale data sets to other low resolution data sets can be done efficiently, b) harmonization of data sets produced by different mapping methodologies, c) preservation of data consistency among related multiple representations, and d) query functionality for information drilling: users have the ability to query in a certain area or a certain object in the map or even in certain attributes which are not stored explicitly with a certain feature but feature linked to this in another representation level.

Potentials of an MRDBMS, specific for Sri Lanka: a) application of MRDB concepts into current production process to generate small scale Digital Landscape Models (DLMs), b) to propagate updates from base DLM to other small scale DLMs, and c) to generate Digital Cartographic Models (DCMs) from DLMs to create visually enhanced end products for users at different scale levels efficiently.
### 1.4 Project set-up

#### 1.4.1 Method adapted

- A review and analysis of the existing situation of topographic databases in the Survey department by examining data sets and sending a questionnaire to the department.

- Literature study to review:
  - fundamental concepts of object orientation, object relational database models and MRDB principles,
  - elements of the UML relevant to the modeling phase of MRDB in this research,
  - spatial data models, modeling spatial relationships and consistency constraints in spatial databases,
  - needed functionalities that will be used to enrich the existing base data set,
  - data enrichment possibilities and procedures, and
  - existing generalization algorithms for building and road layers.

- Define and formalize specifications for the two feature classes by studying existing specifications.

- Enrich the existing base data set by introducing unique identifiers and essential attributes required to obtain generalization knowledge.

- Building conceptual models using UML for two feature classes and for the MRDBMS.

- creating and testing the generalization algorithms in order to select suitable one for the created model.

- Creating and implementing object links between two levels during the generalization process itself.

- Evaluation of generalization quality

- Specifying functionalities for MRDBMS.

- Implementing MRDBMS

- Implementing and testing functionality for update propagation such as insert, update and delete data and other functionality such as database querying.

- Testing and validation of the MRDBMS.
1.5 Structure of the thesis

Following this introductory chapter, thesis is organized into six chapters and they are described as follows:

Chapter-2 describes ‘State of the Art’ of generalization and review of generalization research. Foci are set on conceptual framework of generalization, model and cartographic generalization, generalization operators and generalization algorithms, orchestration of generalization process and quality assessment of generalization.

Chapter-3 starts with review of MRDBMS research. Then it describes concepts in data modeling and Unified Modelling Language as a standard modelling language, modelling approaches for MRDBMS followed by different techniques to make links between different representation levels, requirements of an MRDBMS and finally the concept of automatic propagation of updates in an MRDBMS.

Chapter-4 starts with topographic data and databases maintained in the Survey Department of Sri Lanka together with existing specifications and new specifications for MRDBMS. Then chapter describes about data enrichment and its requirements, design of MRDBMS illustrating general system architecture of an MRDBMS. Finally chapter presents a conceptual data model for MRDB to model building and road features for topographic mapping.

Chapter-5 starts with a description about the data and implementation environment for MRDBMS in this work. Then it describes data enrichment procedures and generalization process with results at each stage together with evaluation of generalization for building features. Further it describes the system architecture and implementation of MRDBMS in geodatabase in ArcGIS software platform. Finally chapter describes the concept of update propagation used in this work, specifications for update propagation and querying functionality with implementation results at each stage.

Chapter-6 is the conclusions and recommendations starting with a brief discussion about the work together with an outlook for further improvements and possibilities.
Chapter 2

Generalization for Topographic Mapping

2.1 Overview

This chapter first introduces generalization with a review of generalization research (section 2.2). Then chapter describes the concept of graphic and conceptual generalization (section 2.3) and conceptual architecture for automatic generalization (section 2.4). Section 2.5 of the chapter describes knowledge acquisition for generalization including data enrichment purposes as preprocessing stage for generalization. Then chapter further describes models of generalization (section 2.6), generalization operators used on the type of generalization i.e. model vs cartographic generalization (section 2.7) and generalization algorithms (section 2.8). Finally the chapter describes orchestration of generalization process to see how generalization processes are sequenced and tuned according to purpose (section 2.9) and generalization quality assessment which is important during evaluation of generalization outcome (section 2.10).

2.2 Introduction

Geographic information is presented on a printed map in traditional cartography. The smaller the map scale the more the representation is simplified and abstracted. Therefore the choice of map scale has important implications on how generalization is performed. The purpose of generalization is to produce a good map, balancing the requirements of accuracy, information content and legibility as the scale decreases, the space that is available to portray a certain area on a map sheet gets smaller. In other words, generalization encompasses the modifications of the information in such a way that it can be represented on a smaller surface, retaining the geometrical and descriptive characteristics. The essence of original should be maintained at smaller scales (see figure 2.1). However, the map scale is not the only factor to determine generalization process. The purpose of the map also influences selection of appropriate map scale thereby indirectly affecting generalization process [27]. Different attempts have been made to keep this generalization process as general as possi-
ble to achieve stringent definition. The International Cartographic Association has defined the process of generalization as 'the selection and simplified representation of detail appropriate to scale and/or the purpose of a map' [2].

Manual cartographic generalization is holistic in nature because several decisions are applied at once to portray important aspects of an area like which objects are important to preserve, which relationship between objects are important and which objects can be displaced and omit unwanted detail. This holistic nature of generalization is difficult to describe formally and it is a combination of both scientific and artistic elements [31].

Attempts to automate generalization have a history that reaches back to the advent of computers in cartography. So far these attempts have only been partly successful. There are several reasons why generalization process is difficult to fully automate. One reason is as mentioned the holistic and artistic nature of generalization. Generalization is described as subjective, interactive, idiosyncratic and comprehensive in its perception and execution. Another reason is the difficulty to define what the good generalization is because results of generalization of same area of a map by different experienced cartographers are not always the same. McMaster [3] notes that: "Even the most skilled manual cartographers would have trouble precisely replacing their results from one day to the next". Yet another reason why cartographic generalization is diffi-
cult to automate is complexity of different object classes in a topographic map and requirements of generalization are different for different object classes [31]. Maintaining the topological relationships of different feature classes in generalization process is another difficulty in automation. For instance, after simplification of a road, building on one side must not lie on the other side.

Automate cartographic generalization was first tried out to find solutions for restricted spatial problems, in particular for linear features by Douglas and Peucker [50] and point and area features by Töpfer and Pillewizer [51]. Moreover a number of authors proposed several conceptual aspects and models to better understand the generalization process.

A first conceptual framework by Ratajski identified two fundamental type of generalization processes: a) quantitative generalization, which consists of a gradual reduction in map content depending on scale change, and b) qualitative generalization, which results from the transformation of elementary forms of symbolization to more abstract forms. An important concept in Ratajski’s framework is the generalization point at which the map capacity is reduced to the level where a change in cartographic method is necessary (e.g. see fig.2.2, A to B). This changing capacity of a map can be represented by a triangle, with the base of which illustrates the maximum capacity and top of which illustrates the limit of map capacity (minimum capacity). When the map capacity is near the top of the triangle, a new cartographic method must be applied in order to initiate a new generalization cycle. For instance, as illustrated in figure 2.2 (B), single homes must be replaced with built-up areas for the settlements which is considered as a point or line method (a qualitative measure). With continued generalization, built-up areas are converted into settlements according to magnitude (a quantitative measure). These conversions are illustrated clearly in figure 2.2 with the difference of qualitative and quantitative methods [44].

After Ratajski’s model, several frameworks for automated generalization have been proposed over the past two decades. Among these, there are some most influencing models. Steiniger and Weibel [69] distinguish these frameworks into two models; process oriented model which structures the entire generalization process and object level oriented models which address the level of map objects. Examples for few influential process oriented models are the frameworks of Brassel and Weibel [58] and McMaster and Shea [68]. In the model of Brassel and Weibel, five separate processes of generalization are included: (a) structure recognition, (b) process recognition, (c) process modelling, (d) process execution, and (e) data display. Model proposed by McMaster and Shea [68] is one of the early initiatives towards automated cartographic generalization in which, entire generalization process is divided into three critical components: a) a consideration of philosophical objectives (why to generalize), (b) cartometric evaluation to maintain clarity with appropriate content of features (when to generalize), and (c) spatial and attribute transformations, i.e. performance of actions of generalization (how to generalize).
Attempts towards automatic generalization at the early stage involved a GIS or digital mapping software application applying geometric algorithms, one at a time, to simplify, displace, exaggerate, aggregate, collapse or otherwise modify an individual map feature. The limitation of this approach is that the algorithms operate in isolation. A major step forward was the introduction of object-oriented data models and associated object-oriented spatial toolkits. This enabled algorithms to operate in the context of the feature, so that it could use neighbourhood information to modify the effect of the algorithm. In object level oriented modelling, two major generalization processes can be identified: a) rule based modelling in which whole generalization process is divided into conditions-actions. b) constraint based modelling in which iterative refinement strategy with trial and error (i.e backtracking facilities in generalization process) is adapted [69]. Weaknesses in the first method include: acquisition and formalization of cartographic rules, necessity of great amount of rules and sequencing of generalization operations since different operators may affect each other. Second model has advantages over the first because constraints formulate requirements of a generalized map; in other words, the conditions generalized map should adhere to. However, in contrast to rules, violation of such a condition is not bound to an action and choosing an action to solve a problem is the result of synthesis of conditions.

The object level oriented framework by Ruas and Plazanet [42] realizes a constraint based modeling approach. The proposed framework consists of three levels of processing. On the highest level a 'global master plan' determines the sequence of generalization tasks to apply on the level of entire map (e.g. aggregate all buildings of same class). On the second level, geographic situation is selected according to the given task (e.g. buildings in an urban block). On the third level (local) a generalization plan for every situation is developed and
executed and the generalization results evaluated. In this local level, generalizations operations, parameters and their sequence are determined. If the evaluation phase produces bad solution, three actions are possible: a) selection of other parameters, b) another algorithm may be used, and c) a new local plan is chosen. This process is then continued until a satisfactory solution is obtained. This model has been refined and re-implemented in the Automated Generalization New Technology (AGENT) project [69].

The AGENT includes agents which are geographical objects for generalization. Cartographic constraints represent the goal of agents which are recursively formed from map objects: micro-agents represent single cartographic objects (e.g. a building or a road), meso-agents represent groups of objects (e.g. city blocks or road networks) and macro-agents represent feature classes. Each agent makes use of measures or methods to detect spatial conflicts and evaluate the satisfaction of its goals as well as generalization algorithms to resolve spatial conflicts. The application of the measures and generalization algorithms is controlled by the constraints that apply to a particular type of agent (e.g. a building should be large enough, wide enough, should not change its position too much, etc.). These constraints can be of different types: graphic, topological, structural or gestalt. [73].

When constraints are unsatisfied, agent proposes a new generalization plan to remedy a constraint violation by changing the goals for new specifications for generalization by sorting the available plans, then agent triggers the first plan and re-evaluates its own happiness. Depending on the success of the plan, (see figure 2.3) it may backtrack and trigger the next best plan. When a perfect state is reached, plan is stopped or a new cycle starts if the happiness has improved [43].

![Agent Lifecycle Diagram](image-url)

Figure 2.3: Agent life cycle of automated generalization (43).
2.3 Graphic and conceptual generalization

Two types of generalization concepts can be distinguished: graphic and conceptual generalization (see figure 2.4) in applying generalization operators during generalization. Graphic generalization is characterized by simplification, enlargement, displacement, merging and selection. None of these operations affects the symbology. Whereas conceptual generalization is also characterized by merging and selection in addition to symbolization and enhancement operators which indeed affect the symbology of the map. In other words, graphic generalization mostly deals with geometry of the spatial objects while conceptual generalization deals with attributes of the same.

As the scale decreases, the less details can be represented and more important the generalization becomes, i.e. more conceptual generalization activities are needed if the scale reduction is significant (e.g. from 1:5000 to 1:25000). If the scale change is considerably small (e.g. from 1:2000 to 1:5000) more graphic generalization is needed. Graphic generalization is similar to cartographic generalization and conceptual generalization is similar or equal to model generalization in which more reduction of database contents are required as the scale decreases. Thus selection of generalization operations depending on the target scale and purpose is knowledge based.

2.4 Conceptual architecture for automatic generalization

There are two approaches to perform automatic generalization: a) ladder approach, and b) star approach Eurogeographics (2005) [41] (see figure 2.5). According to Stoter [41], presently some NMAs adapt both approaches and some-
times a mixed approach including both. In the ladder approach, small scale data sets are derived from large scale data set in steps (from scale to scale). In the star approach, each and every small scale data set is generalized from the base data set. In the mixed approach, large to middle data sets are derived from the base data set while smaller scales are derived from one middle-scale data set. Deciding which approach is chosen is a crucial factor in model generalization. If ladder approach is used, derivatives at different scales is difficult to maintain at sufficient accuracy as the spatial accuracy gets lower in databases up in the ladder. The the main idea of database generalization is to reduce data under quantitative-statistical control [15]. If the star approach is used, all databases are derived from source database and hence spatial accuracy of databases would be much better than that of ladder approach. Mixed approach is also a good solution because the databases at smaller scales are not required to have higher accuracy, compared to intermediate scales. For this work, these two approaches become the same since only data sets at two scales are involved and dealt with.

2.5 Knowledge acquisition for generalization

One reason given by Weibel and Dutton [67] for the difficulty in formalising generalization knowledge is as they claim "Cartographic knowledge is different from other knowledge types (e.g. the knowledge needed in medical diagnosis) in that it is essentially graphical and therefore hard to verbalise and formalise”.

There are several ways to classify generalization knowledge. According to Armstrong [4], generalization knowledge can have three different forms for a ruled based system:
2.5. Knowledge acquisition for generalization

- geometrical knowledge: refers to geometry of object, shape and the pattern formed by number of objects,

- structural knowledge: refers to cartographer's expertise in for example, geomorphology, cultural geography or hydrology, and

- procedural knowledge: refers to knowledge about which generalization operators and algorithms to use and how parameters to be set for a particular algorithm in a particular situation.

Ormsby and Mackaness [49] classifies knowledge needed to automate cartographic generalization in object-oriented paradigm considering: the geometry of the object, semantic properties of object and relationship among objects.

According to Kilpelinen [13], generalization knowledge is classified into four types: geometric, topological, context-related and culture-related. Geometric rules contain thresholds such as minimum size of buildings to be selected and minimum length of a road to be selected. Topological rules relate to express relations of different features. For example, a house must remain on the same side of the road before and after generalization. Context-related rules refer to rule of thumb for terrain properties. For example, aggregation of buildings should not be performed if there is a canal between them. Culture-related rules refer to historic or cultural characteristics such as age of a building.

An understanding of the knowledge expressed as constraints rather than rules is very much important in the generalization process. Rules define actions that should be taken when certain criteria are fulfilled. A constraint is such a criterion that should be met to achieve an acceptable solution. [31]. A typical example for a constraint is: size of a building should be greater than 100m². All constraints must be satisfied or resolved, but any number of actions can be applied to resolve them [28]. Classification of constraints in the context of generalization according to Weibel and Dutton [58] consists of five categories: a) graphical (specify the minimum size and width and are directed by graphical limits), b) topological (ensure that the relationship between features are maintained), c) structural (define criteria describing spatial and semantic structure), d) gestalt (relate to aesthetics and visual balance), and e) process (mainly influence how operators are selected and sequenced) [27].

Neun et al. [66] have discussed better methods for complex pattern and structure recognition and integration of structural and procedural information into generalization procedures specially in cartographic and multi-scale databases. According to their view, data enrichment is necessary to include raw spatial data with additional information (auxiliary data) about objects and their relationships for variety of purposes within the overall generalization process. These purposes are:

- characterization: refers to maintain important map objects, patterns, and relationships suppressing unimportant ones. Hence spatial and semantic
characteristics of objects have to be identified in order to obtain priority orders among objects; meaningful group of objects e.g. clusters of buildings, objects aligned in a particular arrangement based on adjacency, proximity relations and hierarchical relations,

- conflict detection: can be achieved by comparing the results of characterization steps with thresholds imposed by cartographic principles (e.g. minimal size or distance threshold for the target scale), and

- algorithm and parameter selection: appropriate algorithms have to be selected depending on the character of the data to be generalized. This auxiliary data relating to character of data helps not only to select most appropriate generalization algorithms but also to set appropriate values to parameters which control these algorithms.

Even though the current data enrichment methods mainly focus on horizontal relationships which means dealing with same level of details representing common structural properties (e.g. neighbourhood, pattern, alignments), they can also be used to represent vertical relationships between homologous objects or object groups. These links can be based on attributes especially on geometric relations. MRDBs have vertical relationships (i.e. relationships between map objects across scales). Vertical relationships often represent hierarchical relations which exist between the objects and their composite object on a higher level of detail.

### 2.6 Models of generalization

A more wider meaning has been given to generalization in digital cartographic systems and GIS. It can be considered as a process of representing real world between different models with abstract details while preserving maximum information required depending on the purpose of the application. Generalization takes influence through building the first model as DLM (Digital Landscape Model) also known as object generalization (see figure 2.6). Then as a part of the derivation of special purpose, reduction of the first model leads to model generalization. Finally model generalization is responsible for cartographic visualization of primary or secondary models depending on the scale and map purpose (cartographic generalization) creating a Digital Cartographic Model (DCM) [67].

#### 2.6.1 Object generalization

Object generalization process takes place at the time of defining and building the original database. In this phase, at first structure of the original database is considered to achieve a complete data set of real world as much as possible. Generalization occurs here in form of abstraction, selection and reduction of information from real world objects depending on the purpose of the database [27].
2.6. Models of generalization

In contrary to cartographic generalization, model generalization operates completely at the data model level. It does not need to concern itself how to make the best use of limited space on a map. Instead its main role is to reduce amount of data to the level suitable for the target scale and removal of detail that is not required for the target scale. According to Weibel [5] model generalization should cause fewer generalization effects in which reduction of data volume is maximized while at the same time modification of source data is minimized. Therefore model generalization mainly affects the symbology of the map due to selection, removal and reclassification of objects and not the geometry of features. In general, model generalization consists of the following operations: a) selection by class, b) individual selection of objects by attribute value and/or context, c) geometry-type changes (collapsing), and d) typification (optional). But according to the classification of peng [14] and Neuffer et al. [43], geometric simplification is also a generalization operation which affects geometry of features in model generalization phase. Further, Neuffer et al. [43] have argued on the basis that geometric simplification does not displace objects where displacement is treated in cartographic generalization phase. Also the same idea is born by Weibel (Weibel, 1986) discussed in [15]. Validity of this argument is rather vague since simplification itself displaces the vertices of features. But my argument is that geometric simplification can still be used in model generalization phase if it will not create conflicts among features. In other words, only if it is subjected to minimum average displacement [58]. It means that depending on the type of features to be generalized, the generalization procedures, algorithms and intended use of data, selection of generalization operations may be changed. This leads to a situation where it is difficult to identify a clear-cut boundary between model generalization and cartographic generalization. For example, simplification operator can still be used on isolated single buildings which do not create conflicts during model generalization. Where as simplifi-
cation operator might not be used in generalizing road network during model generalization since it might create conflicts among features due to significant displacement of vertices of roads.

2.6.3 Cartographic generalization

Cartographic generalization can be considered as the second process immediately followed by model generalization. Unlike model generalization, cartographic generalization deals with how to make best use of map space to optimize legibility at a given scale and for a particular purpose of the map. It includes following operations: a) applying map-specific representations, b) displacement of features to legibility issues, and (c) label placement [43]. Cartographic generalization mainly differs from model generalization in the representation of objects through graphical symbolization.

2.7 Generalization operators

Decomposition of overall generalization process into sub processes is achieved using generalization operators. Traditionally cartographers have used different operations such as selection, simplification, combination and displacement to describe the various stages of generalization process. In the digital context, a functional breakdown of generalization process is very much important as it clarifies identification of constituents of generalization and enrich development of specific solutions to sub problems in generalization process [67].

The main task of generalization operator is to solve a specific generalization problem. Combination of generalization operators can then be used to build an entire generalization process or workflow. There are different criteria to determine which operator must be applied to a given situation. Also generalization operators do not behave equally. The use of generalization operators is influenced by three main elements: a) the feature class (e.g. road, building), b) preceding situation analysis, and c) the map scale. Figure 2.7 provides a textual and graphical overview of the generalization operators used in generalizing various feature classes according to the classification of Galanda [28]. Particularly generalization operators can be divided into two groups [27]:

- **independent**: this kind of operator is applied to individual objects or groups of objects independent of their spatial context (no spatial relationships with other features; e.g. simplification and smoothing), and

- **contextual**: context dependent operators like selection, aggregation, displacement or typification can only be triggered and controlled by spatial analysis of the context. For instance, relation between surrounding objects is needed to keep spatial integrity when objects are displaced.

Table 2.1 shows the relation and sequence of generalization operations used in model generalization phase in an operation matrix, according to Peng [14]. According to the figure the first operation in model generalization phase is the
2.7. Generalization operators

selection and ordering of generalization operations can be done in both directions (row wise and column wise). Selection operation always appears in the diagonal as seen in the figure. Character 'X' describes that there is no connection of one operation to the other.

Table 2.1: Operation matrix - Legend: *: the first operation; X: no connection; 1, 2, 3,...: priority order. Adapted from Peng (14)

<table>
<thead>
<tr>
<th>*</th>
<th>Selection</th>
<th>Reclassification</th>
<th>Aggregation</th>
<th>Typification</th>
<th>Deletion</th>
<th>Simplification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reclassification</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Aggregation</td>
<td>x</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Typification</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Deletion</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Simplification</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>Smoothing</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
2.8 Generalization algorithms

While a generalization operator defines the transformation in the generalization process, a generalization algorithm is used to implement the particular transformation. This implies that in contrast to generalization algorithms, operators are independent of a particular data model (e.g. vector or raster) [27]. Many generalization algorithms implement one or several operators. For example, there are algorithms that perform line generalization as a combination of the operators selection and exaggeration; bends that should be maintained are selected and bends that are too small are exaggerated [74]. There are countless algorithms developed over the past three decades for both vector and raster data. Many algorithms have been developed for vector data using context-independent operators for points and line features such as selection, simplification and smoothing [67].

2.9 Orchestration of generalization process

Application of generalization operators which are universally valid for every map scale and purpose in a fixed sequence or order is not simple or sometimes not even possible. Thus depending on the purpose and the situation or specific problem, generalization operators and algorithms must be re-arranged and tuned. Also if the user needs to constantly interacts with the system to choose algorithms and parameters to generalize, orchestration of the generalization process is hindered [31]. Therefore development of optimised human-computer interaction mechanisms is required in accordance with specific requirements of generalization during several phases of the development and use of generalization systems: a) during algorithm development and testing, b) for selection and fine tuning of generalization algorithms, and b) parameters and for the evaluation of results [67].

2.10 Generalization quality assessment

Traditionally, cartographic generalization is manually performed and visually assessed. Also evaluation of automated generalization to a large extent relies on visual assessment and comparison with manually produced maps [67]. The generalization methods that are used strongly influence the resulting generalization effects. Manual generalization causes different generalization effects from what is given by automated algorithms. In automated generalization when new algorithms are produced, results of their effects can be compared with existing algorithms in order to have a relative assessments of the results [15].

With the advancement of technology, database storage of digital data has enabled systematic control of generalization results by computer processing. Nevertheless evaluation of generalization with the help of this automation is not easy to convert into simple procedures. For example, a rule for deleting fea-
2.11 Concluding remarks

This chapter explored generalization from manual methods to automated methods by introducing the review of generalization research. Since fully automatic generalization is not possible nowadays, model generalization can be seen as a good alternative (a pre-processing stage) for subsequent cartographic generalization. Model generalization is the process of reducing details of the master data set; a kind of data filtering process to create small scale data sets preserving the geographic extent of features. The main difference between model generalization and cartographic generalization is that cartographic generalization is more concerned about the visual quality of the final map considering features in the small scale would be to select and eliminate all tertiary roads from the road network. But a tertiary road leading to a group of buildings will not be eliminated due to its importance because of building features. In this kind of a situation, it is really difficult to assess the quality of generalization because it is hard to say that keeping the tertiary road in the database is either an error or not. This implies that generalization rules exist but are adapted to the context of generalization and therefore, evaluation of generalization has become more and more difficult (see figure 2.8).

The quality of results of generalization mainly depends on two aspects: a) whether the generalized data are related to user requirements, and b) whether generalized data preserves the reality. Bard [26] has developed an assessment model for generalization without manual evaluation. This model is based on: a) data characterization which means description of features by particular geographic properties such as area, length or width before and after generalization, b) evaluation which means data quality assessment from the difference between the observed result and the theoretical result, and finally 3) the aggregation of the various assessment results to summarize data quality with help of an assessment algorithm.

Figure 2.8: Generalization of an urban block: non uniqueness of results [26].
visual enhancement generalization operators such as enhancement, exaggeration and displacement while model generalization deals more with database content. This research is dealt with model generalization to create a small scale data set from master data set to represent topographic information in a multiple representation database environment. Next chapter is devoted to MRDB review and modelling and creation of MRDBs.
2.11. Concluding remarks
Chapter 3

Multiple representation databases

3.1 Overview

This chapter starts with a research review of multiple representation databases (MRDB) (section 3.2). Then it describes in general the existing database models and Unified Modelling Language (UML) as database modelling language (sections 3.3 and 3.4). Next the chapter elaborates modeling of MRDB with two modelling approaches (section 3.5). Further chapter describes creation of MRDB with the methods and techniques of establishing links between different representation levels (section 3.6). Section 3.7 describes the specification requirement for a multiple representation management system. Finally the chapter deals with the concept of propagation of updates at different levels with different approaches adapted in the literature for update propagation (section 3.8).

3.2 Introduction

While the real world is assumed to be unique, the way it is represented in a database depends on the intended use. This means that same geographic phenomenon can be represented in several geographic data sets according to purpose. For instance, the Survey Department of Sri Lanka maintains its base data for topographic mapping, for all road features only with center lines while Road Authority maintains the same roads as polygon features in order to have more details which would be more helpful them to make decisions in designing new roads, road widening and laying down road reservations. Thus different applications that use same geographic phenomena may have different perceptions and therefore required different representations; i.e. different sets of objects, links, properties and cartographic resolutions [47].

Multiple representation may also be caused by different approaches in data collection and different semantic definitions. For example, it may be accidental when two unrelated databases represent the same entity (e.g. road) with differ-
3.2. Introduction

ent geometry at same scale due to different semantic definitions [52]. In cartographic point of view, map makers need to prepare maps for the same area with different representation level of details. The process to derive smaller scale maps from large scale detailed maps, known as cartographic generalization is still not fully automatic and is an interactive process as is described in section 2.2. Hence many National Mapping Agencies (NMAs) are tempted to maintain different data sets or data bases at different resolution levels with no interaction, inter database consistency and update propagation among each other. A multiple representation geographic information system enables storing all representation in a single database and allows consistency through different levels with appropriate object links which eventually helps update propagation at different levels from high resolution data sets to low resolution data sets.

Multiple representation concept well fits in the federated database (FDB) approach, where several distributed, autonomous and heterogenous databases are integrated into a federation layer. Distribution means that the data is stored in different database management systems (DBMS) at different locations. Heterogeneity because databases can use different DBMS and autonomy because the different database systems (DBS) can be designed independently from each other and it is possible to run DBS independently from federation layer. This layer contains a working database to store all needed meta data such as links between autonomous databases and registration of these databases together with rules for update propagation of MRDB [30].

The origin of the Multiple Representation Database (MRDB) concept started with the research initiatives of the National Centre for Geographic Information Analysis (NCGIA), 1988-1990 [6]. Since then various aspects of MRDBs have been discussed and studied by several authors.

Kilpelinen [9] addressed the issue of how automatic incremental generalization process would be applied in updating an MRDB. In this process, she suggested to cut generalization problem into modules where only the modules influenced according to the definitions in conflict sets had to be processed. Kilpelinen [13] in her PhD research on ‘Multiple Representation and Generalization of Geo-Databases for Topographic Maps’ has focused on the maintenance and generalization of geodatabases with topographic data. Her work is based on a conceptual model for generalization in databases for model generalization and cartographic generalization within the concept of multiple representations with a model for an MRDB. However she has only given a proposal for generalization operators in model generalization which is required to implement MRDB environment. Also development of methods for update propagation has not been considered but a concept called incremental generalization has been introduced to tackle updating geographic databases.

A case study has been carried out to implement the connectivities between the multiple representation levels using object-oriented approach for the MRDB model created by Kilpelinen [13] in Laser Scan Gothic object-oriented database.
with Gothic API (Application Programming Interface) and LAMPS2 software with Laser Scan user language called 'Lull'. Results of this study have showed that the object-oriented approach is suitable for implementing an MRDB. However complete implementation of a prototype system to test the operationality of multiple representations has not been covered in this case study [20].

Harrie and Hellström [60] have developed a prototype system for the propagation of updates for cartographic data sets at different scales based on object-oriented techniques with Lamps2 as platform. Their prototype is based on a conceptual framework of four steps: examination, propagation, generalization of updates and solution of spatial conflicts. Rules formulated in these steps are triggered when an object is edited in the master data set. Finally their prototype has been tested with a map, produced manually by cartographers. The results of the test have revealed that the prototype system did not perfectly match with manual generalization. The results of manual generalization have been better due to the fact that cartographers could make judgements as what to generalize and not unlike computer assisted generalization which work on defined rules.

Twumasi [25] in his M.Sc research on 'Modeling spatial object behaviours in object relational geodatabase' has created a multiple representation environment (road polygons and road centerlines) for road layer in the Netherlands at the same scale. He has not considered derivation of one representation at lower resolution from the other representation at higher resolution. In other words, automatic generalization with maintaining links to obtain a smaller scale data sets has not been considered.

MurMur (Multi Representations - Multi Resolutions) is a research project completed in the year 2002 on multiple representations. It has taken into account two different perspectives: a) multi-scale databases, where representations at various resolution levels are stored in a single database, linked through cartographic generalization functions, and b) temporal databases, to support multiple representations (at the same scale) due to time varying information. The project started with an existing spatio-temporal conceptual model and extended it with multi-representation capabilities such as ability to users to define the content of the database, to automatically generate the specifications of the database according to target implementation platform, to define queries and to visualize the query results taking into account the different representations [23].

Cecconi [27] in his PhD research 'Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping' has discussed implementing an MRDB to bypass missing operations and complex algorithms used in on-the-fly generalization in Web mapping for feature classes buildings, roads and rivers. The results of this integration have shown that the efficiency of generalization capabilities to generate web maps at different scales by subdividing generalization process into two phases; an off-line processing phase
and on-line real-time phase has increased. This is because the results of more complex generalization algorithms are stored as levels of details (LODs) in the MRDBMS and the realtime generalization during second phase is completed based on these LODs.

Hampe et al. [59] in their paper on MRDB 'Applications for Data Revision and Real-Time Generalization' have discussed advantages of MRDB for storing and serving spatial data on web. They have mentioned in detail how an MRDB is created with links at different levels of details during the implementation of generalization algorithms.

Hampe and Sester [62] in their paper on 'Generating and Using a Multiple Representation Database (MRDB) for Mobile Applications' have focused on the development and implementation of generalization algorithms to produce several layers of an MRDB by providing links between different levels and development of applications that exploit the MRDB structure for the use of mobile maps in small devices. They have not considered development of functionalities of the MRDB in their study. However their future work would concentrate on the development of functionalities of the MRDB. Also they need to investigate to determine which generalization procedures can be processed in realtime and which need support from the MRDB.

Dunkars [31] in his PhD research 'Multiple representation databases for topographic information' has covered creation of MRDB using existing data sets provided by the National Land Survey of Sweden at three different resolutions 1:10,000, 1:50,000 and 1:100,000. Also he has not done any data enrichment and generalization processes on these data sets. These data sets are pre-generalized data, created through cartographic generalization. In his study links between different data sets have been established for roads and buildings using automatic data matching procedures.

### 3.3 Data models

A conceptual data model (or schema) is a representation of data structures that are required by the database system. The data structures include the data objects, the associations between data objects, and rules which define operations on the objects. Also classification of database systems depends on the use of data model. A data model focuses more on what data are required and how they are organized or structured rather than what operations will be performed on the data. Nowadays there exist three popular data models, namely, relational, objected-oriented and object-relational. As a whole, a data model can be described as:

- a conceptual design of the abstract of real world depending on the user-defined view on relevant data applications,
- a formal and descriptive method of describing real-world entities, their attributes and relationships, and
• independent from computer systems but dependent on database management systems (relational, object oriented, object relational).

3.3.1 Relational data model

Relational databases are based on the relational model, which organizes and stores data as collections of tables (called relations) that are logically associated to each other by shared attributes. The individual records are stored as rows in the tables while the attributes are stored as columns. Each column can contain attribute data of only one type, numeric, string, date and so forth. These types are called the domain of an attribute [29]. The advantages of this data model are: a) simple table structures, b) intuitive, simple user interface, c) attribute tables can be linked to tables describing the topology necessary for a GIS, and d) direct access to data provides fast and efficient performance. However there are several disadvantages of this data model including: a) limited representation of the real world, b) limited flexibility of queries and data management, c) complex data relationships are difficult to model, and d) slow sequential access.

3.3.2 Object-oriented data model

Object-oriented databases were introduced as a means of overcoming the limitations of relational model and to handle more complex application requirements, which the relational model cannot capture. In an object oriented database model, real world entities are abstracted and held as objects. All objects belong to object classes. For each class there may be many objects, but each object belongs to only one class. The class defines what values can be held by an object. Values can be simple data types (integers, strings, dates, etc.) together with more specialist types (geometries, locations, rasters, and tables). Furthermore, objects can hold structural information or references between objects. In object orientation, behaviours are the methods or operations that an object can perform (e.g. an account may know how to subtract money from its balance when a withdrawal is made). Behaviours can also be used to send messages to other objects to inform the state of an object with its current values, to store new values or to perform calculations [29]. There are number of advantages to this model: a) allow complex representation of the real world, b) multiple simultaneous updating, c) maintains history in the database, d) ensures a high level of data integrity, and e) encapsulation combines attributes and operations making an object accessible through well defined methods.

3.3.3 Object-relational data model

Object relational model is the most recent development in the domain of logical data models. These systems build object oriented capabilities on top of a robust relational database. Object-relational systems are extended by software that incorporates object-oriented behaviours, but data are not encapsulated unlike in object-oriented systems. Database information are stored in tables, but some of the attribute columns can include a richer data structure called an abstract data type. The object-relational model has advantage of speed and the ability to
handle complexity as well as the database building integrity of object-oriented design. It has the additional advantage of supporting Structured Query Language (SQL) and can access typical RDBMS (relational databases) [29].

3.3.4 Geodatabase model

The geodatabase data model is an object-relational data model for geographic data extended with object oriented behaviour. Basically, a geodatabase is a storage mechanism for spatial and attribute data that contains specific storage structures for features, collections of features, attributes, relationships between attributes and relationships between features. All geodatabases can store tables, feature classes, feature data sets, and functionality such as relationship rules, attribute domains, and geometric network connectivity rules available in object-oriented systems. Such functionality are not available in object-relational systems [29]. In geodatabases, the features become smarter by endowing them with natural behaviors, and to allow any sort of relationship to be defined among features [18]. The geodatabase data model brings a physical data model closer to its logical data model. The data objects in a geodatabase are mostly the same objects which are defined in a logical data model, such as owners, buildings, parcels and roads. Further, the geodatabase data model implements the majority of custom behaviors without writing any code. Most behaviors are implemented through domains, validation rules, and other functions of the framework. Writing software code is only necessary for the more specialized behaviors of features such as implementing consistency constraints and update propagation events.

3.4 Unified modelling language

The Unified Modeling Language (UML) is a standard modeling language for visualizing, specifying, constructing and documenting the artefacts of software systems [53]. The UML is called a modelling language, not a method. Most methods consist, at least in principle, of a modeling language and a process. The UML comprises a number of graphical elements that are integrated to create models in order to have multiple views of the system. UML only describes what a system is supposed to do and does not show how to implement a system [25].

3.4.1 Objects

An object is shown as a rectangle with two compartments in UML notation. The top compartment shows the name of the object and its class, all underlined. The second compartment shows the attributes for the object and their values as a list (see figure 3.1).
3.4.2 Classes

Classes are represented in the UML by a solid-outline rectangle with three compartments separated by horizontal lines. The top compartment holds the class name and other general properties of the class (e.g. stereotypes); the middle compartment holds a list of attributes; the bottom compartment holds a list of operations. Abstract classes have their names in italics (see figure 3.1).

3.4.3 Association

Association is represented in the UML as a line connecting two classes with the association name just above the line. This kind of association is called binary association in the UML. Association role is shown at both ends of the line next to the class. Association cardinality or multiplicity is shown just above the association line near the appropriate class (see figure 3.1). Each role of an association has a multiplicity value that indicates how many objects of the given class may be linked to an object of the other class. Multiplicity is a piece of information carried by the role, and it corresponds to a bounded integer expression: $1$: one and only one, $0..1$: Zero or one, $M..N$: From $M$ to $n$ (Natural Integers), $*$: From zero to any positive integer, $0..*$: From zero to any positive integer, $1..*$: From one to any positive integer.

3.4.4 Generalization

Generalization in UML notation is a relationship between a superclass and its subclasses that share properties and methods with the super class. In UML, generalization is represented by a line that connects the subclass to the super-
3.5 Modelling MRDB

Features in a database are abstractions of the real world and these abstractions must be meaningful in accordance with the purpose. When creating a model for an MRDB, relation between objects (geometry and semantics) in different data sets that represents the same real world phenomenon should be analysed. One issue that has to be analysed carefully is to what extent different database objects represent the same real world entity. Fundamental for the representation of these entities is the scale which influences the level of detail of an object being represented. For instance, one-real world building entity is represented as a symbol in one data set and as a polygon in another data set depending on the scale [31]. An MRDB can be modeled using Unified Modeling Language (UML) that has been explained in the previous section.

The object classes used for spatial data models often form hierarchies [14], [7] & [11]. One example of a hierarchy is: deciduous forest, forest and vegetation. In UML syntax, this kind of hierarchical relationship can be modeled as a generalization (see figure 3.2). This is also called as ‘IS-A’ kind of relationship. In the UML relations, associations can also be used to represent relationship between object classes. Associations do not have a hierarchical nature [31]. Molenaar [11] has described how spatial data could be generalized in a database system, i.e. his focus was on model generalization. He suggested that the generalization hierarchies such as the one shown in figure 3.2 have a relation to the resolution of the geographical data set. Also he explained how reclassification hierarchies express reclassification of objects as the scale decreases. According to his view, after reclassification neighbouring objects that are of the same type can be aggregated. For instance, the forest objects at the intermediate scale can be formed by aggregating neighbouring deciduous, mixed and coniferous forest objects from the large scale data set.
There are two approaches to design an MRDB. The first approach is the integrated approach also called as object approach [13] and the second approach is the inter-relationship approach. In the first approach, it is possible to allow one object to have several geometrical representations for a geographic phenomenon [54]. There are no dependencies (no correspondences between group of objects) among these representations because it integrates objects in a class hierarchy [52]. But both object and value correspondence can be maintained between a representation class and its integrated class. In the second approach, also called level-based approach, each object has one geometry and connects between different representations (dependencies between pairs of source and target classes are maintained) of the same real world phenomena. In this thesis inter-relationship approach is adapted because dependencies between pairs of source and target classes are sufficient in the field of data warehousing, geographic generalization and propagation of updates between objects affecting geometrical changes. Creating relationships between data sets at different levels in this approach will be discussed in detail in sub-section 3.6.1.

UML is used for modeling the MRDB in this thesis for number of pragmatic reasons. UML is a standard language for object oriented modeling and is platform independent. It can also be extended according to requirements.

### 3.6 Creation of MRDB

When the model for MRDB is established, next step is to populate data to the model. Spatial data has to be imported into the different levels (levels - 1:10K and 1:50K in this research). There are two different ways to populate data to MRDB model for implementation [62]:

- The different levels are created through automated generalization of large scale data set, and
The levels are populated with separate data sets and then link of objects between different data sets are established through matching operations.

In the first approach, new data sets are created from existing base data by applying generalization operations. In the second approach, matching operations have to be developed to create links. In this research, first approach is used since data sets at two levels i.e. 1:10K and 1:50K are not consistent to do matching operations.

### 3.6.1 Deriving links between objects in corresponding data sets

In the research of Kilpeläinen [13] & [20], the MRDB consists of a core data set with higher resolution and lower resolution data sets were created automatically through model generalization of the core data set. In her approach, links between different levels were bidirectional. Hampe et al., [59] have discussed three different approaches in creating links in an MRDB: a) attribute variant, b) variant bottom-up, and c) variant top-down. In the 'attribute-variant' approach, the whole MRDB is stored in one data set. It uses additional attributes to describe the scale on which, a particular object will appear and the geometric changes of objects while changing the level of details. The 'variant bottom-up' approach keeps track of the object IDs of the target data set in source data set. In the 'variant top-down' approach, objects are linked in the opposite direction. If this approach is used, it would be better to use an extra table for storing links (see figure 3.4). Because this variant in one to many relationship (like a built-up area contains many buildings) would create a higher number of empty cells in the table. Hampe and Sester in their work [62] have adapted "variant bottom-up" approach where they store links from large scale data sets to the small scale data set. This link is one way i.e. object IDs of the features of small scale are kept track in the source data set with an additional attribute; Link _ 50K (see figure 3.3).

![Figure 3.3: Links stored in the MRDB (left); tables show the structure of database tables (right) (62).](image-url)
In accordance with the table '1:10K' in figure 3.3, it stores links to the tables '1:25K', '1:50K' and '1:100K' in the attribute fields 'Link_25K', 'Link_50K' and 'Link_100K'. For example, building 1012 in the base data set is represented by building 2513 in 1:25K as a simplified building; as an aggregated building in 1:50K and as part of the built_up area in 1:100K level. These links have been established in an object relational database with the object ID's of features at different levels. Although these links are only stored in one direction, requests are possible at both directions. For example, built_up area 9858 can be retrieved where building 1012 is located in and in turn from 'Link_100K', buildings inside the built_up area 9858; 1012 and 1013 can be retrieved [62].

If bi-directional connectivities are maintained as explained by Kilpeläinen, it introduces large number of columns to store possible \( n \) connections (in 1 : N case) at small scale data set creating many null values in the table, leading to inefficiency in making query analysis between different levels. Creating one-way links as mentioned is a very efficient approach in this respect. However, in the case of representing \( m \) to \( n \) relationships, a separate table is required with the primary keys of the tables at both levels represented as foreign keys in the separate table.

### 3.7 Multiple representation management system

The main purpose of the multiple representation database management system (MRDBMS) is to actively maintain multiply represented entities with respect to a set of consistency rules. Therefore MRDBMS should contain a multiple representation schema (MR schema) which defines a number of consistency, matching and restoration rules. The MRDBMS should have the following requirements for the management of multiple representation [52]. The MRDBMS must:

- be configurable according to the rules defined in the MR schema,
3.8 Automated propagation of updates in MRDB

Once the links of objects between different data sets are established, next step is how to propagate updates to the small scale levels when the source data set is updated. Kilpelinen and Sarjakoski [9] & [70] suggested an approach called incremental generalization for propagation of updates in geodatabases. Their principle of making incremental generalization work is dividing the generalization problem into modules by ordering the generalization steps and identifying the conflict sets. In this manner, it was not necessary to generalize globally and only the objects that were subjected to changes could be updated. Kilpelinen [13] in her thesis has further mentioned that in an incremental generalization environment, the generalization process is performed for the entire geodatabase only once and following update transactions to the geodatabase, the old generalized output is also updated in an incremental manner concentrating only on modules influenced by updates.

The prototype system for propagation of updates, developed by Harrie and Hellström [60] has been tested on an MRDB where links between objects have been created manually although automatic methods are possible. This system is based on a conceptual framework which consists of four steps: a) examination, b) propagation, c) generalization of updates, and d) solution of spatial conflicts. Examination checks the current status in the target data set and determines how the update affects the target data set. Propagation executes the outcome of the examination step and validates updates and integrity of the MRDBMS. Generalization of updates step creates new or modified object in the target data set with selected generalization operations. In solution of spatial conflict phase, (e.g. a new road may overlap with existing buildings) displacement of objects is considered with the constraint method. For example, for a new road in the target data set, conflict set for all objects that are affected by new road is created. Then analytical constraints are set up based on the geometry and displacement

- be able to match the representation objects (r-objects) of representation databases according to the matching rules stated in MR schema,
- provide a repository over the multiply represented entities in order to keep track of the entities that are consistent and those that are not,
- evaluates the consistency rules with respect to a multiply represented entity,
- monitor the representation databases for changes to the r-objects that may affect the consistency of multiply represented entities,
- send requests for consistency restoration actions when an inconsistency is detected. These restoration rules should be given in the MR schema, and
- be able to perform spatial and attribute queries between different representations to visualize the results.
behaviour types of the objects in the object set. These constraint are expressed as linear equations of displacement of points and finally the solution of these equation system provides required point displacements to solve the spatial conflicts.

The update propagation in the prototype developed by Harrie and Hellström [60] has dealt with functionality; creation, modification and deletion of objects. In accordance with their system, not more than one object would be created in the target data set during update propagation. The results of the prototype is dependent of the order the objects are inserted in the large scale data set. The steps in the system have been implemented with reflexes (i.e. specific rules for propagation process). For instance, one rule would be to represent a foot path having length greater than 500m in the target data set. These propagation rules have been defined because they created the prototype with already generalized cartographic data sets and rules were not available at hand unlike the dealing with propagation of updates on MRDBMS created by making links during generalization process itself.

Skogan [24] has proposed a rule-based incremental generalization method to realize update propagation based on incremental generalization proposed by Kilpelin and Sarjakoski [70]. Skogan’s system is rule-based where a generalization schema contains generalization rules expressed in a generalization schema language. For example, rule R1 is an aggregation rule that generates built-up areas based on building clusters. Rule R2 creates single buildings and rule R3 simplifies roads. The batch generalization process reads the schema, decides the order of the generalization rules and starts processing rules one by one. According to Skogan, the three classical database change actions are insert, modify and delete. Insert actions involves creating an object and inserting into the source data set. The modify action involves changing an existing object by modifying one of its attributes and delete action removes an existing object from the database. Although these actions seem to be quite simple, consequence of each action is quite complex. Figure 3.5 shows state of source data set S and target data set T just after batch generalization at time 0. At time 1, few changes have occurred; insertion of building marked S12, close to dead-end road S1 triggers rule R2. Also this new situation should trigger rule R3 to include the road previously not generalized. Insertion of two new buildings marked S14 and S15 triggers rule R1, which results in a new built-up area t7. The t1, t2 and t3 buildings still exist in the target database and should be deleted to make the target database consistent. Therefore for each change of object state, it is necessary to search for close objects influenced by respective object change. This set of close objects is called influence set. For example, influence set for new buildings S14 and S15 are found by finding the close source objects that contributes to the built-up area t7 generated by rule R1.

The difference of incremental generalization process with batch generalization is that firstly, incremental generalization should be started with a single changed object as origin and should compute the object’s influence set to
re-evaluate appropriate generalization rules. Secondly, it should find existing appropriate target objects and update them instead of only creating new objects.

Anders and Bobrich [30] describe the concept of MRDB as a directed graph which provides relationships between objects at different levels. These relationships are used to propagate updates from large scale to small scale data sets. According to their concept, three type of update events are used: a) insert - a new object is created, b) remove - an object is deleted, and c) change - an object is changed. Object changes are further divided into change attribute, change geometry and change attribute and geometry. A set of generalization operators are selected by a rule based system for the update process. Their concept is still under development.

Haunert and Sester [1] have developed a method for propagation of updates for aggregation of adjacent polygon features. Their approach is based on incremental generalization. The basic idea in this method is that updates which are inserted into a source data set can be propagated to a generalized target data set without repeating the generalization of entire map. In this approach, they identify the influenced region created by the updates and generalization algorithm for aggregation described by Oosterom [71] is applied to this region. Their method has been successful for the case of aggregation with area partitioning by this algorithm. However, updates that produce gaps or overlapping areas are not allowed. This implies that inserting or deleting a single feature or changing the geometry of a single feature is not feasible with the exception of an attribute change of an area. Interesting point is whether this method can also be applied or extended on other generalization cases apart from aggregation of adjacent areas.
3.9 Concluding remarks

This chapter described the concept of multiple representation databases and how they are modelled with an introduction to existing database models and UML. Geodatabase model and UML are used to model and represent MRDB in this research. Further chapter described two modelling approaches to create an MRDBMS with several possible methods of establishing links between different levels of representations and their advantages and disadvantages. Then chapter described the general specifications for a multiple representation management system with several rules. Finally the chapter described the concept of propagation of updates in MRDB with an important update propagation concept called incremental generalization. Next chapter deals with the data specifications, data enrichment methods, specifications for generalization and MRDBMS and design of MRDB model for road and building classes with their relationships and cardinality constraints at two representation levels (1:10K & 1:50K) in this research.
3.9. Concluding remarks
Chapter 4

Design of a multiple representation database

4.1 Overview

This chapter starts with a description of existing data for road and building features and existing topographic data sets at 1:10K and 1:50K maintained in the Survey Department of Sri Lanka including existing and new data specifications (section 4.2). Further it describes about the data base enrichment and its requirements in this work (section 4.3). After data enrichment, the chapter describes the design of MRDBMS by illustrating its general architecture (section 4.4). Finally the chapter provides the conceptual data model used for roads and building features in this work (section 4.5).

4.2 Data source

The foundation of the system is a multiple representation database (MRDB) with predefined levels of details (LODs). For this research, two topographic data sets; TOPO-10K & TOPO-50K of the the Survey Department of Sri Lanka (National Mapping Agency) are considered (see figure 4.1). TOPO-10K represents the National Topographic Maps at the scale of 1:10,000 and TOPO-50K represents the same maps at the scale of 1:50,000. The small scale data set is generated from large scale base data set for this work through automatic generalization since the existing small scale data set is not consistent with the base data set (see section 1.2) to represent these two data sets in an MRDB. Each data set consists of different thematic layers maintained in ArcInfo coverage format. Each layer includes geo-referenced objects (points, lines & polygons), attributes and topology (relationship to neighbouring objects). For this work only two layers; Building and Trans (road layer) is dealt with at modelling phase. However, enrichment of land use data is also considered at data enrichment phase to assist building data enrichment (see sub section 5.3.1).

- **Building**: including all types of buildings;
- **Trans**: including all roads and foot paths.
4.2. Data source

4.2.1 Topographic base data model TOPO-10K

TOPO-10K is the digital base data model of Sri Lanka which has been created by digitizing the National Topographic Maps at the scale of 1:10,000. The model consists of eleven thematic layers which include: transportation network including railways, building features, hydrographic network, terrain relief, administration boundaries, land use features, utility network, geographic place names, reserves, geodetic control network and grid. In this data set, building features are represented as polygon features while road features are represented by line features with centre line of roads.

4.2.2 Topographic data model TOPO-50K

TOPO-50K is a digital data model of Sri Lanka which has been created by digitizing the National Topographic Map at the scale of 1:50,000. The model consists of all thematic layers included in 1:10K except utility network. Building features in this model are represented as point features while road features are represented by line features with centre line of roads.

4.2.3 Existing specifications for 1:10K and 1:50K

The existing data specifications for all feature layers have been based on the topographic map series at 1:10K and 1:50K. There are no unique identifiers for objects. All layers have common attribute fields such as geographic feature code (GFCODE), geographic feature name (NAME), year of data collection (YEAR) and method of data compilation (METHOD). The GFCODE represents the type of feature and its geometry. For example, 'High Courts'; a polygon building feature is specified as 'HCTRA' where first four characters are the abbreviation for feature type while the last character represents geometry type of feature (i.e. point, line or area). Method of data compilation defines how data have been collected. For example, method 1 is for data digitized from 1:10K maps and method 2 for data digitized from 1:50K maps. A description of existing specifications can be found in the appendices A, B and C for building, road...

Figure 4.1: Example of existing topographic data sets: data layers of buildings, roads and land use features at 1:10,000 scale (left), data layers of same features at 1:50,000 scale (right). Note: figures are not to scale.
and land use features. Detailed description of all the specifications are found in the Topographic Database Specifications and Data Dictionary of the Survey Department of Sri Lanka.

### 4.2.4 New feature specifications for 1:10K and 1:50K

The existing data specifications for road and building layers need to be changed in this work to new specifications to accommodate multiple representation environment in database generalization. For example, for both layers, there is no field to represent unique object identifier for each feature in the existing specifications. Therefore objects cannot be identified uniquely. The existing specifications have been made by investigating the available representation of topographic map series at both scales. For instance, building features in 1:10K have been represented as polygon features while in 1:50K they are point features. In this scale, for example, the same 'High Courts' building is specified as 'HCTRP'. In the case of road network, classification (GFCODE) of roads at both scales are similar (see appendix B).

<table>
<thead>
<tr>
<th>Building Feature Class</th>
<th>Feature ID (FID_10K / 50K)</th>
<th>Geographic Feature Code (GFCODE)</th>
<th>Category</th>
<th>Name</th>
<th>Ownership</th>
<th>Year of Data collection (Map_Year)</th>
<th>Method of data compilation (Method)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hospital - General</td>
<td>1</td>
<td>GHSPA</td>
<td>Health</td>
<td>Colombo</td>
<td>State</td>
<td>1999</td>
<td>1</td>
</tr>
<tr>
<td>Hospital - base</td>
<td>2</td>
<td>BHSPA</td>
<td>Health</td>
<td>Galle</td>
<td>State</td>
<td>2000</td>
<td>2</td>
</tr>
<tr>
<td>House</td>
<td>4</td>
<td>HOUSA</td>
<td>Residential</td>
<td>St. Amss</td>
<td>Private</td>
<td>2000</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Road Feature Class</th>
<th>Feature ID (FID_10K / 50K)</th>
<th>Geographic Feature Code (GFCODE)</th>
<th>Category</th>
<th>Name</th>
<th>Rd Level (Level)</th>
<th>Rd Type (Type)</th>
<th>Year of Data collection (Map_Year)</th>
<th>Method of data compilation (Method)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expressway</td>
<td>1</td>
<td>EXPRL</td>
<td>Motor road</td>
<td>Kandy road</td>
<td>Above ground level</td>
<td>A1</td>
<td>2002</td>
<td>1</td>
</tr>
<tr>
<td>Main Road</td>
<td>2</td>
<td>MNRLDL</td>
<td>Motor road</td>
<td>Galle road</td>
<td>Ground level</td>
<td>A2</td>
<td>1999</td>
<td>1</td>
</tr>
<tr>
<td>Foot path</td>
<td>3</td>
<td>SDRDL</td>
<td>Foot path</td>
<td>-</td>
<td>Ground level</td>
<td>H2</td>
<td>1999</td>
<td>2</td>
</tr>
</tbody>
</table>
4.2. Data source

The new specifications for the objects of the two feature classes are based on the classification hierarchy described in section 3.5. The representation geometry of buildings at 1:50K is changed to polygon geometry from existing point features in the new database design. Figure 4.2 illustrates two levels of classification hierarchy for building and road objects used for this work. GFCODE is classified into a new attribute field called 'CATEGORY' based on the usage of buildings and roads (e.g. legal, commercial for buildings) according to this concept. Therefore this generalization hierarchy enables aggregation of objects that are of the same type during database generalization (e.g. aggregation of buildings). In addition to Feature ID and classification code, new attribute fields 'OWNERSHIP' and 'RD_LEVEL' for building and road features respectively are introduced. New specifications of few elements for buildings and roads layers with all attribute fields are listed as examples in tables 4.1 & 4.2. Fields with dark grey colour in the tables are the fields used in 1:50K level while light grey attribute fields belong to 1:10K. Whereas attribute fields not highlighted in tables (attributes for Feature ID and road level) are used in both levels. A description of revised new specifications for important attribute fields for these two layers and for land use can be found in the appendices A, B and C.

![Diagram of classification hierarchy of building and road features](image)

Figure 4.2: Classification hierarchy of building and road features, the arrows indicate the IS-A relationships.
Chapter 4. Design of a multiple representation database

4.3 Data enrichment

The process of data enrichment also known as data enhancement can be described as adding auxiliary information to an existing base data set to make it more enriched in order to support generalization process (see section 2.5). Time consuming complex generalization algorithms such as partitioning features into regions, clustering features into similar groups can be processed beforehand during data enrichment process and their attribute values or results can be stored in the database to be used at subsequent generalization process. However, adding information of whatever type must be well considered so as not to congest the database with unnecessary information. There are three main reasons for data enrichment for this work is to: a) identify homogeneous regions for area partitioning b) simplify more complex time consuming generalization algorithms by storing the necessary attribute values with the use of preprocessing algorithms (e.g. clustering algorithm); these values can then be used as parameters for subsequent generalization process to increase its efficiency and, c) make links between two data sets using the same attribute class in both data sets. This means during generalization process, values of important enriched attributes of features in the source data set such as group number can be used to make the links between two data sets. This method can be considered as vertical data enrichment between data sets at different resolutions in an MRDB as mentioned in section 2.5.

4.4 Design of multiple representation management system

The design of MRDBMS can be seen as a federated database system (FDBS) [52], [59] & [30]. FDBS as explained in chapter 3.2, acts as a federation layer for several independent component databases which can be run independently from federation layer. Figure 4.3 shows the general architecture of a federated database system [30]. In an MRDBMS, the federation layer can be designed in two different ways: a) federation layer provides one global interface to different spatially distributed MRDBMSs to unite them into one big MRDB which consists of same number of representation levels with same scales. b) federation layer itself acts as the MRDB by integrating the component DBSs. Every component DBS stores one specific representation level (e.g. BaseDLM, DLM50, DLM250) and these databases are called representation databases (r-databases) [30]. The federation layer should contain a working database which manages multiply represented entities of component databases. Figure 4.4 shows the architecture of an MRDBMS according to Friis et al. [52]. It consists of six components and is associated with a set of component database systems containing representation databases $D_1, ..., D_n$. These six components can be seen as the components in the working database in the federated layer as explained in this section. The MRSchema component represents the MR schema which manages the multiply represented entities (see section 3.7). The Multiple Representation Engine is the main processing component which is responsible
for evaluating consistency rules laid down in MR schema. It uses MR repository to store necessary information about the multiply represented entities and their consistency state. To interact with the r-databases, Multiple Representation Engine uses Action Scheduler, Object Matcher and Change Monitor components. The Action Scheduler is used to detect inconsistencies among representation databases and send restoration action requests to the r-databases. The Object Matcher is responsible for finding corresponding r-objects using object links in the underlying databases that form multiply represented entities. The Change Monitor component is responsible for monitoring r-databases for changes. The concept of this MRMS architecture is more or less similar to that of Anders and Bobrich [30]. The only difference is that Friis et al. [52] have modelled the MRDBMS with integrated approach while Anders and Bobrich [30] have used the inter-relationship approach. Therefore the architecture by Friis [52] can be used as the general architecture for an MRDBMS.

Figure 4.3: General architecture of a federated database system.
4.5 Data model for the MRDB

Figure 4.5 shows the conceptual data model designed to represent building and road features in MRDB. The relations between feature classes at two levels are established as associations in the data model. The reason for modelling two representation levels with association relation is the use of inter-relationship approach to model MRDB (see section 3.5) where the objects at two levels have only a connection (general relationship) representing same real world phenomena.

For building class, there are two associations. One association is to represent a set of similar buildings at 1:10K as a reduced set of typified building symbols at 1:50K (set to set relationship). Therefore this association has many to many cardinality relation between two scales. The other association represents two situations. The first situation represents a set of similar building polygons in 1:10K as a single aggregated and simplified building polygon at 1:50K. Whereas the second situation just represents a single building polygon at 1:10K as another single simplified building polygon at 1:50K. The cardinality of the association of this feature class is therefore one to many relation.

Road feature class in the source data set represents segments of road centre lines as objects. Cross road junctions in the source data set are also a simplified representation by just extending all roads to meet at a particular point (similar to representation 2 in figure 1.3). Therefore road classes are modelled as association between two classes (relationship of equivalence type) with one to one cardinality relation to represent one road segment at 1:10K with corresponding road segment at 1:50K.
4.6 Concluding remarks

This chapter discussed the existing situation of the data specifications and data models in the Survey Department of Sri Lanka. It further described the new data specifications and purpose of data enrichment. Then the chapter described the general architecture of MRDBMS together with its functionalities. In this management system, there is a separate database to manage multiply represented entities where MR schema is used to manage r-databases in which all rules pertaining to querying, linking features at r-databases and restoration (update propagation) rules are specified. With this architecture, both modelling concepts: integrated approach and inter-relationship approach can be modeled. Finally the chapter proposed a conceptual data model for the MRDBMS for both feature classes in this work. Next chapter deals with the implementation of MRDBMS where data enrichment, model generalization process, update propagation and querying functionality of MRDBMS are involved.

4.5.1 Object links between representations

In literature, there are several methods to make connectivities between representations (see sub-section 3.6.1). For this work, 'variant bottom-up' approach is used for number of reasons: a) even if the link is one way oriented, requests of data at both directions are possible in 'bottom-up' approach, b) if 'variant top-down' approach is used, a lot of null values are introduced in case of 1:N relationship due to increase of number of columns, c) having bi-directional connectivities is not efficient since it increases information in the databases unnecessarily and introduces a lot of null values as in the case of top-down approach, and d) in 'attribute-variant' approach, the whole MRDB is stored in one data set (a federation layer) and links are not established between representation databases (see section 4.4). For this work, since representations are only managed in a single database, the last method is not applied.
Chapter 5

Implementation of MRDBMS

5.1 Overview

This chapter starts with a brief description about the data and implementation environment used for this work (section 5.2). Section 5.3 describes the data enrichment procedures to the source data sets of buildings, roads and land use features with some important results. Then the chapter explains in detail the generalization process adapted by describing the generalization specifications, operations and algorithms, establishment of links of objects between two levels with the generalization results at each stage (sections 5.4) and finally evaluation of generalization (section 5.5). The chapter further describes the system architecture of MRDBMS in ArcGIS adapted in this work (section 5.6) and its implementation in ArcGIS platform (section 5.7). Section 5.8 of the chapter describes the concept of update propagation and its specifications used in this work together with some results of updates. Section 5.9 describes the conflicts that arise due to insertion of new features and updates of these features to target scale. Finally the chapter describes the specifications for queries in the MRDBMS and results of some queries (section 5.10).

5.2 Data and implementation enviornment

Topographic base data set described in section 4.2, in map sheet number 73-05, covering western part of the country (Kalutara area) is used. The overall region covered by this map sheet is $40 \text{km}^2$ (8km x 5km) consisting of both populated area (urban) and sparsely populated area (rural). All examples included in this thesis are within the above map sheet.

For the implementation of the multiple representation database, ArcGIS software has been chosen as the implementation platform due to several reasons: a) ArcGIS software is very much popular and user friendly commercial software among GIS community, b) ArcGIS is the software used by the Survey Department of Sri Lanka to create and maintain topographic information databases, c) data model used in ArcGIS geodatabase has built in behaviours to check topological consistency, data integrity together with custom domains to cus-
5.3 Data enrichment

As mentioned in the previous chapter, enrichment of the source data set with generalization-specific information helps reduce computation time and deriving links between data sets. Three feature classes; buildings, roads and land use are considered in data enrichment phase. In this work, all data enrichment methods have been automated and geoprocessed with Python Programming Language (PPL) [38], [32], [40], [56], [72] & [36] in ArcGIS.

5.3.1 Data enrichment for building features

Several data enrichment methods are needed for building features before generalization process.

Adding unique object identification number

The existing building feature base data set does not have unique object identifiers. Therefore a new attribute field called 'FID _10K' is added to the building feature data set and a unique feature identification number is calculated to each feature automatically by PPL. Table 5.1 shows the enriched attribute field 'FID_10K' (see appendix D:D.1 for dialogue menu).

Classification of features into category

Each building feature has a non-unique geographic feature code (GFCODE) according to existing data specifications (see sub section 4.2.3). This code is classified into another attribute field called 'CATEGORY' based on the usage of buildings (e.g. commercial, residential) according to classification hierarchy (see figure 4.2). Purpose of this classification is described in sub section 4.2.4 in detail. This field is inserted into the base data set and is populated with attributes according to new data specifications (see appendix A) automatically by PPL. Table 5.1 shows the enriched attribute field 'CATEGORY' (see appendix D:D.2 for dialogue menu).
Classification of features on importance

Distinguishing between important and unimportant buildings in the source data set is an important factor to determine the buildings to be represented in the target generalized scale. Because important buildings need to be always retained in the generalized scale. Therefore this enrichment on importance value is essential in selecting features to be eliminated at the target scale, based on the criteria defined in the generalization process. Building features are enriched in the source data set with an attribute called 'PRIORITY' with relative importance based on their cultural and social values (see data specifications in appendix B). Important buildings are assigned with a value '1' and unimportant buildings relative to important are assigned a value '2' in the 'PRIORITY' field which is automated using PPL. Table 5.1 shows the enriched attribute field 'PRIORITY' (see appendix D:D.3 for dialogue menu).

Partition features into regions

Partitioning involves identification of homogeneous geographic regions of the area concerned to separate areas with different characteristics such as urban areas, rural areas, hilly areas and flat areas [48]. It is required to understand and characterize of the geography of the area to have amplified intelligence in the context of generalization. For example, there are cases where rural-like buildings appear inside urban areas and places where buildings have a urban-like pattern in rural areas [37]. Even though this type of cases are rare, they can exist due to improper land use management. Partitioning buildings into regions is important because importance of buildings in rural areas is more higher than that of in urban areas due to small number of buildings. Therefore, even unimportant buildings in rural areas should also be considered to preserve in the generalized target scale. An algorithm for area partitioning has been developed for generalizing Ordinance Survey (OS) master map in which criterion used to find out rural amalgams is based on area of a particular cluster. Clusters having area greater than 200000m$^2$ are considered to be rural amalgams and others considered to be urban amalgams. These clusters have been created by proximity analysis [37]. Clustering algorithm developed in this research is also based on proximity analysis and the algorithm itself is not used for area partitioning since it is difficult to distinguish between urban and rural clusters based on the area of the clusters (no sound reason to impose area threshold as in OS for partitioning). Therefore the criterion used in this work for partitioning these two regions is based on the spatial land use pattern in Sri Lanka. For this purpose land use features are also enriched with an additional attribute called 'CATEGORY' in the base data set automatically (see appendix C for more details). Building features in regions where land use 'CATEGORY' is 'Homestead', 'Builtup' or 'Recreation', are considered to be urban, while rest is considered to be rural. However, building features in 'Homesteads' which are totally surrounded by rural land use categories are considered to be rural based on social and cultural knowledge. Algorithm used for this work enriches
building features with an additional attribute field called 'REGION'. Buildings having attribute 'R' are considered to be rural and 'U' considered to be urban in this field. Table 5.1 shows the enriched attribute field 'REGION' (see appendix D:D.4 for dialogue menu).

Table 5.1: Attribute table of building layer of the source data set

<table>
<thead>
<tr>
<th>Attributes of Buildings</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Occupancy</td>
<td>Main Use</td>
<td>Age</td>
<td>Height</td>
<td>Region</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Occupied</td>
<td>Residential</td>
<td>2</td>
<td>3</td>
<td>Rural</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Unoccupied</td>
<td>Commercial</td>
<td>1</td>
<td>2</td>
<td>Urban</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Occupied</td>
<td>Industrial</td>
<td>4</td>
<td>1</td>
<td>Rural</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Unoccupied</td>
<td>Residential</td>
<td>3</td>
<td>2</td>
<td>Urban</td>
<td></td>
</tr>
</tbody>
</table>

Creation of building clusters

Spatial clustering facilitates finding the similarity or dissimilarity between geographic objects and group them into clusters in terms of a variety of geometric, topological and semantic properties. Geometrical properties such as shape and roof detection (in three dimension (3D)) of buildings help identify building clusters. This is also called structure recognition. Finding out topological relations such as proximity between features and alignment of features (e.g. buildings along roads) is another approach for cluster detection. Feature semantics in other words meaning of features (e.g. buildings having same usage) also forms similar clusters. Cluster detection for building features is an important analysis for generalizing building features. This initially requires to automate the extraction of spatial relationships between buildings. Upon cluster detection, it can be easily determined which generalization operations can be applied on each cluster. For example, in model generalization, simplification operation can be applied on single buildings while operations aggregation and typification can be performed on group of buildings.

Several cluster detection algorithms for building features have been developed up to now. Regnauld [12] has presented a method to detect clusters for building features based on the regularity of the distance between neighbouring buildings using graph-based method called minimal spanning tree (MST). ESRI [19] has described a method developed to aggregate group of buildings selecting road network as a constrained feature to avoid aggregating buildings across road network. Doihara [65] has developed a clustering algorithm to group buildings based on the minimum distance between two buildings calculated from building centroids and storing all links between buildings by their lengths, adapting an
Chapter 5. Implementation of MRDBMS

iterative procedure until all buildings are grouped. Jiang [34] has discussed two spatial clustering algorithms: hierarchical and non-hierarchical techniques not only applicable to building features, but also to other features such as road networks. The hierarchical methods put different objects into appropriate clusters iteratively creating hierarchically organized clusters. Non-hierarchical method partitions objects into different non-overlapping clusters such that the partition optimizes a given criterion, thus only one pattern with requested number of clusters will be formed.

The algorithm developed in this work is also non-hierarchical but is different from the non-hierarchical algorithms discussed above. It is based on proximity analysis of building features using buffering functionality in ArcGIS. This algorithm can be used to find clusters for both polygons and point features for vector data. Clustering has been constrained with semantics of buildings (i.e. buildings with different categories are not clustered), region of buildings (i.e. buildings falling within urban area are not grouped with buildings in rural areas) and road network (i.e. buildings are not clustered when they are separated by roads). This algorithm has been automated with PPL in ArcGIS (see appendix D:D.5 for dialogue menu). The algorithm works as follows.

1. Create a buffer around a single building feature with the threshold parameter to create building cluster. By default, this parameter has been set to 12.5m considering the minimum resolution of an object that can be distinguished by human eye on the map. This resolution is 0.25mm according to Joao [15]. Therefore distance threshold at 1:50K is 12.5m. Then constrain this buffer selection with building semantics and region of the building used to make the buffer (i.e. constrain selection with attribute ‘CATEGORY’ and ‘REGION’ of the building around which buffer is created using subset selection).

2. If this selection contains more than two buildings, assign a GROUP_NO for these selected buildings if any one of the buildings does not belong to a GROUP_NO; (i.e. GROUP_NO = 0). If any one of the buildings already belongs to a group, assign that particular GROUP_NO to all selected buildings.

3. Go to the next single building (i.e. building with GROUP_NO = 0 and follow the above two steps and iterate through all single buildings in the data set.

4. Upon completion of above steps, now there exist only building groups to be clustered. Follow the steps 1 to 3 for selection of two buildings after applying similar attribute constraints ‘CATEGORY’ and ‘REGION’.

5. Once above four steps are completed, there can still exist building pairs in groups which belong to another grouped cluster. This situation illustrates in figure 5.1. When the buildings in group no.1 is buffered, buffer ring shown in Blue dotted lines do not select buildings having group no.2. Therefore only the buildings in group no.1 are grouped when running the
5.3. Data enrichment

![Figure 5.1: Clustering buildings with creating buffers.](image)

algorithms for more than two buildings. After this when the algorithm is run for group of buildings, the other two buildings are assigned group no.2. But all buildings shown in figure should belong to one group. To sort this problem out, run the same above explained procedure this time not for a single building, but for a grouped building cluster and if building(s) belonging to another cluster(s) is selected with the same threshold parameter, as shown in figure 5.1 with bigger buffer ring, regroup the all building groups selected by this bigger buffer ring into one group (group no. 1 in this example). Follow the similar procedure for all groups.

6. Although, all buildings have been clustered in this manner, algorithm has still not considered road network as spatial constraint. Now algorithm, creates outward buffers for each grouped buildings with the same threshold used above and dissolve these buildings with the use of GROUP_NO to create aggregated polygons. Then perform inward buffering with same threshold for aggregated polygons to create aggregated buildings in another feature class.

7. Create a small positive buffer (0.5m) to all road features to cut out aggregated building polygons which lie across road features. Then erase aggregated building polygons with buffered road polygons to create multipolygon building features. Then convert this multi-polygon features to single polygons. Now the polygons are separated and do not cross road network.

8. Now select building clusters by intersecting buildings with separated polygons and regroup building clusters. This is because, before separation of polygons from road network, all the buildings belonged to only one group. But there can still exist erroneous groups due to polygon separation. For example, there can be only one building with a group number in a polygon.

9. Therefore cluster buildings again with the above mentioned algorithm by
reselecting the buildings in each separated building polygons to finalize clustering.

Figure 5.2 shows an example for few clustered buildings of the test area. Although the two clusters encircled belong to same group according to proximity analysis, they have been grouped into two groups considering the semantics (i.e. classification of buildings). It also shows that clustering has taken into account road features as a spatial constraint. Cluster number zero (0) in the figure denotes single buildings.

![Figure 5.2: Clustered buildings in the source data set.](image)

**Sub-classification of clusters**

Sub-clustering is performed on already grouped building clusters having only square or rectangular shape buildings for further subdivision to represent a group of similar objects by a group of less objects based on proximity analysis for typification operation during generalization process. Square or rectangular buildings are only typified because, discriminative characteristics of each particular building are retained as much as possible after generalization. The optimal distance threshold value should be selected by testing the classification with several thresholds since the classification changes with this value. This algorithm has been automated with PPL in ArcGIS (see appendix D:D.6 for dialogue menu). The algorithm works as follows:

1. Find out a building cluster which has only square or rectangular buildings by calculating the number of sides of each building.
2. Select the building which selects most number of buildings (e.g. n buildings) with 'SUB_GROUP' = 0, by creating a buffer around it with the sub-cluster threshold value and assign a sub_group number.

3. Then select the next building which selects n-1 number of buildings with 'SUB_GROUP' = 0 using subset selection and assign next incremental sub_group number.

4. Repeat above two steps until algorithm selects only couple of buildings (i.e. n=2) and assign a sub_group number.

5. Then select a building with SUB_GROUP = 0 if exists and create a buffer around it with original clustering threshold (i.e. the threshold used to create main clusters) and assign next sub-group number for selected buildings if selection is greater than 1.

6. Now only single buildings remain with SUB_GROUP = 0. Iterate through single buildings and assign sub-group numbers continued from the previous step until all buildings in the main group are assigned with a sub-group number.

![Figure 5.3: Sub classification of clustered groups.](image)

Figure 5.3 shows the sub classified buildings in the main groups encircled into sub groups (e.g. sub groups 1, 2, 3 & 4 in big circle) for typification during generalization process. It can also be seen that although the building marked sub-group no.2 in top-right group should belong to the group with big ellipse when proximity is considered, due to semantic constraint (classification code) it has been grouped with the top left group. The proposed data enrichment approach is based on a six-step procedure illustrated by figure 5.4.
5.3.2 Data enrichment for road and land use features

Following data enrichment has been done on road and land use features in this work. However, for generalization of roads, some more enrichment methods such as connectivity of sub roads to main roads, length of road objects, identification of roads leading to village or settlements (necessary to be considered in elimination of roads) are needed. Data enrichment for land use has been done to assist area partitioning of building features (see sub section 5.3.1). Same enrichment functionality used for building enrichment are supported for this enrichment (see appendix D).

Adding unique object identification number

The existing road and land use feature base data sets do not have unique object identifiers. Therefore a new attribute field called ‘FID _ 10K’ is added to both feature data sets and a unique feature identification number is calculated to each feature automatically as done for building features. This is done automatically by PPL.

Classification of features into category

Each road feature has a non-unique geographic feature code (GFCODE) according to existing data specifications (see sub section 4.2.3). This code is classified into another attribute field called ‘CATEGORY’ based on the usage of roads.
Generalization process

The generalization process for this work is based on the sequential processing proposed by Peng [14] for model generalization as illustrated in Table 2.1. The process is sequential because operations must be ordered in a proper way to establish a logical sequence in underlying process to derive generalized database. This is done in order to avoid repetitions of the process and frequent corrections and different results by different orders of operations. Selection is the first operation in any generalization process because if objects are not selected, it would be meaningless to apply any other operations on these objects. Reclassification of objects can then be done on selection. Aggregation takes priority over typification because depending on the results of aggregation of building groups, some groups need to be re-clustered for typification (see sub section 5.4.2 for detailed explanation). Simplification should be executed after deletion has been conducted. Because it is not necessary to simplify an object if it eventually be eliminated. Deletion should not be conducted before aggregation is performed, as a group of adjacent small objects may be aggregated into a single one of which the size is larger than criterion for deletion. Generalization operations have been automated with the use of Python Programming Language (PPL) and Arc Macro Language (AML) [39]. The generalization operations in this work are described in this order.

- Selection: extraction of purpose and scale adapted objects or group of objects based on database attributes.
- Reclassification: reclassify objects into another category to enable aggregation with objects having the same class.

Figure 5.5: Example of possible aggregation of two close area objects.
Chapter 5. Implementation of MRDBMS

- **Aggregation**: combines an objects with other objects of the same or a similar class to a new object. Figure 5.5 shows possible examples of aggregation of close area objects. In figures 5.5a, 5.5d & 5.5e, the orthogonality of boundaries of these features is preserved while the boundaries of rest of the figures are non-orthogonal.

- **Typification**: transformation of an initial set of objects into a subset, while maintaining and preserving the characteristics of distribution and pattern of the original data set.

- **Deletion**: removes an object from the data set

- **Simplification**: reduces the granularity of an outline of an object.

Figure 5.6 illustrates the entire generalization process.

![Diagram of Generalization process](image)

**5.4.1 Specifications for generalization operations**

- **Selection; criterion**: all unimportant urban single buildings, having area \( < 156m^2 \) are attributed with value '1' in a field called Remove.50K in order to eliminate buildings from the small scale data set. This area is selected based on the minimum resolution of an object 0.25mm. Based on this resolution, minimum side length of a building is 12.5m at 1:50K scale.
Therefore objects having area less than 12.5m x 12.5m (approximately 156m²) are removed.

- Reclassification; criterion: buildings having similar GFCODEs are categorized into a different class called ‘CATEGORY’. This is preprocessed during data enrichment but performed in generalization process.

- Aggregation of building clusters; criteria: a) at least one building or more having complex shapes (i.e. sides of a building > 5) in a main group; b) area of the polygons to be aggregated larger than the criterion for deletion.

- Typification; criterion: building groups consisting only square or rectangular buildings are classified into sub-groups considering the spatial distribution and average orientation.

- Deletion; criteria: all buildings enriched with attribute value 1 in the field Remove_50K are eliminated while buildings having an attribute value 0 are retained during simplification process. Buildings are also deleted at target scale when the area of aggregated buildings is less than the threshold for deletion (see figure 5.7 for an example).

![Figure 5.7: Example for an aggregated building, outline highlighted to be deleted from target data set according to area threshold.](62)

- Simplification: although simplification affects geometry of features, simplification operator is applied on both single buildings and generalized aggregated buildings in the model generalization phase in this work (see sub-section 2.3.2); criterion: buildings having sides less than a particular threshold. This threshold distance should be found out by examining the building granularity and side lengths of the data set.
5.4.2 Generalization algorithms

Generalization algorithm is a set of logical rules arranged in order to implement a particular generalization operator or operators which in turn define the spatial transformations necessary to achieve generalization (see section 2.8) in a generalization software. This section describes the vector generalization algorithms used for the following generalization operations. These algorithms have been implemented in ArcGIS software for this work.

- Selection:
  - read area of each building record in the large scale data set,
  - select the single buildings with GROUP_NO = 0,
  - find out the buildings subject to criterion of elimination, and
  - populate the attribute field 'Remove_50K' with value '1' (see table 5.1 for population of the above attribute field).

Appendix E:E.1 shows the dialogue menu interface to manipulate selection operation in ArcMAP.

- Reclassification:
  - classify each object based on GFCODE into different categories according to the new specifications, and
  - populate the field called 'CATEGORY' of each object with new category (see table 5.1 for population of the above attribute fields).

Appendix D:D.2 shows the dialogue menu interface to manipulate classification operation in ArcMAP.

- Aggregation: area aggregation functionality in workstation ArcInfo 9.0 is used. This has been further customized to input its parameters in dialogue menu (not in command prompt as in ArcInfo) and to run in ArcMAP data frame using AML and PPL (see appendix F for this tool). Also it has been further extended with the facility to apply aggregation only for specific clusters in the data set by AML (clustered identified by group numbers in this work). This facility is not available in the standard aggregation tool in workstation ArcInfo in which aggregation is applied to the whole data set only.

This function first converts the buildings of input building group to grid (raster) and uses grid functions EXPAND and SHRINK to group buildings within the specified distance of each other in creating the combined raster feature [19]. The distance between buildings used in this function should not be less than the clustering distance threshold used under data enrichment in order to capture all buildings in creating the raster feature (use the same threshold or higher value depending on the results). Then this feature is converted back to vector with proper construction of new boundaries. Cell size for raster to vector conversion must be very small.
for orthogonal features like buildings. Therefore the default cell size according to ESRI is \textbf{0.3m} in order to capture sides of buildings at least \textbf{1m} in length (see ArcInfo help on AreaAggregate command). Also, this function has the capability to create non-orthogonal features (see figure 5.5f). But aggregation is performed only with orthogonal option on buildings because sides of buildings are more or less orthogonal by nature. It is better to test parameters with a small sample data before processing a large amount of data. Aggregation distance and cell size used for this work is 15m and 0.2 respectively after testing several parameters. Steps for aggregation operation for this work is as follows:

- select the main groups which do not have sub groups to exclude typifying groups,
- run aggregation function developed in ArcMAP for each group to create aggregated buildings through ArcGIS interface (see appendix F for this tool),
- if the result of aggregation is not satisfied due to uneven arrangement of buildings in clusters (e.g. buildings situated side to side mostly in diagonal direction as shown in figure 5.5f), algorithm does not aggregate buildings in such clusters (see figures 5.8 and 5.9 for possible cases). In such cases, user can interactively identify and recluster groups according to generalization specifications. Such two possible cases are described below: case1; building groups in left and right of figure 5.8 and the bottom building in the middle of figure 5.8 can be considered as single buildings by changing the attribute field ‘GROUP_NO’ = ‘0’; case2; building cluster in the left of figure 5.9 is split into two groups after aggregation (see right). This implies that original cluster should be divided into two clusters. Due to this division, building cluster at left bottom in figure 5.9 meets the specifications for typification (since buildings are square and rectangle). Therefore, sub-classification needs to be done on that particular group in order to create sub groups for typification, and
- run aggregation function again to perform aggregation to get proper result after exclusion of bad clusters (see figure 5.6 for generalization process).

Appendix E:E.2 shows the dialogue menu interface to manipulate aggregation operation in ArcMAP.
Chapter 5. Implementation of MRDBMS

Figure 5.8: Example of building clusters, located in diagonal direction; aggregated buildings in Blue colour overlayed with buildings in the source data. Aggregation has created a deformed polygon (left), while aggregation has created a polygon excluding one building of the cluster (middle). Whereas in the right, only two separate buildings created (cannot be seen due to overlay).

Figure 5.9: Example of a cluster, where buildings are located in diagonal direction; aggregated buildings of this cluster (right) have split the main cluster into two clusters.

Figure 5.10: Example of building aggregation result, buildings in the source data set before aggregation (left), result of the aggregated building (right). According to the figure at right, it can be seen that aggregation provides better results when the buildings in the cluster are approximately aligned.

- Typification: algorithm for typification is considered the average orientation and distribution of sub-classified buildings developed in PPL and used as one of the generalization tools in ArcMAP (see appendix F for this tool). The algorithm is as follows:

---

65
5.4. Generalization process

- recreating building sub groups for already grouped building clusters on proximity analysis. This is performed during data enrichment process as described in sub section 4.3.1 and after aggregation if necessary as explained in the previous section,
- calculate the new centroid \((X_0, Y_0)\) of typifying symbol by averaging the x,y of centroid of sub-group buildings,
- calculate the average orientation of sub group \(-\alpha_i\),
- find the building having maximum size in sub-group and determine its diagonal distance \((d)\), and
- find new four vertices and create square symbol using the coordinates derived from equations as described using the figure 5.11.

\[
\begin{align*}
X_1 &= X_0 - d \cos \alpha, \quad Y_1 = Y_0 - d \sin \alpha \\
X_2 &= X_0 - d \sin \alpha, \quad Y_2 = Y_0 + d \cos \alpha \\
X_3 &= X_0 + d \cos \alpha, \quad Y_3 = Y_0 + d \sin \alpha \\
X_4 &= X_0 + d \sin \alpha, \quad Y_4 = Y_0 - d \cos \alpha
\end{align*}
\]

Figure 5.11: Typification method.

Figure 5.12 shows an example of typification results of the test area. Building clusters encircled have been not typified since these clusters consist of buildings with granularity greater than 5 (see specifications of sub section 5.4.1). Also it can be seen that the average orientation of the classified groups are preserved. Appendix E.E.3 shows the dialogue menu interface to manipulate typification operation in ArcMAP. User has the ability to select a template file to create attribute schema and spatial reference of the new file of typified buildings in this menu.

- Simplification: building simplification functionality available in workstation ArcInfo 9.0 is used since this functionality is not available in ArcMap.
Chapter 5. Implementation of MRDBMS

Figure 5.12: Typified buildings in 1:50K shown in red, overlayed with buildings in 1:10K.

This function has been further customized to input its parameters in dialogue menu (not in command prompt as in ArcInfo) and to run in ArcMAP data frame using AML and PPL (see appendix F for the tool). This algorithm can be applied to disjoint buildings with or without holes and buildings connected with straight lines. No simplification is done for the buildings connected in complicated ways (see figure 5.13). During simplification, number of vertices of buildings will be reduced but the measured area will remain roughly the same as original. If the simplification tolerance applied is relatively large compared to the size of building, the building will be simplified directly to a rectangle centered at its own center of gravity but the area will remain the same. The sides of the resulting rectangle will be the same ratio as the sides of the bounding box aligned to the longest side of the original building. (see ArcInfo help on BuildingSimplify command). Also the simplification tolerance 0 is not allowed. This simplification algorithm takes geographical extent into account with CHECKCONFLICT option to find out potential conflicts between objects during simplification such as line-crossing and line-overlapping. In this work, conflicts need not be checked, since simplification is performed on single buildings and aggregated single buildings. Hence there are no potential conflicts. In this work, this operation is applied on both single isolated buildings (i.e. buildings having 'GROUP_NO' = 0 and aggregated buildings). Steps for simplification in this work are as follows:

- find the buildings having 'GROUP_NO = 0' to select single buildings from the source data set or select the generalized aggregated buildings in the target scale, and

- use the building simplification tool developed in ArcMAP through ArcGIS interface (see appendix F for the tool). The simplification tolerance used to simplify these two types of buildings were not the same (for single buildings tolerance applied is 6.1m while for aggregated buildings it is 4.5m after testing several tolerances). Because aggre-
5.4. Generalization process

The generalization process creates more complicated buildings with sides sometimes not perfectly orthogonal.

Appendix E: E.4 shows the dialogue menu interface to manipulate simplification operation in ArcMAP.

---

**Figure 5.13:** Simplifying buildings in ArcGIS.

**Figure 5.14:** Example of simplified single building having group number zero (0).

**Figure 5.15:** Example of simplified aggregated buildings. Building clusters in source data set (left), aggregated buildings of these clusters (numbers 11 and 18) (middle), result of aggregated buildings after simplification (left). Figure also clearly shows that the building clusters having sides square or rectangular are not aggregated (see specifications of sub-section 5.4.1.)
5.4.3 Establishment of links

In the process of typification, links between the 1:10K source data and 1:50K building layer for typification (building50Ksymbol) are established during the generalization process itself by creating a table to represent **many to many** relationship (see table 5.2). Links between the source data and generalized buildings both from aggregation and simplification of groups of buildings and simplification of single buildings are established in the source data layer of buildings adapting ‘variant bottom up’ approach (see table 5.2) when these two data sets are appended to create one building layer (Building50KPolygon) using ’Append/Link’ tool developed with PPL in ArcGIS (see appendix E:E.5 for dialogue menu interface).

Table 5.2: **many to many** attribute table of the typified layer (Building50KSymbol) at left, feature class table of the source building features representing links in target data layer (Building50KPolygon) with ‘LINK50K’ attribute field column highlighted at right.

![Table 5.2](image)

5.5 Evaluation of generalization

Model generalization is done for analytical reasons and therefore, by definition, needs to keep close control of the impact of generalization on data quality. It aims at minimizing errors and also has to be done under parametric control. Therefore to assess the quality of generalization results, produced from computer driven genetic algorithms, it is much easier to have built-in evaluation functions in terms of quantitative evaluation rather than comparing with manually generalized outputs. According to João (1994) [8], it was found that the generalization effects caused by manual generalization were typically greater than automatic generalization. This means that using the parameters (tolerances) in automatic generalization, model oriented quality objectives (i.e. reduction of data with minimum errors such as conflicts and significant displacements) can be maintained. Particularly due to generalization rules and specifications, it is difficult to formalize evaluation methods. For example, during model generalization, one rule would be to eliminate all buildings which are
5.5. Evaluation of generalization

less than a particular area threshold. But there are instances where this rule cannot be applied due to adaptation of these rules to context of generalization. In the above example, there may exist some buildings to be eliminated according to the area threshold for elimination, but are not eliminated since they fall in rural areas.

The generalization effects by algorithms in automated environment are a function of the format of the data (raster or vector), the generalization algorithm chosen, respective parameters (tolerances) applied and the characteristics of features (e.g. in case of lines, the number of initial points, the sinuosity of the line and how the algorithm is applied - from which end). Also effects on model generalization depend on the type and number of algorithms chosen and in which sequence they are applied. Mostly input parameter selection probably results in more variations in the final output. Figure 5.16 visualizes the generalization outcome in two areas (urban and rural) with traditional map (1:50K) of same area.

Generally, the quality of generalization can be measured quantitatively and qualitatively [10]. Criteria such as violations of topological relations can be assessed objectively while overall quality of the map image is very hard to express clearly and the assessment is quite subjective. In model generalization phase, quantitative assessment is much important than qualitative assessment because in this stage, more concern is paid to reduce the amount of details with
minimizing errors (i.e. more statistical assessment is required) rather than the overall quality of the visual output. Quantitative measures support the assessment by determining computationally to what degree the generalization specifications are violated and how closely the effects are matched with a reference (e.g. digitized template of a manually produced map). Possible measures include the following:

- **geometrical measures**: a first measure can be used to highlight cases where minimum thresholds used are violated. No tools have been developed to check this measure automatically in this work (e.g. clustering threshold, area threshold of buildings). It can be checked in this work interactively. Second measure can be used to determine the deviation of the shape of the representation with the original source data or the map that is used as reference (e.g. digitized template of a manually produced map). In this work, the evaluation of the shape can be compared with the original source to some extent (see results in figures 5.10, 5.12, 5.14 &5.15). But comparison cannot be done with the digitized map of the target scale since it represents buildings in point geometry and there is no data consistency between two scales (see figure 5.17). Also this comparison is not a good method for the evaluation of model generalization as explained above by João [8],

- **topological measures**: the purpose of these measures is to identify violations of topological relations that need to be maintained from the original source data. There are countless number of algorithms available for generalization but most of the algorithms are not capable of tracking conflicts although new algorithms can be developed to check this. The algorithm used for simplifying buildings (see sub section 5.4.2) has facility to check conflicts. But this algorithm is applied on isolated buildings in this work and hence there are no chances of having conflicts (see figures 5.14 and 5.15 for results of simplification), and
• software related measures: several aspects of software performance such as control processing unit (CPU) time, person hours spent on a particular generalization task, equipment hours and cost and entire duration of the project are needed to measure the productivity. For example, clustering algorithm developed to group buildings approximately takes around 15 minutes to process about 2000 buildings. As a qualitative part of the assessment (less involved in model generalization), evasive software related factors such as work satisfaction must also be assessed.

5.6 System architecture of MRDBMS

Figure 5.18 shows the system architecture of the MRDBMS in this work. The main difference of this system with the general architecture of MRDBMS discussed in Chapter 4 is that there are no several component databases for each representation scale, which are all managed by a working database (see section 4.4). Instead multiple representation is managed by a single database which is a personal geodatabase in ArcGIS. Representation databases at different scales in this system are represented as feature classes as shown in the figure 5.18. Object links of these two representation levels are maintained in feature class tables and geodatabase tables. Generalization tools written in Python and Arc Macro Languages are run in a new ArcObjects component called the geoprocessor. The geoprocessor manages all the geoprocessing functions available within ArcGIS. The geoprocessor implements automation using the Microsoft’s Component Object Model (COM) IDispatch interface [40]. Visualization of the representations is through the data frame in ArcMap (ArcGIS). Updating and querying between the two representations are done by Visual Basic Applications (VBA) and ArcObjects which is the development platform for ArcGIS desktop. ArcObjects is behind the menus and icons that are used to perform tasks in ArcGIS. VBA provides the programming language and ArcObjects provides objects and their built-in properties and methods for developers to access data and perform tasks programmatically.
5.7 Implementing MRDB in ArcGIS 9

MRDB is implemented in geodatabase data model in ArcInfo (ArcGIS). Generally there are three methods to create a geodatabase in ArcInfo: a) creating a schema to create an empty database in ArcCatalog with its available tools, b) importing existing data in different formats (e.g. shapefiles, coverages, INFO tables) into a geodatabase from ArcCatalog, and c) building geodatabases with CASE tools to create new custom objects and generate a geodatabase schema with the use of UML (see section 3.4 for detailed description of UML). The first method is used if there is no data yet that needs to be loaded into a geodatabase with available tools in ArcCatalog. The second method is applicable only if data is available in formats that are convertible into a geodatabase. The last method is the more systematic method of creating a new geodatabase schema with UML class diagrams which can be used to design items in a geodatabase schema such as feature data sets, feature classes, tables, geometric networks and relationships [21]. This is the method used in this work for geodatabase schema creation. The steps of this method are outlined below.

- Design the geodatabase logical model in UML.
- Export the UML model to Extensible Markup Language (XML) Metadata Interchange (XMI).
- Generate geodatabase schema from XMI with CASE tools.
- Load data into the geodatabase.

Figure 5.19 shows the steps of creating a geodatabase using CASE tools.

![Diagram showing steps of creating a geodatabase schema using CASE tools](Image)
5.7. Implementing MRDB in ArcGIS 9

5.7.1 Designing the logical model

In creating the schema, ArcInfo makes use of Microsoft Visio, which supports the UML to automate the process of creating the logical model.

Basic terminology

The following terminologies are used in ArcInfo:

- **Object class**: A database table with which behaviours can be associated. Rows in a table are object instances that can have behaviour.
- **Feature class**: A collection of geographic objects or features of the same type.
- **Feature attributes**: Properties stored as fields in a feature class table.
- **Feature data set**: A collection of feature classes with the same spatial reference.

The ArcInfo UML model

ESRI provides the basic geodatabase model in UML that can be loaded into Visio template and extended to create custom models. The ArcInfo UML model contains the object model required for using UML to model the geodatabase. The object model has five packages: Logical View, ESRI Classes, ESRI Interfaces, ESRI network and Workspace. These UML packages act as directories in which different parts of the object model are maintained.

- **Logical View**: this package is the root level and contains the other three packages.
- **ESRI Classes**: contains the portion of geodata access components necessary to create object model.
- **ESRI Interfaces**: contains definitions of interfaces implemented by the components shown in the ESRI classes package.
- **Workspace**: this holds user-defined models created as extensions to the geodatabase model.

Designing the FeatureClass model

Custom models should be created as extensions to ESRI geodatabase model. Certain rules should be followed in creating the model to enable ArcInfo to read the custom model correctly.

- All custom models should be created under the Workspace Package.
- Custom feature classes that store spatial data should be created as specializations of ESRI feature class. This allows custom feature classes to inherit the behaviour of ESRI feature class, which in turn inherits from ESRI object class.
• Classes that store non-spatial data are created as specializations of ESRI object class.

• Feature data sets are modelled in the UML as packages stereotyped as ESRI feature data set.

Figure 5.20 shows the MRDB logical model for the building and road feature classes.

Attributes

Once each class is created, attribute fields and their domains can be specified for the class. Attribute types must be specified as ESRI field types or custom domains. Most of the feature classes have their defined custom domains. If new domains are needed they can be defined according to purpose.

Custom domains

ESRI provides two types of templates for defining custom domains; range domain and coded value domain. Range domain allows to define the allowable range of values for an attribute, for example, 1-10. Where as coded value domain allows to restrict attribute values to certain types. For example, Ownership attribute may only be 'State' or 'Private'. Coded value domains have been defined for both feature classes in this work. These domains are shown in figure 5.21.
5.7. Implementing MRDB in ArcGIS

Table 5.3: Tag values used in the model.

<table>
<thead>
<tr>
<th>Tag Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeometryType</td>
<td>Used to set the geometry type for a feature class</td>
</tr>
<tr>
<td>OriginPrimaryKey</td>
<td>The name of the primary key of origin class in an association</td>
</tr>
<tr>
<td>OriginForeignKey</td>
<td>The name of the foreign key of origin class in an association</td>
</tr>
<tr>
<td>DestinationPrimaryKey</td>
<td>The name of the primary key of destination class in an association</td>
</tr>
<tr>
<td>DestinationForeignKey</td>
<td>The name of the foreign key of destination class in an association</td>
</tr>
<tr>
<td>AllowNulls</td>
<td>Used to set null values for attributes</td>
</tr>
<tr>
<td>Length</td>
<td>Used to set the length of a character field</td>
</tr>
<tr>
<td>Precision</td>
<td>Used to set the number of digits in case of integer fields, total number of digits in case of double fields</td>
</tr>
<tr>
<td>IsAttributed</td>
<td>Used to specify attributes in a relationship class</td>
</tr>
</tbody>
</table>

Exporting the model

Once the model has been completed, it has to be exported into the XML format using ESRI XMI export function. Before exporting the model into XMI, it is important that the entire model is checked for semantic errors by using Visio UML semantic checker to verify the model. This prevents eliminating errors of the model during the automatic schema generation.

5.7.2 Generating the geodatabase schema

The schema generation wizard of the CASE tools in ArcCatalog is used to generate an empty geodatabase schema from the repository UML model. Model stored in XMI file created by exporting the logical model is used to create the MRDB geodatabase schema.
Loading data into the Geodatabase schema

Once the empty schema has been created, existing data sets can be loaded into that schema. The attribute fields of the data loading into the model should conform with the empty geodatabase model. The existing generalized data sets are in shapefile format and are well structured to conform with the model. To load data into the model Object Loader tool in ArcMap is used. The Object Loader automatically converts the shapefile features into geodatabase objects and generates an error log file if any. Figure 5.22 shows the loaded data into the geodatabase to represent MRDBMS.

Figure 5.22: Source building layer (left); model generalized building layer (right) in geodatabase shows typified buildings in Red and simplified single and aggregated buildings in Blue.

5.8 Update propagation in the MRDBMS

When the real world is changed, MRDB representing real world phenomena should be updated. These updates lose consistency between layers in the MRDB if relations among objects representing the same entity are not maintained. This maintenance should be managed by MRDBMS. Theoretically, this inconsistency can be restored by a complete new generalization of the dependent layers. This method is practically not a good solution as it requires enormous amount of processing and is very time consuming. In this work update propagation is divided into three main events: a) find relevant updates b) updating itself and c) conflict detection after updates. To find relevant updates, a method that only includes features that are influenced by updates is adapted. This method is called incremental generalization as firstly introduced by Kilpelainen and Sarjakoski [9] & [70].

It is important to differentiate between the two methods in implementing an MRDBMS as described in section 3.4 at the stage of update propagation. In the first method, data enrichment procedures and generalization rules are known since links to implement MRDBMS are established during generalization process. In the second method, independently collected data sets (cartographic
data sets) with different scales are linked and kept in the database through matching procedures to create MRDBMS. Therefore, new generalization specifications, algorithms and parameters have to be defined during update propagation. These criteria can be defined based on the studies of existing maps and cartographic data sets. Hence strategies for the propagation of updates for the two cases might be different.

For this work, data enrichment procedures and generalization rules used to create the MRDBMS using the first method are adapted. This means that during incremental generalization in update propagation, user (user in this work is a person who manipulates the MRDBMS) has to adhere to the same generalization procedure defined in the generalization phase which is a rule based system to see which operator should be used under certain conditions. These conditions can be defined as type of update event, type of the object to update (e.g. a single building or a building in a group) and threshold to find out influential groups. Tools can easily be developed to run these rules in a schema, in order to automate whole process. Then user interaction is less where user just needs to click respective tools to invoke process. There are several disadvantage in this method: a) all parameters used in the process have to be hard coded (user cannot change parameters), and b) during generalization, user cannot check the output of each operation. This might lead to improper outcome at the end because there is no possibility to change parameters and rerun some operations if output is not good.

Therefore, in this work user must be aware of the MRDBMS to invoke tools and input proper parameters of respective tools (MRDBMS tools are developed separately) in update propagation. Update propagation in this work is divided into three main events as suggested by Anders and Bobrich [30]:

- Change: an object is modified,
- Insert: a new object is created, and
- Remove: an object is deleted.

The Change event is further divided into:

- Change Attribute: only attribute values are changed,
- Change Geometry: only shapes of objects are changed, and
- Change Attribute and Geometry: the attribute values and the shapes are changed.

For update propagation, Visual Basic language [16] with ArcObjects [17], [22], [35] & [64] have been used in this work.

### 5.8.1 Specifications for change event

For change of attributes in the source data set, three cases are possible; a) change of attribute of a single building b) change of attribute of a group of buildings and c) change of attribute in some buildings in a group. In the first two
cases, only the attributes of the buildings at the target data set are affected. But in the last case, geometry of the buildings for aggregated buildings and pattern and number of symbols for typified buildings may also be affected. An example for change of attribute can be considered as changing the Geographic Feature Code (GFCODE) of a building or a group of buildings.

5.8.1.1 Change of attributes

Change of attribute of a single building

- If classification of a single object is changed due to an attribute change in GFCODE, then the classification (CATEGOY) of the corresponding feature at the target scale is automatically changed with the use of object link field; LINK_50K in the feature class table (see figure 5.23 for an example of update and appendix F for update tools of MRDBMS in ArcMAP).

![Figure 5.23: Dialogue menus in the left figure show how a user interacts with the data frame of ArcMAP to update attributes. During update propagation, user can confirm the update in the target data set. Right figure shows in the circle the building highlighted in the source data set of which classification code (GFCODE) is changed by the user by selecting this building and clicking attribute update tool in ArcMAP. This update is triggered in the target scale changing the classification code. Attribute tables (left) in the right figure show the attributes of both data sets before updates (see highlighted attributes) while the tables at right show the updated attributes.](image)

Change of attribute of a group of buildings

- If classification of at least a single building in a group is changed due to change of GFCODE, then the classification (CATEGOY) of other buildings
is also automatically changed in the source data set and the corresponding aggregated feature or typified features at the target scale is also changed with the use of object link field; LINK_50K in the feature class table in case of an aggregation and link fields FID_10K and FID_50K in many to many attribute table for typified objects. Same attribute update tool is used in this change event by the user as explained in the previous case for a single building.

**Change of attributes of some buildings in a group**

- If the classification of some buildings is changed, apply the clustering algorithms (both clustering and sub clustering) for the affected group in order to re-cluster the influential group.

- Record new group numbers in separate attribute fields 'NEW_GROUP' and 'NEWSUB_GROUP' in the source data set.

- Delete the existing objects in the target scale with the use of object links.

- Follow the generalization steps to the influenced groups and recreate the objects in the target scale with new object IDs and links.

**5.8.1.2 Change of geometry**

- If the geometry is changed in a single building, find the influential group of buildings due to change of geometry using clustering algorithms.

- Record the numbers of new groups if created in a separate fields called 'NEW_GROUP' and 'NEWSUB_GROUP' in the source data set.

- Delete the existing target objects using links.

- Depending on the consequence of clustering algorithms (may be a single building will belong to a group after geometry change), apply the generalization rules and recreate the new objects in the target scale with new object IDs and links using the existing and new attribute fields to record group numbers.

- If the geometry of buildings is changed in a whole group or some buildings in a group, find the new influential groups with two clustering algorithms recording similarly as in single buildings the new group numbers.

- Delete the existing corresponding objects in the target scale with the help of links.

- Follow the generalization steps and recreate the new objects with new object IDs and links using the existing and new attribute fields used to record group numbers.

**5.8.1.3 Change of geometry and attributes**

- Follow the same procedure as explained in change of geometry event with the calculation of new attributes for newly generalized buildings at the target scale.
5.8.2 Specifications for insertion of new objects

There should not be conflicts after new insertions in the environment (with other layers such as hydrographic features and road features in the source data set). Chances of conflicts are less since this work is based on model generalization. But if there are conflicts, they need to be solved before propagation of updates to the target scale. However conflicts after update propagation in the target scale will have to be dealt with separately after update propagation. Moreover, clustering algorithm does not take into account other features in the environment (except road features) such as hydrographic features (e.g., there can be buildings in a cluster which lie on either side of a channel due to new insertions of buildings). Such cases are to be solved interactively before update propagation.

- When new objects are inserted, perform data enrichment on new objects in the source data set.
- In data enrichment, influential group of buildings due to new insertions are found.
- Record the new group numbers in the two attribute fields 'NEW_GROUP' and 'NEWSUB_GROUP' (these two fields are used to identify newly inserted objects).
- Delete the existing target objects using links if existing objects in the source data set are influenced by new updates.
- Follow the generalization steps and recreate the new objects in the target data set with new object IDs and links using the existing and new attribute fields used to record group numbers.

5.8.3 Specifications for deletion of objects

For deletion of objects from the source data set, three cases are possible: a) deletion of a single building, b) deletion of all buildings in a group, and c) deletion of some buildings in a group.

- If a single building is deleted in the source data set, find and delete the corresponding object in the target data set using object link.
- If all buildings belonging to a particular group are deleted, delete the corresponding aggregated object or typified objects with the use of links in the feature class table in case of aggregation and with the use of separate many to many table in case of typification.
- If some buildings of a particular group is deleted, it will change either the geometry of the aggregated building or the pattern and number of symbols in the typified buildings depending on the type of that particular group.
- Delete corresponding objects at both scales with links.
5.9 Conflicts after updates

- Regroup the building cluster after deletion and record the new group numbers in the two new attribute fields for new groups.

- Follow the generalization steps and recreate the new objects with new object IDs and links using the existing and new attribute fields used to record group numbers.

Figure 5.24 shows an example of how a single building is deleted with delete tool of MRDBMS in ArcMap (see appendix F for this tool). The same tool is used to delete buildings in a group in the source data set which corresponds to an aggregated building in the target data set by selecting at least one building in the respective group and clicking delete feature tool.

Figure 5.24: Dialogue menus in the left figure show how a user interacts with the data frame of ArcMAP to delete building features in delete event. During update propagation, user can confirm the update in the target data set. When user selects a single building as seen highlighted in Red and click delete tool, user is prompted to follow the delete event to trigger deleting corresponding building in the target data set.

5.9 Conflicts after updates

The MRDBMS should have functionality for detecting and resolving conflicts after propagation of updates. The updates in the source data set can cause topological conflicts not only in source data set, but also in target data set after update propagation. In GIS technology, topology is the relationship among features used to describe how features share geometry and maintain spatial relationships. The geodatabase supports an approach to modeling geography that integrates the behavior of different feature types and supports different types of key relationships. In this context, topology is a collection of rules and relationships with a set of editing tools and techniques. In ArcGIS, there are tools to analyze topological integrity of data. Topology offers a set of integrity tools...
that define the behaviour of spatially related geographic features. ArcGIS also has facility to create custom tools to validate these topological relationships. For example, after updates, a building must not overlap with another building; a building must not intersect a road and so on.

Not only topology provides the users with a set of validation rules for the topologically related features but it also provides a set of editing tools that let users find and fix integrity violations. In solving conflicts, it is important to know which feature classes are more likely to be moved. For example, when there is a conflict between a building and a road, moving the building would be more appropriate (e.g. these displacement rules are more important in cartographic generalization). For example, when features in one feature class are known to have more reliable positions than another set of features, you may want less reliable features to snap to the more reliable ones. Ranks can be assigned to the feature classes in the topology to accommodate this kind of situation with a cluster tolerance.

Also, conflicts due to symbolization of features and geometry changes due to generalization can occur among other features. These conflicts are called graphical conflicts. Resolution of these conflicts can be mainly achieved by displacement, elimination and amalgamation. However, in this work, graphical conflicts do not occur even though symbolization (in typification) and geometry changes (in simplification) occur. Reasons for not having graphical conflicts in this work are: in typification, symbol size chosen to represent in the subset is the maximum size of the building in the set of buildings in the target data set and simplification is performed on single isolated buildings.

It is important to solve conflicts of features of the source data set on new updates, before propagating these updates to the generalized data set at the target scale. Because in conflict solving phase, movement of features in the source data set might create new influential groups to be updated during incremental generalization discussed in the previous section. In accordance with the concept for propagation of updates with finding building clusters of influential sets in this work, the chances of having conflicts in the target data set are very rare. However, solving conflicts are not dealt with in this work.

5.10 Query functionality in MRDBMS

Queries can mainly be divided into three categories: a) attribute queries, b) spatial queries, and c) combination of spatial and attribute queries. In general, queries between layers of different features can be performed in ArcGIS with available tools. The same tools can also be used to make queries between multiply represented features in an MRDBMS without the use of object links. For example, finding a corresponding feature in the target data set by interactively selecting a feature in the source data set and performing 'select by location' spatial analysis tool in ArcGIS. But to deal with more complex queries, objects
links are needed. For building customized queries in ArcGIS, Visual Basic Applications (VBA) together with ArcObjects should be used like Standard Query Language (SQL) used in more advanced DBMS such as Oracle databases. Following describe some queries specified to query building features at the two data sets with the help of object links.

5.10.1 Attribute queries

These queries are performed only with the use of attributes of objects.

- Find the number of buildings that are aggregated to form an aggregated building by selecting a particular aggregated building in the target data set.
- Calculate the summation of area of all buildings in the source data set that are represented by an aggregated building in the target data set.
- Find how many buildings are in the source data set which represent buildings of a particular category (e.g. 'commercial') in the target data set.

Figure 5.25 shows the query result for the first two attribute queries under one query.

Figure 5.25: Top figures show the building cluster encircled (left) and corresponding aggregated building (right). Output form shows the number of buildings attributed for aggregation and the summation of area of each building, when a user queries these information in the target data set by selecting the aggregated building (see building highlighted right) and clicking query tool.
5.10.2 Spatial queries

Spatial queries are performed only using the spatial relationships between features.

- Select all buildings in the source data set that are within a certain distance of a selected single or aggregated building in the target data set.

5.10.3 Combination of attribute and spatial queries

These queries are performed with the use of both attribute and spatial information of objects.

- Select all buildings in the source data set that represents aggregated buildings having a particular 'CATEGORY' (attribute selection) which fall inside (spatial selection) a particular land use.

5.11 Concluding remarks

This chapter described the methodology adapted to implement MRDBMS. It started with a brief discussion about the data and GIS software used to implement MRDBMS. Then chapter described in detail implementation of data enrichment procedures by explaining the important algorithms developed with some results. After data enrichment, chapter discussed the rule based model generalization process adapted in this work starting with generalization specifications, generalization algorithms and implementation of these algorithms with generalization operations in ArcGIS providing results of the implementation at each stage. Further chapter described how links between the two levels have been created and finally how the evaluation of generalization results have been analysed. Then chapter explained the system architecture for MRDBMS used in this work and how implementation was done in geodatabase system in ArcGIS by designing and creating logical data model for the MRDBMS and uploading data into the model. Further chapter described the implementation of update propagation in MRDBMS with the incremental generalization concept and results of implementation of some specifications. Also chapter discussed the conflict situation after updates on both levels. Finally chapter described the querying in MRDBMS with specifications for some querying functionality providing an example for querying result. Next chapter deals with the overall discussion, conclusions and recommendations in this research.
5.11. Concluding remarks
Chapter 6

Conclusions and recommendations

6.1 Discussion

The overall objective of this research was to enhance the current functionality of GIS data management software to support MRDBMS. This is because of the fact that automated real-time map generalization is presently not feasible - some generalization operators are still need improvements and some are still unsolved, others are too time consuming, to derive data sets or visual outputs of any scale at any time. Combination of both a multi-scale databases (MSDB) and generalization is proposed to create digital landscape models (DLMs) to remedy this situation in mutual support of each other. On one hand, MRDBMS allows flexibility to propagate updates from large scale data sets to other small scale data sets. Therefore, it is not necessary for NMOs to keep data sets independently and maintain updates separately for each data set at different scales. On the other hand, it allows analysis of data at different resolutions with query facility between different representations. Multiple representation can also refer to as several representations of the same real world phenomena at the same scale captured in different moments in time for different purposes. However, time dimension has not been considered in this research.

6.2 Conclusions

The first specific objective is to analyse the existing situation in topographic databases in the Survey Department of Sri Lanka. Three questions have been formulated to deal with this specific objective.

- How are data sets at different scales maintained, e.g. based on full automation of a basic core data set, in a multiple representation environment or independently from each other?

The Survey Department of Sri Lanka maintains several data sets at different scales. Data set at the scale of 1:10,000 is considered as the base data in the Survey Department. Data sets at smaller scales (1:10K - 1:250K) have been
produced by digitizing existing map series at corresponding scales. These data sets at smaller scales are produced and maintained independently with independent updates to data sets at different scales with different update cycles. There is no inter database consistency due to collection of data from different sources.

- What architecture and data format is used to produce and maintain data sets and what software is used to make products?

Data is maintained in ArcInfo coverage format. The data is stored as tiles (separated into map sheets) in ArcInfo workspace created for each map sheet. The software used is the workstation ArcInfo by Environmental System Research Institute (ESRI).

- Where is generalization applied in the production process? Interactively?

Currently, there is no generalization involved (even manual generalization) in the map production process.

The second specific objective is to define specifications for buildings and roads feature classes for both 1:10,000 and 1:50,000 data sets and for the MRDBMS. Three questions have been formulated to deal with this specific objective.

- What are the existing specifications for building and road classes at both scales?

The existing specifications at smaller scales have only focused on creating topographic data sets at different scales independently by digitizing existing maps. Therefore specifications depend on the feature attributes and feature geometry available in these maps. For example, in 1:50k data set building features are shown in point geometry and specifications have been defined to represent point symbols. This has reduced the quality of the data set maintained at this level. Whereas according to these specifications, objects in data sets cannot be identified uniquely. These specifications have not been based on creating representing data in an MRDB environment.

- What are the requirements for the respective topographic database system?

Major requirements for the topographic database system were identified as automatic propagation of updates from large scale to small scale data sets and analysis of data between different representation levels by creating and maintaining DLMs through model generalization in an MRDBMS.

- How to define new specifications at both scales for building and road classes with hierarchical relationships of objects at these two scales in accordance with the requirements and for MRDBMS?

New specifications for buildings and roads were made based on the generalization hierarchy concept which expresses reclassification of objects as the scale decreases (i.e. from 1:10K to 1:50K) in model generalization to create MRDBMS.
Specifications for the MRDBMS were defined as maintenance of consistency rules, consistency restoration actions such as update propagation, requests for consistency restoration actions and manipulating queries between two representation levels.

The third specific objective is to enrich the existing base data sets in order to facilitate model generalization. Three questions have been formulated to deal with this specific objective.

- What auxiliary information (spatial and thematic) is used during data enrichment to support the generalization process?

The data enrichment process can be seen as preprocessing stage for generalization to support complex generalization algorithms to optimize generalization. Thematic information such as feature identification, feature classification according to usage and priority of buildings based on importance and spatial information such as location of buildings based on land use and proximity (spatial distance between buildings) were considered during data enrichment process.

- What are the existing functionality available in the software that can be used to enrich the existing base data sets at the scale of 1:10,000 for two feature classes?

Existing functionality such as selection of features, calculation of attributes in ArcGIS software are insufficient for more effective data enrichment procedures to support generalization. In ArcGIS there are also spatial statistical tools to analyze patterns and cluster analysis. But most of these tools are useful only for raster data. Even the tools available for vector data only provide visual output rather than results in terms of figures as additional attributes. Therefore new tools were created for data enrichment.

- How data enrichment is performed interactively or automatically?

Several data enrichment operations were developed and performed automatically in this research using Python programming language such as introducing unique identifications to each object in the data set and reclassification of attributes based on generalization hierarchy for buildings, roads and land use layers, while classification of buildings on their important hierarchy, partition buildings into regions such as urban and rural areas and clustering buildings into groups were performed to building layer. The most important enrichment in this process was the clustering algorithm (vector) developed for building features in this work. It takes several constraints into consideration in this process: attribute (semantic) constraint - it clusters only the buildings belonging to a particular category (class), spatial constraint - it takes into account the region (urban/ rural) in which buildings are located (i.e. buildings in both, urban and rural areas are not clustered into one group), and road features - buildings across roads are not clustered into groups. The whole generalization process is based on the results of this clustering algorithm. Further, already clustered buildings are reclassified into sub-groups based on the same proximity analysis.
finding out buildings with close proximity (use small distance threshold).

The fourth specific objective is to find out suitable model generalization algorithms to create 1:50,000 level through automated generalization of the large scale data set at the scale of 1:10,000. Two questions have been formulated to deal with this specific objective.

- What are the appropriate generalization operations and algorithms for this research?

There are countless number of algorithms for automatic generalization. These algorithm can implement one or more generalization operations. But the generalization operations involved in both model generalization and cartographic generalization are not the same. Because model generalization is more focused on reduction of data under a statistical control where there should not be conflicts (i.e. operations such as displacement enlargement and exaggeration are not considered) among features while during cartographic generalization these operations are considered. However, there is no clear cut separation of operations in these two phases (e.g. simplification operation according to literature is applicable in both phases). Applying simplification operation during model generalization is in contrary according to the definition of model generalization. But simplification for building features was performed in this work under the argument that it can be used if features are subject to minimum displacement and do not make any conflicts on application of this operation.

Suitable model generalization operations together with simplification selected for this work to generalize building features are: selection, reclassification, aggregation, typification, and deletion. Algorithms for selection, reclassification, typification and deletion were developed with python programming language to manipulate as generalization tools in ArcMAP data frame with dialogue menu interfaces to input generalization parameters, while algorithms used for building aggregation and simplification are the algorithms developed by ESRI. These two algorithms have been implemented in workstation ArcInfo as generalization functionality by ESRI. The main disadvantage of these generalization functionality in workstation ArcInfo is the lack of user friendless. These two functions can only be applied on the whole data set and there is no flexibility to input generalization parameters. Since the model generalization tools for other operations have been created to manipulate in ArcMap data frame environment, the two generalization functions were further customized and extended with facility to enable users to work in same ArcMAP data frame environment with menus to input generalization parameter and to apply these functionality on selected features or group of features. This customization was developed with both Arc Macro Language (AML) and Python language.

- How is generalization assessment done by comparing the generalized data with original data?

The outcome of generalization depends on various factors such as the generalization algorithms used, generalization specifications, sequence of generalization operations adapted and generalization parameters. Therefore it is difficult
Chapter 6. Conclusions and recommendations

to have a direct assessment. However relative assessment can be done with the source data or with existing generalized data from manual maps. In this work, this assessment could only be done with source data. Assessment could not be done with existing generalized data (data at 1:50K) since these data is not consistent with source data.

The fifth specific objective is to model MRDB by designing a conceptual model, using UML for buildings and roads feature classes at two representation levels: 1:10,000 and 1:50,000. Three questions have been formulated to deal with this specific objective.

- What are the modeling concepts available to create and manage multiple representations of geographical data?

Two modeling approaches were studied to represent and manage multiple representation (MRDBMS): integrated approach, and inter-relationship approach. Integrated approach maintains both object correspondence and value correspondences (i.e. link between attribute) but it cannot maintain correspondence between representation objects since it integrates in a class hierarchy. Whereas inter-relationship approach maintains correspondence between group of objects but not the correspondence between attributes are maintained.

- How can these concepts be applied most optimally to Sri Lankan case?

In this research inter-relationship approach was adapted because dependencies between source and target classes are sufficient in geographic generalization to model topographic data sets in the MRDBMS.

- How geographical information at different levels can be related to each other?

Relationships between features at two levels were established as associations because of the use of inter-relationship approach to model MRDB where objects at two levels have only a connection (a general relationship). The building and road features were modelled with this approach in the MRDB. The buildings were modelled with three relationships between 1:10K and 1:50K levels: 1:1 relationship between source data and simplified single buildings at target scale, 1:m relationship for aggregated buildings of these two scales, and m:n relation for typified buildings. For road features, only 1:1 relation was considered since source data represents road segments from centre line of road features.

The sixth specific objective is to implement MRDBMS. Six questions have been formulated to deal with this specific objective.

- Which database platform and software is used to create prototype system for MRDBMS?

There are several GIS software systems: Geomedia by Intergraph, ArcGIS by ESRI, Microstation by Bently and MapInfo and database platforms: IBM DB2, IBM Informix, Microsoft SQL server and Oracle to implement an MRDBMS.
ArcGIS and geodatabase are the respective platforms used under this research due to number of reasons: a) ArcGIS is the most popular commercial GIS software familiar to many users, b) it is the GIS software currently used to maintain and create topographic data in the Survey Department of Sri Lanka, c) all programme codes written in Python and Visual Basic (except few written AML & ArcObject) are platform independent and can be used in other GIS systems, and d) geodatabase management system in ArcGIS is very powerful to maintain topographical integration and database consistency of features extended with object oriented capabilities such as custom behaviours and custom domains.

- How object links are implemented at different levels?
Features representing the same geographic phenomena at two representation levels (1:10K and 1:50K) were linked each other through object links. These links were created automatically during model generalization phase. For typification, separate table to represent $m:n$ table was created during generalization. Number of methods such as attribute variant, variant bottom up and variant top-down to establish these links have been discussed in this research. Out of these methods, variant bottom-up approach was used to represent $1:1$ and $1:n$ relations between two levels for number of reasons: even if the link is one way oriented, requests of data at both directions are possible; if variant top-down approach is used a lot of null values are attributed; having bidirectional connectives are not efficient since it introduces lot of null values; and attribute variant approach is not used since only representation are managed in a single database.

- How conceptual model is converted into logical model?
The conceptual model was converted into logical geodatabase schema by the use of ArcInfo UML model. The logical model was further extended with custom domain - coded value domain to restrict attribute values to certain types. Microsoft Visio supported by ArcInfo was used to automate this conversion.

- How to implement feature classes and tables for defined levels of representation?
Source data and generalized building layer in shapefile format were loaded into the geodatabase logical schema using object loader function in ArcMap. It was not possible to import the $m:n$ relationship table into the relationship class table created to represent $m:n$ relation in the geodatabase schema and the reasons were not investigated. However the created table during typification itself can be used to maintain links between source and typified data in target scale.

- How can update propagation be implemented with insert/update/delete functions?
The update propagation in this work is based on the incremental generalization where the concept is to identify objects influenced by new updates and apply the same generalization rules adapted in creating lower scale level (1:50K
building layer in this work). In this work, clustering algorithm developed to support generalization is an added advantage to identify the influential sets. Specifications for update events were defined and propagation of updates from source data set to target data set was developed with Visual Basic applications through established links, supported by ArcObject components in ArcGIS. Updates were tested for consistency between the two levels for update events; attribute updates and delete objects. Tools in ArcMap were developed for update propagation.

- How to implement other functionality in an MRDBMS such as querying different representations?

The ability to query between different representations for different analysis purposes are possible, once the links between different representations are established in MRDBMS. Several specifications for queries were defined in this work. Some of the queries were developed with query tools and tested by visualizing query result in ArcMap.

6.3 Recommendations

Although data enrichment procedures were performed for buildings, roads and land use features, and the model for MRDBMS was created for building and road layers, the implementation of generalization and MRDBMS was done only for building features due to limited time available. Therefore it is recommended to create MRDBMS for road layer.

The classification for clustering already grouped building features for typification depends on the distance threshold used. Therefore it needs to be further developed.

The outcome of the generalization operations were based on the selected algorithms. It is therefore recommended to assess the outcome by testing several algorithms for generalization operations.

Although specifications for propagation of updates and query events in the MRDBMS were specified, not all specifications were implemented due to limited time. More implementation is needed to realize full capabilities of the MRDBMS.

The performance of data enrichment and generalization should also be tested with different data sets.

The MRDBMS in this research was implemented in the personnel geodatabase in ArcGIS which is limited to small capacity of data storage and lacked with multi-user editing capabilities. Also MRDBMS in this research was established in a single database where links were established between feature class tables in the geodatabase itself. This linkage system is not efficient when links
between several data sets are required to maintain. Therefore it is recommended to implement MRDBMS within the federation database concept where data at different scales are maintained in different databases (representation databases) and these databases are maintained by another database (working database) which stores all links between representation databases, update propagation rules and matching rules. Access to working database from other databases can be provided by sequential query language (SQL) interface through a database connector (e.g. ADO, ODBC, JDBC). Access to the spatial data can be done directly using oracle interfaces or ArcSDE from ESRI for ArcGIS. ArcSDE is a generic database interface which can be used to connect ArcGIS to the most commercial database systems like oracle which supports high capacity of data storage and multi-user editing capabilities.
Bibliography


[48] Lee D. and Hardy P. (2005). Multiple Representations with Overrides, and Their Relationship to DLM/DCM Generalization. 8th ICA Workshop on Generalization and Multiple Multiple Representation, A Coruña, Spain, July 7-8, 2005.


# Appendix A

## Specifications for buildings

### Table A.1: Specifications for building features at 1:10K and 1:50K data sets

<table>
<thead>
<tr>
<th>Existing specifications for Buildings - 1:10K and 1:50K</th>
<th>New specifications for Buildings - 1:10K and 1:50K</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FEATURE</strong></td>
<td><strong>GFCODE</strong></td>
</tr>
<tr>
<td>Building - Unspecified (AREA/POINT)</td>
<td>BLDGA</td>
</tr>
<tr>
<td>Buddhist Temple (AREA/POINT)</td>
<td>BTMPA</td>
</tr>
<tr>
<td>Hindu Temple (AREA/POINT)</td>
<td>HTMPA</td>
</tr>
<tr>
<td>Church (AREA/POINT)</td>
<td>CHRHA</td>
</tr>
<tr>
<td>Mosque (AREA/POINT)</td>
<td>MOSQA</td>
</tr>
<tr>
<td>Schools - Unspecified (A/P)</td>
<td>SCHUA</td>
</tr>
<tr>
<td>Schools - Male Vidyapeeth (A/P)</td>
<td>MSVPA</td>
</tr>
<tr>
<td>Schools - Primary (A/P)</td>
<td>PSHA</td>
</tr>
<tr>
<td>Schools - Other types (A/P)</td>
<td>OSCHA</td>
</tr>
<tr>
<td>School - Unspecified (A/P)</td>
<td>SCHUA</td>
</tr>
<tr>
<td>Hospitals - General (A/P)</td>
<td>GHSPA</td>
</tr>
<tr>
<td>Hospitals - Male (A/P)</td>
<td>MSCHA</td>
</tr>
<tr>
<td>Hospitals - Primary (A/P)</td>
<td>PSHA</td>
</tr>
<tr>
<td>Hospitals - Other types (A/P)</td>
<td>OSCHA</td>
</tr>
<tr>
<td>Hospitals - Unspecified (A/P)</td>
<td>HUSPA</td>
</tr>
<tr>
<td>Dispersary (AREA/POINT)</td>
<td>DSPNA</td>
</tr>
<tr>
<td>Police Station (AREA/POINT)</td>
<td>POLSPA</td>
</tr>
<tr>
<td>Police Post (AREA/POINT)</td>
<td>PPSPA</td>
</tr>
<tr>
<td>High Courts (A/P)</td>
<td>HCRTA</td>
</tr>
<tr>
<td>District Courts (A/P)</td>
<td>DCRTA</td>
</tr>
<tr>
<td>Magistrate Court (A/P)</td>
<td>MCRTA</td>
</tr>
<tr>
<td>Court - Unspecified (A/P)</td>
<td>UCRTA</td>
</tr>
<tr>
<td>Park (AREA)</td>
<td>PNSPA</td>
</tr>
<tr>
<td>Sub Post Office (AREA)</td>
<td>SOPFA</td>
</tr>
<tr>
<td>Agency Post Office (A/P)</td>
<td>APOFA</td>
</tr>
<tr>
<td>Post Office - Unspecified (A/P)</td>
<td>UPSPA</td>
</tr>
<tr>
<td>Hotel (AREA)</td>
<td>HOTLA</td>
</tr>
<tr>
<td>Rest House (AREA)</td>
<td>RSHPA</td>
</tr>
<tr>
<td>Circuit Building (AREA)</td>
<td>CTSPA</td>
</tr>
<tr>
<td>Library Building (AREA)</td>
<td>LIBSPA</td>
</tr>
<tr>
<td>Light House (AREA)</td>
<td>LTHSPA</td>
</tr>
<tr>
<td>Private House (AREA)</td>
<td>PTHSPA</td>
</tr>
<tr>
<td>Other Tars Building (AREA)</td>
<td>OTBSA</td>
</tr>
<tr>
<td>Historical Sites (AREA)</td>
<td>HISTA</td>
</tr>
<tr>
<td>Bank Building (AREA)</td>
<td>BNSKA</td>
</tr>
<tr>
<td>Commercial Building (AREA)</td>
<td>CMBSKA</td>
</tr>
<tr>
<td>Factory Building</td>
<td>FCBPA</td>
</tr>
<tr>
<td>Building Under Construction</td>
<td>BUCAP</td>
</tr>
<tr>
<td>Railway Station</td>
<td>RSTA</td>
</tr>
<tr>
<td>Railway Halts</td>
<td>RHLTA</td>
</tr>
<tr>
<td>Bus Stand</td>
<td>BUSTA</td>
</tr>
<tr>
<td>Bus Stop</td>
<td>BUSHA</td>
</tr>
<tr>
<td>Filling Station</td>
<td>FLSTA</td>
</tr>
<tr>
<td>NOTE: In the new specifications for 1:10K, fields 'PRIORITY' and 'CATEGORY' are data enrichment columns while 'GFCODE' is an attribute field. In 1:50K, 'CATEGORY' is an attribute field.</td>
<td></td>
</tr>
</tbody>
</table>
## Appendix B

### Specifications for roads

Table B.1: Specifications for road features at 1:10K and 1:50K data sets

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>GFCODE</th>
<th>FEATURE</th>
<th>GFCODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expressway (LINE)</td>
<td>EXPRL</td>
<td>Railway</td>
<td>RAILL</td>
</tr>
<tr>
<td>Main Roads (LINE)</td>
<td>MNRDL</td>
<td>Railway line on Bridge</td>
<td>RLBRIL</td>
</tr>
<tr>
<td>Main road on Bridge (LINE)</td>
<td>MRBRL</td>
<td>Railway line on Bund</td>
<td>RLBNL</td>
</tr>
<tr>
<td>Main road on Bund (LINE)</td>
<td>MRRNL</td>
<td>Railway line along Tunnel</td>
<td>RTLNL</td>
</tr>
<tr>
<td>Main road along Tunnel (LINE)</td>
<td>RMTNL</td>
<td>Expressway</td>
<td>EXPRL</td>
</tr>
<tr>
<td>Main road on Causeway (LINE)</td>
<td>MRCWL</td>
<td>Main Road</td>
<td>MNRDL</td>
</tr>
<tr>
<td>Secondary/Minor Roads (LINE)</td>
<td>SDRDL</td>
<td>Main road on Bridge</td>
<td>MRRBL</td>
</tr>
<tr>
<td>Secondary/Minor Road on Bridge (LN)</td>
<td>SRBRNL</td>
<td>Main road on Bund</td>
<td>MRRNL</td>
</tr>
<tr>
<td>Secondary/Minor road on Bund (LINE)</td>
<td>SRBNL</td>
<td>Main road along Tunnel</td>
<td>MRTNL</td>
</tr>
<tr>
<td>Secondary/Minor road along Tunnel (LN)</td>
<td>SRTNL</td>
<td>Main road on Causeway</td>
<td>MRCWL</td>
</tr>
<tr>
<td>Jeep/Cart Track (LINE)</td>
<td>TRCKL</td>
<td>Minor Road</td>
<td>SDRDL</td>
</tr>
<tr>
<td>Track on Bridge (LINE)</td>
<td>TRBRL</td>
<td>Minor Road on Bridge</td>
<td>SRBRNL</td>
</tr>
<tr>
<td>Track on Bund (LINE)</td>
<td>TRBNL</td>
<td>Minor road on Bund</td>
<td>SRTNL</td>
</tr>
<tr>
<td>Track along Tunnel (LINE)</td>
<td>TRTNL</td>
<td>Minor road on Causeway</td>
<td>MRCWL</td>
</tr>
<tr>
<td>Track on causeway (LINE)</td>
<td>TRCWL</td>
<td>Jeep/Cart Track</td>
<td>TRCKL</td>
</tr>
<tr>
<td>Lane (LINE)</td>
<td>LANEL</td>
<td>Track on Bridge</td>
<td>TRBRL</td>
</tr>
<tr>
<td>Footpath (LINE)</td>
<td>FPTNL</td>
<td>Track on Bund</td>
<td>TRBNL</td>
</tr>
<tr>
<td>Footpath on Bridge (LINE)</td>
<td>FPBRNL</td>
<td>Track along Tunnel</td>
<td>TRTNL</td>
</tr>
<tr>
<td>Footpath on Bund (LINE)</td>
<td>FPBNL</td>
<td>Track on causeway</td>
<td>TRCWL</td>
</tr>
<tr>
<td>Footpath along Tunnel (LINE)</td>
<td>FPTNL</td>
<td>Major Street</td>
<td>MSTRL</td>
</tr>
<tr>
<td>Railways (LINE)</td>
<td>RAILL</td>
<td>Lane</td>
<td>LANEL</td>
</tr>
<tr>
<td>Railway line on Bridge (LINE)</td>
<td>RLBRL</td>
<td>Street</td>
<td>SRTL</td>
</tr>
<tr>
<td>Railway line on Bund (LINE)</td>
<td>RLBNL</td>
<td>Path</td>
<td>PATHL</td>
</tr>
<tr>
<td>Railway line along Tunnel (LINE)</td>
<td>RLTNL</td>
<td>Private Road</td>
<td>PRTVL</td>
</tr>
<tr>
<td>Runway (LINE)</td>
<td>RNWYL</td>
<td>Private road</td>
<td>PRTVL</td>
</tr>
<tr>
<td>Taxiway (line)</td>
<td>TXWYL</td>
<td>Footpath</td>
<td>FPTNL</td>
</tr>
<tr>
<td>Bridge (LINE)</td>
<td>BRIDGL</td>
<td>Footpath on Bridge</td>
<td>FPBRNL</td>
</tr>
<tr>
<td>Road Under Construction (LINE)</td>
<td>RUCLK</td>
<td>Footpath on Bund</td>
<td>FPBNL</td>
</tr>
<tr>
<td>Ferry</td>
<td>FRRYL</td>
<td>Footpath along Tunnel</td>
<td>FPTNL</td>
</tr>
<tr>
<td>Miscellaneous Line (LINE)</td>
<td>MISCL</td>
<td>Runway</td>
<td>RNWYL</td>
</tr>
<tr>
<td>Railway line on Bund (LINE)</td>
<td>RLBNL</td>
<td>Aviation road</td>
<td></td>
</tr>
<tr>
<td>Railway line along Tunnel (LINE)</td>
<td>RLTNL</td>
<td>Ferry</td>
<td>FRRYL</td>
</tr>
<tr>
<td>Runway (LINE)</td>
<td>RNWYL</td>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>Road Under Construction (LINE)</td>
<td>RUCLK</td>
<td>Miscellaneous Line</td>
<td>MISCL</td>
</tr>
</tbody>
</table>
## Appendix C

### Specifications for land use

Table C.1: Specifications for land use features at 1:10K and 1:50K data sets

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>GCODE</th>
<th>CATEGORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marsh</td>
<td>MRSHA</td>
<td>Inaccessible</td>
</tr>
<tr>
<td>Swamp</td>
<td>SWMPA</td>
<td>Inaccessible</td>
</tr>
<tr>
<td>Paddy</td>
<td>PDYAA</td>
<td>Agricultural</td>
</tr>
<tr>
<td>Chena</td>
<td>CHNAA</td>
<td>Agricultural</td>
</tr>
<tr>
<td>Mixed tree and other perennial crops</td>
<td>MIXDA</td>
<td>Agricultural</td>
</tr>
<tr>
<td>Barren land</td>
<td>BRRNA</td>
<td>Barren</td>
</tr>
<tr>
<td>Quarry</td>
<td>QRRYA</td>
<td>Barren</td>
</tr>
<tr>
<td>Paddy-abandoned</td>
<td>PDYAA</td>
<td>Barren</td>
</tr>
<tr>
<td>Tea</td>
<td>TEAA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Rubber</td>
<td>RBBRA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Coconut</td>
<td>CCNTA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Cinnamon</td>
<td>CNMNA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Citranelia</td>
<td>CTNLA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Cashew</td>
<td>CSHWA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Palmynrah</td>
<td>PLMRA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Oil Palm</td>
<td>OLPMA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Sugarcane</td>
<td>SGCNA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Other cultivation</td>
<td>OTHRA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Associated non-agricultural land</td>
<td>NNDNA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Sparsely used cropland</td>
<td>SPRSA</td>
<td>Cultivation</td>
</tr>
<tr>
<td>Dense Forest</td>
<td>FRSDA</td>
<td>Forestry</td>
</tr>
<tr>
<td>Open Forest</td>
<td>FRSDA</td>
<td>Forestry</td>
</tr>
<tr>
<td>Forest Plantation</td>
<td>FRSPA</td>
<td>Forestry</td>
</tr>
<tr>
<td>Forest - Unclassified</td>
<td>FRSUA</td>
<td>Forestry</td>
</tr>
<tr>
<td>Scrub land</td>
<td>SCRBA</td>
<td>Forestry</td>
</tr>
<tr>
<td>Grassland</td>
<td>GRLA</td>
<td>Grassy</td>
</tr>
<tr>
<td>Distorted surface</td>
<td>ISTSA</td>
<td>Distorted</td>
</tr>
<tr>
<td>Rock</td>
<td>ROCKA</td>
<td>Rocky</td>
</tr>
<tr>
<td>Park</td>
<td>PARKA</td>
<td>Recreation</td>
</tr>
<tr>
<td>Playground</td>
<td>PLGDA</td>
<td>Recreation</td>
</tr>
<tr>
<td>Homesteads/Garden</td>
<td>HOMSA</td>
<td>Homestead</td>
</tr>
<tr>
<td>Aquatic farms</td>
<td>AQTCA</td>
<td>Farm</td>
</tr>
<tr>
<td>Agricultural farms</td>
<td>AGRCA</td>
<td>Farm</td>
</tr>
<tr>
<td>Livestock farms</td>
<td>LYSTA</td>
<td>Farm</td>
</tr>
<tr>
<td>Unclassified</td>
<td>UNCLA</td>
<td>Unclear</td>
</tr>
<tr>
<td>Reservation</td>
<td>RSRVA</td>
<td>Reserved</td>
</tr>
<tr>
<td>Cemetery</td>
<td>CMFYA</td>
<td>Crematorial</td>
</tr>
<tr>
<td>Sand</td>
<td>SANDA</td>
<td>Sandy</td>
</tr>
<tr>
<td>Built up area</td>
<td>BLTPA</td>
<td>Built-up</td>
</tr>
<tr>
<td>Plantation area (Photogrammetry data)</td>
<td>PLTNA</td>
<td>Plantation</td>
</tr>
<tr>
<td>Mangrove</td>
<td>MNGRA</td>
<td>Plantation</td>
</tr>
</tbody>
</table>
Appendix D

Menus for data enrichment

D.1 Adding unique Feature ID

Figure D.1: Menu in ArcMap to enrich Feature-ID.

D.2 Classification features into category

Figure D.2: Menu in ArcMap to classify features.
D.3 Prioritize buildings

Figure D.3: Menu in ArcMap to prioritize building features.

D.4 Partition buildings into regions

Figure D.4: Menu in ArcMap to partition buildings into regions.
D.5 Cluster buildings

Figure D.5: Menu in ArcMap to Cluster buildings.

D.6 Sub-classification of buildings

Figure D.6: Menu in ArcMap to sub-classify buildings.
Appendix E

Menus for generalization

E.1 Selection

Figure E.1: Menu in ArcMap to select and eliminate buildings according to specifications.

E.2 Aggregation

Figure E.2: Menu in ArcMap to aggregate buildings.
E.3 Typification

![Menu in ArcMap to typify buildings.](image)

Figure E.3: Menu in ArcMap to typify buildings.

E.4 Simplification

![Menu in ArcMap to simplify buildings.](image)

Figure E.4: Menu in ArcMap to simplify buildings.
E.5 Establishing links

Figure E.5: Menu in ArcMap to establish link between building polygons of source data set and target data set.
E.5. Establishing links
Appendix F

Generalization and MRDB Tools

Figure F.1: Generalization tools and MRDBMS tools in ArcMap.
Appendix G

Terminology

A list of often used words in this thesis has been defined to avoid misunderstanding.

**object**

An object is a real existing geographical instance belonging to one feature element (e.g. Hengelo street)

**feature class**

A feature class is a collection of geographic features with same geometry type (such as point, line or polygon), the same attributes and the same spatial reference.

**data set**

Any organized collection of data with a common theme.

**database**

A database is a collection of data that is organized so that its contents can easily be accessed, managed, and updated. The most prevalent type of database is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in number of ways. An object oriented database is one that is congruent with the data defined in object classes and subclasses.

**Database Management System (DBMS)**

A database management system (DBMS) is a set of computer programmes that organize the information in a database according to a conceptual schema and provide tools for data input, verification, storage, modification, and retrieval.
layer
A set of references to data sources such as a coverage, feature class and so on. Layers can also be used as inputs to geoprocessing tools.

coverage
A data model for storing geographic features using ArcInfo software. A coverage stores a set of thematically associated data considered to be a unit. It usually represents a single layer. In a coverage, features are stored as both primary features (points, arcs, polygons) and secondary features (tics and annotations).

shapefile
A vector data storage format for storing the location, shape, and attributes of geographic features. A shapefile is stored in a set of related files and contains one feature class.

input data
Data that is entered into a computer, device, program, or process.

output data
Data that is the result of a program or process.

parameter
In geoprocessing, a characteristic of a tool. Values set for parameters define a tool’s behaviour during run time.

table
A set of data elements arranged in rows and columns. Each row represents an individual entity, record, or feature, and each column represents a single field or attribute value.

dialogue menu
In geoprocessing, a form consisting of a tool’s parameters.

geoprocessing
A GIS operation used to manipulate data stored in a GIS workspace. A typical geoprocessing operation takes an input data set, performs an operation on that data set and returns the result of the operation as an output data set. Geo-

processing allows for definition, management and analysis of information used to form decisions.

**Component Object Model (COM)**

A binary standard that enables software components to interoperate in a networked environment regardless of the language in which they were developed. Developed by Microsoft, COM technology provides underlying services of interface negotiation and life cycle management (determining when an object can be removed from the system).