A METHODOLOGY FOR DEVELOPING REMOTE SENSING APPLICATIONS

TAUQIR AHMAD
(2004-PhD-CS-02)

SUPERVISED BY
PROF. DR. ABAD ALI SHAH
(2012)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF ENGINEERING & TECHNOLOGY, LAHORE-PAKISTAN
A Methodology for Developing Remote Sensing Applications

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science (2012)

By
Tauqir Ahmad

Supervised By
Prof. Dr. Abad Ali Shah

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING UNIVERSITY OF ENGINEERING & TECHNOLOGY, LAHORE-PAKISTAN
A Methodology for Developing Remote Sensing Applications

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

By

Tauqir Ahmad

Approved on: 2nd June, 2012

(Internal Examiner):________________________________________
Prof. Dr. Abad Ali Shah
Department of Computer Science and Engineering, UET, Lahore.

(External Examiner):________________________________________
Dr. Rahat Hussain Bokhari
Director, Computer Centre
Quaid-i-Azam University, Islamabad

________________________________________
Prof. Dr. M. A. Maud
(Chairman)
Department of Computer Science and Engineering

________________________________________
Dean, Faculty of Electrical Engineering
ABSTRACT

Remote Sensing Application (RSA) is important as one of the critical enabler of e-systems such as e-government, e-commerce, and e-sciences. In this dissertation, we argue that owning to the specialized needs of RSA such as volatility and interactive nature, a customized Software Engineering (SE) approach should be adapted for their development. Based on this argument we have also identified the shortcomings of the conventional SE approaches and the classical Waterfall software development life cycle model. In this research work, we have proposed a modification to the classical Waterfall software development life cycle model and then using this modified classical Waterfall software development life cycle model as a framework we have proposed a customized software development methodology for RSAs. The proposed development methodology works in two steps, the first step works when a RS application is being developed from scratch, and the second step operates when a change occurs to an already developed RS application. We have identified four (4) different types of changes that may occur to an already developed RS application, the proposed methodology is capable to incorporate all four types of changes. We have successfully demonstrated the applicability of our proposed methodology through three case studies.
ACKNOWLEDGEMENTS

In the Name of Allah, The most beneficent, The most merciful.

I owe my deepest gratitude to my supervisor, Prof. Dr. Abad Ali Shah, who has supported me throughout my thesis with his patience and knowledge. His wisdom has been a constant support to my work throughout the process of formulating and developing this research. His deep knowledge and systematic organization of research ideas have greatly contributed to my scientific thinking in this research. Working with him has been really a great. He is very encouraging, and was available all the time despite his very busy schedule. His innovative comments and ingenious vision greatly improved the presentation and content of this thesis. I also acknowledge the efforts of worthy Vice Chancellor, Gen. (Retd.) Muhammad Akram Khan for his keen interest in research activities in the University. I also acknowledge the financial support by University of Engineering & Technology, Lahore and Higher Education Commission (HEC) of Pakistan. I would like to include my gratitude to Prof. Dr. Muhammad Ali Maud, Chairman, Department of Computer Science & Engineering for his kind and polite attitude towards Ph.D. scholars and faculty. I am grateful to my colleagues for their healthy criticism and suggestions. I would specially like to thank National Geographic Survey of Pakistan and The Urban Unit, P & D Department, Lahore, Punjab Government for providing GIS data to support my case studies in this research work. A special thought is devoted to my parents for a never ending support and prayers for me. Finally, I want to thank all members of my family. The encouragement and support from my wife Nadia and both children Walli Ahmad and Noor Fatima, was a powerful source of inspiration and energy for me during all this time.
This thesis has been evaluated by the following examiners:

**From Abroad**

1. **Dr. Farshad Fatouhi**
   Dean, College of Engineering, Wayne State University, Detroit, Michigan, USA.

2. **Dr. Ambrish Vashishta**
   Professor, Department of Computer Science and Engineering, Wayne State University, Michigan, USA.

3. **Dr. Ashiq Anjum**
   Faculty Member, School of Computing and Mathematics, University of Derby, UK.

**From Pakistan**

4. **Dr. Rahat Hussain Bukhari**
   Director, Computer Centre, Quid-i-Azam University, Islamabad.

**Internal Examiner**

**Prof. Dr. Abad Ali Shah**
   Department of Computer Science & Engineering
   U.E.T., Lahore
Dedicated To My Parents & Family
TABLE OF CONTENTS

1. INTRODUCTION........................................................................................................1

2. LITERATURE SURVEY.................................................................................................4
   2.1 Role of Remote Sensing Applications in E-Systems------------------------------5
   2.1.1 Remote Sensing Data: Issues and Problems ---------------------------------6
       2.1.1.1 Peculiar Characteristics of Remote Sensing Data -------------------6
   2.2 Software Development Methodologies------------------------------------------10
       2.2.1 Structured Systems Analysis and Development Methodologies-----------11
       2.2.2 Object-Oriented Methodologies----------------------------------------13
           2.2.2.1 Advantages and Disadvantages of using--------------------------15
           Object-Oriented Analysis and Design Methodologies
           2.2.2.2 Comparison between Structured System Analysis &---------------16
           Design (SSAD) and Object Oriented Analysis and Design (OOAD)
       2.2.3 Web Based Applications Development Methodologies----------------------17
       2.2.4 Evolutionary Applications Domain Methodologies-------------------------21
   2.3 Limitations of Existing Agile Techniques---------------------------------------25

3. THE PROPOSED METHODOLOGY..............................................................................34
   3.1 A Domain Specific Methodology for RSAs--------------------------------------35
   3.2 Remote Sensing Method---------------------------------------------------------38
       3.2.1 Remote Sensing--------------------------------------------------------38
       3.2.2 Remote Sensing satellites---------------------------------------------38
3.3 Satellite Data Acquisition and Processing (Process a) ..........................41
  3.3.1 Satellite Imagery.............................................................................41
    3.3.1.1 Raster Images............................................................................41
    3.3.1.2 Vector data.................................................................................42
  3.3.2 Processing..........................................................................................42
  3.3.3 Radiometric Correction and Calibration..............................................42
    3.3.3.1 Cloud Removal.............................................................................44
  3.3.4 Geo-Referencing.................................................................................44
  3.3.5 Image Enhancement.............................................................................44
  3.3.6 Image Classification............................................................................49

3.4 Identification Process..............................................................................51
  3.4.1 Image Interpretation and Characterization.........................................52
  3.4.2 Data Dissemination Issues.................................................................55
  3.4.3 Data Provenance Issues......................................................................56
  3.4.4 Potential of Replication and Provenance Techniques..........................57

3.5 System Development at Time Instance $t_0$..............................................59
  3.5.1 Analysis Phase....................................................................................59
    3.5.1.1 Metadata Extraction/Annotation..................................................61
      3.5.1.1.1 Building metadata model.........................................................63
      3.5.1.2 Layer Determination Model.......................................................65
      3.5.1.3 Analysis Report.........................................................................66
  3.5.2 Design Phase......................................................................................66
    3.5.2.1 Design Report..............................................................................68
3.5.3 Image Incorporation and GIS Storage Phase--------------------------68

3.6 System Development after Time Instance $t_0$-----------------------------------68
3.6.1 Types of Changes----------------------------------------------------------69
3.6.2 Similarity Check-----------------------------------------------------------72
3.6.3 Functionality of Pre-Analysis Phase----------------------------------------76

4 OPERATORS FOR FUNCTIONAL REQUIREMENTS ---------------------------------79
4.1 Distance Operator------------------------------------------------------------81
4.2 Area Operators---------------------------------------------------------------82
  4.2.1 Minimum Area and Maximum Area Operators-------------------------------83
4.3 Operators for Change Tracing-----------------------------------------------86
  4.3.1 Operator to Trace Change in Data-------------------------------------86
  4.3.2 Operator to Trace Structural Change (STC)-----------------------------87
  4.3.3 Operators to Trace Combined Data and Structural Changes-------------89

5 CASE STUDIES------------------------------------------------------------------111
5.1 Case Study I---------------------------------------------------------------111
  5.1.1 Development of Islamabad Geographical System (IGIS) at $t_0$-------113
  5.1.2 System Development after Implementation/Maintenance---------------129
5.2 Case Study II---------------------------------------------------------------134
  5.2.1 Development of System at very first image (time $t_0$)--------------135
5.3 Case Study III-------------------------------------------------------------141
5.4 Storage Structure (Hash Table) for Images Taken for all three case studies----148
6 RESULTS, ANALYSIS AND DISCUSSION

6.1 Results of Case Study I and their Analysis

6.2 Results of Case Study II and their Analysis

6.3 Results of Case Study III and their Analysis

6.4 Analysis of Three Case Studies

6.5 Discussion

6.5.1 About Achievements of Goals and Objectives

6.5.2 Problems and Difficulties Faced

6.5.3 Lesson Learned

7 CONCLUSIONS AND FUTURE DIRECTIONS

7.1 Conclusions

7.2 Future Directions

REFERENCES

APPENDICES

Appendix I. Pseudo Code to Capture the Types of Changes

Appendix II. Algorithmic Representations of Operators and Storage Structure
LIST OF TABLES

Table 2.1: Comparison of remote sensing and conventional data--------------------------------------7
Table 2.2: Comparison between SSAD and OOAD--------------------------------------------------------16
Table 2.3: Comparison between different WBA development methodologies -----------------------------20
Table 2.4: Comparison of different software development frameworks/methodologies-------------------27
Table 2.5: Comparison between different SDM over specification parameters---------------------------28
Table 2.6: Comparison of different SDM over Component based parameters -----------------------------28-29
Table 2.7: Frameworks and methodologies supporting remote sensing data-----------------------------30-31
Table 2.8: Difference between biological and remote sensing data characteristics----------------------33
Table 3.1: Layer identification for vegetation type layers---------------------------------------------62
Table 4.1: Effects of different types of changes on different cases (Case I- Case V)------------------104
Table 5.1: Analysis of satellite image by using our proposed methodology-------------------------------116
Table 5.2(a): Detail of metadata (identity information) generated at analysis phase its syntax and semantics
Table 5.2(b): Detail of metadata (reference information) generated at analysis phase, its syntax and semantics
Table 5.2(c): Detail of metadata (spatial reference information) generated at analysis phase, its syntax and semantics
Table 5.2(d): Detail of metadata (entity and attribute information) generated at analysis phase, its syntax and semantics
Table 5.3: Detail of layer identification process at analysis phase, its syntax and semantics
Table 5.4: Results of satellite image of year 1992 after analysis phase-------------------------------125
Table 5.5(a): Metadata (data quality information) extraction at design phase-----------------126-127
Table 5.5(b): Metadata (spatial data organization information) extraction at----------------127-128
design phase
Table 5.5(c): Sub-layers decomposition at design phase---------------------------------128
Table 5.6: Comparison Table for change detection in satellite images of three years-------131
Table 5.7: Analysis report generated by the analysis phase of our------------------------136
proposed methodology
Table 5.8: Area under different layers in the image of year 2005------------------------138
Table 5.9: Area under different layers in the image taken in 2010------------------------139
Table 5.10: Analysis report containing extracted metadata and determined layers--------141
Table 5.11: Area covered (sq.km) by each layer with number of pixels--------------------142
Table 5.12: Landsat TM image layers covering area in sq.km in year 1999------------------144
Table 5.13: Area covered by each layer in the image of year 2003------------------------146
Table 6.1: Results taken (using hash table) from case study I of image taken------------158
in year 1992
Table 6.2: Results extracted from the satellite image taken in year 2000 through--------159
Case study I
Table 6.3: Results taken (using hash table storage structure in Figure 5.10)------------159
through our case study I of the image taken in year 2009
Table 6.4: Change detection between images taken in 1992-2000-------------------------160
Table 6.5: Change detection between images taken in 1992-2009------------------------162
Table 6.6: Change detection between images taken in 2000-2009-------------------------164
Table 6.7: Trend of change in three year images taken in 1992, 2000 and 2009------------165
Table 6.8: Change detection in images (layers) during (2005-2010)-------------------------------170
Table 6.9: Change detection in images (layers) during (1992-1999)-------------------------------173
Table 6.10: Change detection in images (layers) during (1992-2003)-------------------------------175
Table 6.11: Change detection in images (layers) during (1999-2003)-------------------------------177
Table 6.12: Trend of change in three year images taken in 1992, 2000 and 2009)-----------------179
LIST OF FIGURES

Figure 3.1: Block diagram of methodology for GIS application development ------------------------37
to capture or identification in data/information change

Figure 3.2: Flow chart showing the complete research method for GIS application ------------40
development for process (a) shown in Figure 3.1

Figure 3.3: Satellite image without image enhancement------------------------------------------45

Figure 3.4: Histogram of the XS3 (near infrared) band (displayed in red)-----------------------46

Figure 3.5: Histogram of the XS2 (red) band (displayed in green) -------------------------------46

Figure 3.6: Histogram of the XS1 (green) band (displayed in blue)-------------------------------47

Figure 3.7: Grey-Level Transformation for performing linear grey level stretching-------------48

Figure 3.8: Multispectral SPOT image after enhancement by a simple linear grey level---------49

Figure 3.9: SPOT multispectral image of the test area ----------------------------------------50

Figure 3.10: Thematic map derived from the SPOT image using an unsupervised ---------------50
classification algorithm.

Figure 3.11: Scatter Plot of the mean pixel values for each land cover class -----------------51

Figure 3.12: Detailed diagram of process (b) shown in figure 3.1 for remote sensing----------52
application development methodology

Figure 3.13: Different pixel colors show different layers in an image------------------------53

Figure 3.14: Methodology for remote sensing applications development ---------------------60
(when the system is to be developed at time instance $t_0$)

Figure 3.15: The processing unit of analysis phase -------------------------------------------61

Figure 3.16: Decomposition of processing unit” Building Metadata Model”-------------------64

Figure3.17: Decomposition of sub-phase “Building Layer Model”-------------------------------65
Figure 3.18: Output of the design phase (design report) -----------------------------------------------66
Figure 3.19: Methodology for remote sensing applications development---------------------------75
after time instance \( t_0 \) and change detection
Figure 3.20: Decomposition of sub-phase ”classification of prototypes”------------------------77
Figure 4.1: An image showing pixels of different colors, representing different layers---------81
Figure 4.2: An irregular shaped area of a layer in an image, divided into segments --------------82
Figure 4.3: Structured chart showing all five (5) cases of the fourth type of change ------------89
(data change and structural change)
Figure 4.4: Sub-cases of Case I, represented by module (a) in Figure 4.3---------------------------91
Figure 4.5: Graphical representation of fourth type of change -----------------------------------92
(data and structural change) in images taken at \( t_i \) and \( t_j \) (Case I)
Figure 4.6: Representation of sub-cases of Case II (Module (b) in Figure 4.3)-----------------93
Figure 4.7: Graphical representation of fourth type of change (data and structural)-------------96
change) in images taken at \( t_i \) and \( t_j \) (Case II)
Figure 4.8: Sub cases (Module (1) – Module (7)) of Case III of fourth type of change----------96
Figure 4.9: Graphical representation of fourth type of change-----------------------------------98
(data and structural change) in images taken at \( t_i \) and \( t_j \) (Case III)
Figure 4.10: Sub cases of Case IV, represented by Module (d) of structured chart------------99
shown in Figure 4.3
Figure 4.11: Graphical representation of fourth type of change------------------------------100
(data and structural change) in images taken at \( t_i \) and \( t_j \) (Case IV)
Figure 4.12: Sub-categories of Case V into five (Module (1), (2), (3), (4) and (5))----------101
Figure 4.13: Graphical representation of fourth type of change (data and structural change) in images taken at time instances, $t_i$ and $t_j$ (Case V)

Figure 4.14: Representing data ($\text{depletion} \rightarrow \text{growth}$) change in the images $I_i$ and $I_j$, taken at time instances $t_i$ and $t_j$, respectively

Figure 5.1: SPOT image of the forest along Islamabad highway in 1992

Figure 5.2: SPOT image of the forest along Islamabad highway in 2000

Figure 5.3: SPOT image of the forest along Islamabad highway in 2009

Figure 5.4: Land SAT 7 map of image taken in year 2005

Figure 5.5: Land use map taken from satellite image of year 2010

Figure 5.6: Landsat TM image taken in year 1992

Figure 5.7: Landsat TM image taken in 1999 of district Lahore

Figure 5.8: Landsat TM image taken in year 2003 of district Lahore

Figure 5.9: Image taken in 1992 (Case Study I) is stored as Hash Table

Figure 5.10: Image taken in 2000 (Case Study I), stored as Hash Table

Figure 5.11: Image taken in 2009 (Case Study I), stored in pixels form as Hash Table

Figure 5.12: Hash Table storage structure of image taken in 2005 (Case Study II)

Figure 5.13: Hash Table storage structure of image taken in 2010 (Case Study II)

Figure 5.14: Image taken in 1992 (Case Study III) is stored as Hash Table

Figure 5.15: Image taken in 1999 (Case Study III) is stored as Hash Table

Figure 5.16: Image taken in 2003 (Case Study III), stored as Hash Table

Figure 6.1: Area covered by each layer in different years (1992, 2000 and 2009)

Figure 6.2: Trend (frequency) of change (Type I, Type II, Type III and Type IV)
Figure 6.3: Graph showing reusability trend

Figure 6.4: Graph showing change in layers during time interval 2005-2010

Figure 6.5: Graph showing change in layers during time interval 1992-1999

Figure 6.6: Graph showing change in layers during time interval 1992-2003

Figure 6.7: Graph showing change in layers during time interval 1999-2003

Figure 6.8: Trend of change in layers during time interval [1992-1999-2003]
LIST OF PUBLICATIONS


Chapter 1

INTRODUCTION

In the last two decades, e-systems have gained wide acceptance and their usage in various sectors such as e-government, e-commerce, and e-sciences (Bagchi et al., 2001). Among the technologies that enable the e-systems, remote sensing and its applications are one of the most important (Ahmad and Shah, 2010). Remote Sensing Applications (RSAs) have been critically used in various application domains, e.g., social sciences, disaster management, surveillance, security and military (Ahmad and Shah, 2010).

The remote sensing e-systems/applications are data driven and data intensive, collecting huge volume of data/information and this information/data is not only used for short term decision making, but it also used for long term decision making (Ahmad and Shah, 2010). The heterogeneous nature of remote sensing applications and their temporal and spatial diversity compound complexity of their development process, management, and their successful execution (Anselin et al., 2004; Longley et al, 2005). The conventional software development methodologies and database technologies have some drawbacks that make them unsuitable to be used for the development of such type of e-systems (Patrick, 1993; Brown, et al, 2001). Main reasons of their unsuitability are their peculiar characteristics such as temporal instability, diverse data formats, broad context and voluminous nature etc (Ahmad and Shah, 2009). These characteristics demand a need of a new class of specialized information systems but also a customized approach towards the development of these specialized applications, e.g., remote
sensing systems/applications. A major challenge in the development of such applications/systems is to capture their high dynamicity and evolutionary nature. Such evolutionary domains necessitate a highly iterative development processes especially a highly iterative maintenance process after their development (Ahmad and Shah, 2010). Although, the agile development approaches are employed to develop such type of applications/systems but they are unable to develop their all functional requirements (Ahmad and Shah, 2010). The requirement of some applications (such as Geographical Information System (GIS)) is to trace back the history of changes that occur to an application (Ahmad and Shah, 2010). This requirement makes the application development process more complex. So, it becomes important to add time dimension (temporal information) during the development process/methodology (Ahsan and Shah, 2008).

We have figured out in this research work that introduction of new applications domain usually brings new nature of data and functional requirements, and both parameters (i.e., new data type and new functional requirements) influence the software development process and its framework. In other words, introduction of a new application domain needs a new framework of development methodology and a new development methodology based the new framework. In (Ahsan and Shah, 2008), they concluded and followed the same principle in their work.

We have identified the characteristics of Remote Sensing (RS) applications which show that the classical waterfall software development life cycle model is unable to provide framework for proposing a new development methodology for RS applications (Ahmad and Shah, 2010). As we have mentioned earlier that these characteristics and functional requirements of RS applications make the existing software development methodologies unsuitable to be used for the development of RS applications.
To the best of our knowledge that the agile methodologies, the existing software development methodologies (SDM) such as Extreme Programming (XP), Crystal methodologies, Feature Driven Development (FDD) and Rational Unified Process (RUP) do not provide an explicitly iterative mechanism or process to incorporate and store temporal information of a application (Ramsn, 2006; Moeen, 2008).

In this dissertation, we have proposed a development methodology which has both an iterative analysis process and a mechanism for storing temporal information and coupled with other specialized features such as layer modeling. These features of the proposed development methodology make it suitable to be used for the development of remote sensing applications.

The remaining part of this dissertation is organized as: In Chapter 2, we present literature survey of the area (remote sensing), and the existing software development methodologies and their limitations. Based on our findings and arguments in Chapter 2, In Chapter 3 we propose our development methodology for remote sensing applications. In Chapter 4, we have proposed a set of retrieval operators that covers most of the functional requirements of remote sensing applications such as Geographical Information System (GIS). The working of the proposed development methodology has been illustrated by using three cases studies in Chapter 5.

Results and analysis of the three case studies, the development methodology, and their discussion are given in Chapter 6. Conclusions and future directions of this research work are presented in Chapter 7.
The advancement of computing power and communication technologies have revolutionized the world through comprehension of e-systems such as e-government, e-commerce, and e-sciences (Bagchi et al., 2001). The information collected varies from business, medical and personal information to scientific, satellite and surveillance information (Fang, 2002). The data describing this information may itself be spatial, multimedia or hypertext documents (Ahmad and Shah, 2009). This diversity of information and data adds an element of complexity to the management and successful execution of e-systems (Longley et al., 2005). Conventional database technologies have been deficient and unsuitable to handle this huge amount of data and information with such diversity in content and format (Patrick, 1993). E-systems have peculiar needs such as effective, efficient and real-time information retrieval (Brown, et al, 2001). This information further needs to be enriched with semantic information for discovery of patterns. Over the last two decades, most of the data and information resources covering a broad variety of topics have been made available through World Wide Web (WWW) making it the most important repository of data (Ahmad and Shah, 2009). However, with the unstructured, dynamic and heterogeneous nature of its contents (inter-connected with hyper-links), WWW adds another layer of complexity (Cooley, et al., 2005). In addition to the conventional problems of redundancy and inconsistency, various other issues like data provenance, data dissemination, interoperability, and heterogeneity need to be addressed for the successful management of Web-based data and information resources (Ahmad and Shah, 2009; Goble, et al., 2004).
2.1 Role of Remote Sensing Applications in E-Systems

Amongst the resources that enable the e-systems, remote sensing applications (RSAs) based on spatial and non-spatial data resources have attracted much attention more recently. RSAs are being extensively used for the critical applications in various domains such as social sciences, agriculture, disaster management, security and surveillance and, military (Ahmad and Shah, 2010). In social sciences, RSAs are used to monitor environmental conditions, resource management, epidemiology, archaeology, anthropology, international relations, human health conditions, law, demographic and urban studies (Ahmad and Shah, 2010; Ronald et al., 1998). Agriculture benefits from RSAs, are, for example through identification of cultural wastelands, marginal lands and monitoring of temporal behavior of vegetation (Srinivasa et al., 2003). Inland wetlands mitigate the harmful effect of pollutants, help in controlling floods, and are used by fish and wild life as breeding, nursery, and feeding grounds (Ahmad and Shah, 2010). Other direct and indirect benefits include increased yield, improvement of soil fertility, site specific management of agriculture, preservation of eco-diversity and creation of supportive infrastructure such as irrigation (Sherbinin et al., 2006; Bambaradeniya, 2003). RSAs play key role in effective disaster management. They provide, analyze and interpret data for mapping fire fuels, accessing fire effects, monitoring fire danger, and measuring progress in implementing any fire rescue and control plan (Ahmad and Shah, 2010; Emilio et al., 2003). In a similar manner, they help design contingency and mitigation plans for electrical outages, floods, tornadoes, earth quacks, volcanic eruption, tsunami, hurricanes, landslides and human caused disaster (e.g. terrorism) (Haddi et al., 2003). The remote sensing data is also critical in making timely as discussed in (Ahmad and Shah, 2010) and intelligent decisions in military operations. Accurate spatial information is critical to the concept of command, control, communication and
coordination in military operations. GIS is used by military in a variety of applications that include cartography, intelligence gathering, battle field management, terrain analysis and remote sensing of objects (Ahmad and Shah, 2010; Stanley, et al., 2004).

2.1.1 Remote Sensing Data: Issues and Problems

In Section 2.1, we have discussed the role and importance of remote sensing data which is used by remote sensing applications for decision making in many e-systems. However, issues of data dissemination and data provenance need to be addressed if these e-systems are to make any reliable real time decisions based on remotely sensed data (Ahmad and Shah, 2010).

2.1.1.1 Peculiar Characteristics of Remote Sensing Data

Remote Sensing Data is difficult to manage and disseminate because of its peculiar characteristics as shown in Table 2.1.
Table 2.1: Comparison of remote sensing and conventional data

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Remote Sensing Data</th>
<th>Conventional Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>i)</td>
<td>Distributed- Temporally and Spatially categorized</td>
<td>Centralized- Static and Stable</td>
</tr>
<tr>
<td>ii)</td>
<td>Critical Response time</td>
<td>Not critical</td>
</tr>
<tr>
<td>iii)</td>
<td>Temporal in nature: volatile</td>
<td>Stable: non volatile</td>
</tr>
<tr>
<td>iv)</td>
<td>Have validity constraints</td>
<td>Validity is permanent</td>
</tr>
<tr>
<td>v)</td>
<td>Evolutionary</td>
<td>Static</td>
</tr>
<tr>
<td>vi)</td>
<td>Uncertain, imprecise</td>
<td>Precise, Exact</td>
</tr>
<tr>
<td>vii)</td>
<td>Heterogeneous</td>
<td>Homogeneous</td>
</tr>
<tr>
<td>viii)</td>
<td>Voluminous</td>
<td>Medium to small scale</td>
</tr>
<tr>
<td>ix)</td>
<td>Broad Context</td>
<td>Limited Context</td>
</tr>
<tr>
<td>x)</td>
<td>Policy dependant</td>
<td>In most cases is not policy dependant</td>
</tr>
<tr>
<td>xi)</td>
<td>Integration with society perspective</td>
<td>May not be integrated</td>
</tr>
<tr>
<td>xii)</td>
<td>Harmonic information</td>
<td>Information source is unique for discrete data</td>
</tr>
</tbody>
</table>

i) Remotely sensed data is either constantly acquired or may be acquired on request for specific purposes (Ahmad and Shah, 2010). In both the cases, it is organized in some host primary machine using a conventional DBMS (Rajitha et al., 2007 and Gray et al., 1996). However,
fetching fresh data directly from primary site is not efficient because datasets of remotely sensed data may be organized into multiple files containing a subset of data elements categorized temporally and spatially (Gray et al., 2006).

ii) The response time of such data sets may be adversely affected when multiple queries access these datasets for the same temporal or spatial information (Buetow et al., 2003). The effect of write operations may even be more adverse as consistency is compromised resulting in conflicting results (Ahmad and Shah, 2010). Moreover, many real-time applications need to share data distributed from a primary site among multiple sites (Hoschek et al., 2000).

iii) Remotely sensed data can be categorized into temporal data and non-temporal data. Real time sensor data representing the physical world is temporal in nature (Birant et al., 2007). Temporal data objects are difficult to manage and analyze as they have validity intervals and are updated periodically (Ahmad and Shah, 2010). For example, temperature zones and its spread location is an example of temporal data in a fire management application (Kao et al., 2003 and Yuan et al., 2004).

iv) As compared to temporal data, non-temporal data do not change dynamically with time. Thus that do not have validity intervals and they do not need to be updated by periodic system updates (Ahmad and Shah, 2010). However, as most of these data and applications are hosted in a distributed environment, the issues of data dissemination related to availability, consistency and response time also affect non-temporal data (Vestal, 2005).

v) Remote Sensing Data is highly evolutionary, uncertain and incomplete. The latest findings, weather conditions, natural and human activities may invalidate the previous information and data (Ahmad and Shah, 2010).

vi) Remote sensing data is precise, exact and have clear objectives.
vii) Remote sensing data is heterogeneous consisting of sequence data, graphs, high-dimensional data, geometric information, scalar and vector fields, patterns, constraints, images, spatial information, models and, prose and declarative knowledge such as hypotheses and evidence (Ahmad and Shah, 2009).

viii) High volume of remote sensing data must be gathered and then disseminated. Remote sensing data must have: (i) **Factor comprehensiveness**, which reflects the numbers of objects that can be measured at once (ii) **Time-line comprehensiveness**, which represents the time frame within which measurements are made (i.e., the importance of high-level temporal resolution) and (iii) **Item comprehensiveness**, the simultaneous measurement of multiple items. All of these considerations suggest that in addition to being highly heterogeneous, remote sensing data must be voluminous if it is to support comprehensive investigation and interpretation (Ahmad and Shah, 2010).

ix) The usage of remote sensing data is very broad with different perceptions, contexts and objectives (Ahmad and Shah, 2009).

x) Remote sensing data is independent of the policies set by the government or any other agency.

xi) Remote sensing data may or may not necessarily be integrated with the society perspectives (Ahmad and Shah, 2009).

xii) To get discrete data, there is a unique source of information.

Conventional databases and information technologies have many limitations to overcome the problems mentioned above (Ahmad and Shah, 2010).

Now we discuss different software development methodologies available in literature to fulfill the needs of developing software in different domains.
2.2 Software Development Methodologies

The literature has classically viewed the process and the product of system projects in the early days which were developed/implemented in an un-systematic way. However, as the complexity of projects increase, software engineers thought some systematic ways of developing it. System development methodology gave a chance to subdivide a project to reduce the overall complexity (Anthony, 2007). By the use of software development methodology, the system development process seems to be transparent and provide the best control of project management thus reducing the overall risk and uncertainty (Brian, 2003). However, in the last three decades, with the advancement of satellite communication and computational technologies the challenges of developing a good information system is the main issue in computer science (Ahmad and Shah, 2010). Due to this reason, the development of a software engineering methodology suitable for a particular domain is the subject of extensive research (David, 2000).

- A purposeful framework is to be provided by the methodologies to apply some techniques and resources well in time during the software development process.
- It is possible to measure the standard of the development process.
- Methodology may provide a structural framework for the acquisition of knowledge.
- Methodologies may provide the division of labor such as analysis and design, coding and testing with different remuneration rates.

The challenge of developing a good information system is the main issue in computer science. Due to this reason, the development of a Software Engineering Methodology is the subject of extensive research.

Software development community is expanding with the advancement of information technology field. Information retrieval is done by using web technology as a tool.
Enterprise systems having reliability as an important factor is an example of web applications. A methodology is a combination of tools and techniques that may provide guidance to an information system development of large scale (Shah, 2008).

To address the specific organizational needs and enhance the functional capabilities for its business growth, an efficient information system may be used for the development, may be carried out. The development of such an information system can be done with proper planning, analysis, design and implementation (Shalom and Haan, 2006). This process is known as a System Development Life Cycle (SDLC).

The rapid increase in the performance of the hardware and of high level languages have made it possible to design and implement complex and large scale systems (Kurt, 2000).

2.2.1 Structured Systems Analysis and Development Methodologies (SSADM)

Developers were provided with the measure of control by SDCL. With the development of structured methodologies (1970’s), it was possible to design more stable, effective and maintainable systems. SSADM were process-oriented and have a less stress on entities and data modeling. This process centered approach focuses on processing of data into useful information. It is top down step by step approach to SDLC and logically moves from one phase to another (Mohammad, 2005).

The tools of SSADM are: Data Flow Diagram (DFD), Data Dictionary (DD), Entity Relationship Diagram (ERD), Mini Spec, Hierarchy Diagram, State Transition Diagram and Structured Chart (Shalom and Haan, 2006).

There are some advantages and disadvantages associated with SSADMs:

**Advantages:**

- It is easier for the project management to keep track of milestones set for the project.
• The use of graphical tools such as DFD’s makes understanding easier for the programmers and the users.

• It is a well established and well known methodology for the industry.

• It allows means for requirement validation.

• It is easy and simple to understand (Nazereh, 2008).

**Disadvantages:**

• Due to process-oriented approach, non functional requirements may be ignored.

• Management involvement is not direct.

• Due to non iterative behavior, the whole process is repeated if the requirements are changed.

• Users/Analyst interaction is not adequate.

• The only tool provided for user communication is logical design and DFD’s so user can not measure the progress of the project.

• It is difficult to separate the functional decomposition and to start building the system

• Object-Oriented programming languages cannot take advantages of SSAD

• Evolutionary change is not accommodated

• It does not provide the continuous development process, rather ‘one off” view.

• Reusability does not exist

To overcome some of the difficulties as mentioned above posed by the structural systems development and design methodologies, object-oriented methodologies were introduced (Kassab, et al., 2007).
2.2.2 Object-Oriented Design Methodologies (OODM)

The evolution of OODM represents the cutting edge of model driven systems development (Shah, 2003).

Quality, enhancement and maintenance of system development may be accelerated by using Object-Oriented Methodology (Michael, 2001). It is a bottom up approach and provides the ability of reuse/modify existing objects for a useful business application. The analysis model can be redefined and modified for implementation at the design phase. The conventional methodologies are used from scratch for the system development where as the object oriented methodologies can reuse existing code components as per system requirements (Masrek, 2008).

It provides an environment in which software is a combination of discrete objects. These objects are encapsulated with data as well as functions for modeling the real world “objects” (Luiz, 2003).

To develop a system that guarantees the organizational growth and survival is the fundamental challenge of the day. The organization is under pressure to have a faster system and at the same time it should deliver with high quality and low cost (Krishnan and Ulrich, 2001). A large number of software development paradigms have been proposed in the field of information modeling. To specify and develop complex information systems, latest technologies are being used in the business applications (Booch, 2000). One of the most prominent, emerging and accepted approach is Unified Modeling Language (UML). It is a standard graphical language for specification, construction, visualization and documentation of object oriented development process. UML has been adopted and rapidly emerged as a standard language for Object Oriented Systems Development by the Object Management Group (Jeffrey, 2003). The UML lovers claim an attraction due to its simplicity and intuitive notations which are understandable even for
nonprogrammers. On the other hand, many researchers and practitioners have concerns about its complexity and lack of comprehensive methodology (Yogesh et al., 2002).

UML is a standard for using some predefined notations for modeling a software system and then the artifacts involved are defined and documented (Brian, 2003).

There are some objectives/goals for using UML:

- It is a simple visual modeling language to model systems structure.
- It Provides extensibility
- It provides a language and platform independence for system design and implementation.
- Industry standards can be incorporated.
- It Provides Object Orientated Design and apply frameworks/patterns

UML provides entire system’s structure and globally accepted standards to manage system complexities and development cost reduction (Jeffrey, 2003). However, most of the project development round the world doesn’t follow any specific modeling technique for modeling large object oriented systems to avoid risk and complexity. If UML is used intelligently, it can improve quality of software projects, scalability, completeness and time reduction (Joonwoo et al., 2005)).

UML have the following three building Blocks:

i) Model elements

ii) Relationship between models

iii) UML diagrams

The elements of UML model have the following Relationships (Torsten, 2002):

i) Association
ii) Aggregation

iii) Generalization or Inheritance

i) Realization

ii) Dependency

2.2.2.1 Advantages and Disadvantages of using Object-Oriented Analysis and Design Methodologies (OOADM)

Advantages:

Stability: The System produced using Object Oriented Techniques seems to be more resilient to any change in the system and changes can be made with less effort and time (Jorn, 2002).

Reusability: The reusability is due to inheritance and polymorphism feature of the Object Oriented Paradigm. It leads to the reduction of development and maintenance cost.

Maintainability: Maintenance, enhancement and adoptability is easy and quick in the systems produced by Object Oriented Methods (Aurich et al., 2006).

Data Accessibility: Data base design can be done in a better way hence data accessibility and usability is easy (Osinski, 2006).

User Involvement: In using OOADM, user is fully involved in the development process and feels the possession of the system (Ellen, 1998).

Standardization of Objects: Object Oriented Methods helps to standardize the objects which facilitate in design understanding and reduction of project development risk (Peter, 2008).

Disadvantages:

- Sometimes initial designs are so simple to be adequate for the system.

- Focus on code is more than enough.

- Less team work emphasis.
• Difficult to find out all necessary classes and objects needed by the system.

• Object Oriented data base system is not available.

• System performance or software performance may be decreased in some object oriented programming languages due to use of interpreters.

• Training and re-education cost is unavoidable.

• Every one hate cultural change.

• Component library is costly and difficult to manage.

2.2.2.2 Comparison between Structured System Analysis & Design (SSAD) and Object Oriented Analysis and Design (OOAD)

We give comparison between SSAD and OOAD in Table 2.2.

Table 2.2: Comparison between SSAD and OOAD

<table>
<thead>
<tr>
<th>(SSAD)</th>
<th>(OOAD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolved from Structured_Programming (SP)</td>
<td>Evolved from Object Oriented_Programming (OOP)</td>
</tr>
<tr>
<td>Process oriented</td>
<td>Data oriented</td>
</tr>
<tr>
<td>System breakup is done by DFD’s</td>
<td>System breakup is done by Use Cases</td>
</tr>
<tr>
<td>DFD’s are the basis for system components</td>
<td>UML and Class diagrams are the basis for system components</td>
</tr>
<tr>
<td>System data and processes can be separated</td>
<td>System data and processes are encapsulated into objects</td>
</tr>
<tr>
<td>Planning, analysis, design and implementation are some definable step right from start to end</td>
<td>There are some iterative, incremental, continuous testing and refinement approaches involved from start to end</td>
</tr>
</tbody>
</table>
However, some classes of application can be developed adequately by using prototype-based methodologies rather than class-based methodologies (Seiter et al., 2002). This class of applications comprises of remote sensing, computer aided construction, software development environment, and the Web applications (Yang, 2006). These applications have a special characteristic of changing their state, structure or both and other characteristics. These peculiar changes to the objects lead to make a complex class lattice and inefficient so class-based methodologies are not best suitable for such kind of applications. Some applications of such kind require to store and trace back the history of specific object changes. By consulting the history of changes, object maintenance decisions can be made. Such kind of changes, storing and history trace back cannot be handles by class-based methodologies. Therefore we need a new class of methodologies to handle these classes of applications properly, this was later on extended by Shah (Shah, 2003).

However, these new class of methodologies cannot be based on existing conventional frameworks such as System Development Life Cycle (Shah, 2003). Therefore, in this research work we are suggesting changes to existing frameworks to provide proper foundation on which such class of methodologies can build upon.

As the need for real-time interactive and online systems increased to accommodate the dynamic nature of change, web based methodologies were introduced.

**2.2.3 Web Based Applications (WBA) Development Methodologies**

In the beginning of computing, Information Systems (IS) were developed without the use of explicit SDM. The software developers use the heuristic approach to develop an IS and heavily rely on previous experience as compared to the user requirements (Shah, 2003). In most of the
cases, IS were often completed late, over budget and much more maintenance required. This is the case of “software crisis” where the systems are not developed on time, within budget and not of good quality (Johns et al., 2003).

Therefore, SDMs were developed to overcome such crisis in a systematic way. SDM’s can be defined as a set of procedures, tools, techniques, and documentation (Tero et al., 2010). Web-Based Applications (WBA) are real-time, interactive and provide a provision of dynamic information display. Hence, these present opportunities and challenges which are not normally faced in formal systems development (Masrek et al., 2008).

There are some differences between conventional methodologies and the web-based methodologies in terms of the principle and the intended viewers:

i) Many WBA are created normally to be accessible on the web rather than to transform any existing conventional information systems or to provide new functionality (Russo, 2000).

ii) The users of WA’s are possibly outsider of the organizations and often cannot be recognized in advance, thus, it becomes hard to seek the views of users at the time of determining the information requirements (Masrek et al., 2008).

The communication technology (Internet, Intranet and Extranet) differences are also existent. The requirement is also there for multi-platform ease of access, and the non-sequential behavior of the site content dependent on hypertext links to other web documents (Masrek, et al., 2008).

These days, web applications and the Internet are the most dynamic and amazingly growing computer technologies (Brian, 2009). Users of these technologies are rapidly increasing exponentially, day by day. The web has become a most popular environment for a new generation interactive computer applications (Sue, 2006). The conventional DM’s such as SADT and OODM are not capable to analyze, design, implement and test WA’s (Shah, 2003). Web
application development is not a trivial task because a large number of interlinked pages need to be developed in a systematic way (Thelwall, M., 2003). There are many Web Application Development Methodologies (WADM) available in literature such as Hypermedia Design Model (HDM), Relationship Management Methodology (RDM), Object Oriented Hypermedia Design Model (OOHDM), Object Oriented Design Method for Hypermedia Information Systems (OODMHIS) and many more (Jamel et al., 2010).

There are some problems and deficiencies in the existing hypermedia or web applications design methodologies:

- These methodologies do not explicitly follow the phases of any software development life-cycle model such as waterfall model. These methodologies only focus on the design aspect during application development.
- Maintenance of web applications becomes difficult when using these methodologies for development purposes.
- WADM do not support for gathering and creating multimedia information (Jamel et al., 2010).

Table 2.3 gives the comparison of some of the web application development methodologies in literature.
**Table 2.3: Comparison between different WBA development methodologies**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>HDM</th>
<th>RMM</th>
<th>OOHDM</th>
<th>OODMHIS</th>
<th>OODM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for hypermedia application development</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Commercially available for development tools using the methodology</td>
<td>Yes</td>
<td>Information is not available</td>
<td>Information is not available</td>
<td>Information is not available</td>
<td>No</td>
</tr>
<tr>
<td>Modeling primitives</td>
<td>Entity type, entities, components and units</td>
<td>Entities and entity type</td>
<td>Objects, classes, relationships, methods, inheritance hierarchy, subsystems, perspectives, nodes and links</td>
<td>Objects, classes, relationships, methods, inheritance hierarchy, composition, (relationships between media objects)</td>
<td>Objects, page-classes, page-class components, users, user classes, operations</td>
</tr>
<tr>
<td>Link Types</td>
<td>Application, structural perspective</td>
<td>Information is not available</td>
<td>Information is not available</td>
<td>Information is not available</td>
<td>Defined and design in Navigation model</td>
</tr>
<tr>
<td>Navigation classes</td>
<td>Links and access primitives</td>
<td>Links and access primitives</td>
<td>Nodes, Links and access primitives</td>
<td>Information is not available</td>
<td>Local navigation, instance navigation, global navigation</td>
</tr>
</tbody>
</table>
Web based methodologies provide some type of dynamic environment but scientific and evolutionary domains needs some specific methodologies to cater the needs of that specific domain (Shah, 2003). Hence, evolutionary domain specific methodologies were introduced.

2.2.4 Evolutionary Applications Domain Methodologies (EADM)

Agile SDM’s have generated a lot of interest in SE community owning to their alleged suitability for evolutionary, iterative and volatile domains such as bioinformatics and WBA’s (Ahsan and Shah, 2008). However, it is not essential that a firm Agile Method (AM) suits all settings or individuals. In a comprehensive comparison of AM’s, it is pointed out that little importance has been placed on analyzing the situations where AM’s are more suitable than others. On the basis of their results, a new domain of applications is identified in the characteristics specifications of the domain for which AMD’s are more suitable (Ahsan and Shah, 2008). The applications such as Computer Aided Construction (CAC) and the WA’s belong to this set of applications or domain (Shah et al., 2009). One special characteristic of the objects of this class of applications is that these normally change their structure (methods and instance-variables), (state/data values, or both). (Shah, 2003) also added some other characteristics which are prevalent to bioinformatics and some other scientific domains, thus broadening the scope of the applications in compliance to these characteristics.

i) Applications without Hierarchical Structure

Objects of some applications have inherent property that their objects can easily be organized into a strong hierarchical structure (Ahsan and Shah, 2008). Objects of this type of applications interact with each other in a fixed and pre-defined pattern. For example, structural organization of a company’s employees (e.g. president, executive presidents, vice-presidents, etc.) is related to each other in a strong hierarchical structure. Objects of this type of application interact with each
other in a fixed and pre-defined pattern which is defined by the hierarchy of the application. For this type of applications, the class-based methodologies are appropriate to be used in their development. But there is another type of applications such as software development environments. Objects of this type of application are loose or no hierarchical structure among its objects, and they do not interact with each other in any fixed and pre-defined pattern because they are loosely related to each other. It is also possible that an object of such type of applications is defined for only some specific purpose. We consider these applications as a class of applications, and advocate that the prototype-based methodologies are appropriate to be used for their development because the methodologies use the object-modeling technique which is consistent with the characteristics of the class of applications (Shah et al., 2009).

ii) Rapid Prototype Development (RPD)

Objects of some applications such as remote sensing, software development and Computer Aided Design and Computer Aided Manufacturing (CAD/CAM) are of dynamic and evolutionary nature. For a specific purpose a new version of an object of this type application is created, and the effect of this new version of the object is seen in overall application (Shah, et al., 2009). Sometimes different versions of the same or different objects of an application are created quickly (or rapidly) to see and achieve some desired objectives and results. This process of rapid versioning is referred to as rapid prototyping. Note that a new version of an object can be created due to structural/stature change to the object (Shah et al., 2001). We consider the applications which need this type of requirements as the class of applications. Since the prototype-based technique captures all types of changes to an object in a single uniform manner, therefore the
prototype-based methodologies are suitable for the development of this class of applications (Shah, 2001).

iii) Incremental Growth of Objects

In some applications such as RS, CAC and Virtual Reality, knowledge of the objects grow in an incremental fashion. In this type of application, an object is created with its default knowledge (i.e., the knowledge that is available at the time of its creation), and later its knowledge grows over its life-span (Shah et al., 2009). For example in CAC, an object building is defined when it is initially built, then each time the building needs maintenance and the maintenance is done, the knowledge of the object is updated (Nicholas et al., 2007). This update of knowledge shows maintenance that is done to the building. The incremental growth of the knowledge of objects can be in the structure, state or both parameters of the object (Shah, 2001). From our study of object-modeling techniques, it is obvious that the prototype-based methodologies are appropriate for the objects that grow their knowledge in incremental fashion. The class-based methodologies will design a complex and inefficient class-lattice for such type of applications (Ling et al., 2000).

iv) Trace Back Changes about an Object

In the engineering applications i.e., CAC, CAD/CAM and many more multifaceted and engineering linked applications, it is desirable to trace back the history of changes to the structure, state, or both parameters of a specific object. The consultation of the history of a specific object is helpful in making the maintenance decision for the specific object (Shortliffe et al., 2005). The history provides the maintenance engineer, complete details of all changes that occur to the object since its creation. The importance and benefits of maintaining and tracing the
history of changes to the objects are given in (Thomas, et al., 2006). The characteristic of this type of applications is an important feature which makes these applications different from others.

v) When Object Grouping is not needed

The class-based technique puts the objects into a group (or class), which share common properties, and different groups are then linked into a graph (class-lattice) (Shah, 2001). But if we have some applications (such as the application on Web and virtual reality) in which each object is its own type, and the objects of this type of application do not much interact with each other, then it is inappropriate grouping of the objects and make their class-lattice (Shah, 2001). Therefore, for this type of applications in which objects have almost heterogeneous properties, it is not beneficial to organize the objects of these applications into a hierarchical structure like class-lattice. Rather it is more useful to put all objects of such type of application at the same level as the prototype-based technique suggests (Shah, 2001).

vi) Structure and State Capturing of Objects Simultaneously

It is concluded that the class-based technique and methodologies are not capable of capturing simultaneous changes to both parameters of an object. But in some applications such as the hypermedia systems, the web systems and the semi-structure applications, this type of change to objects occurs frequently and it is essential to capture the change (Shah, 2001). This characteristic of these applications makes the class-based methodologies unsuitable for the development of this type of applications.

vii) Polymorphic Behavior of Objects.

There are certain domains in which objects exhibit polymorphic behavior. Such objects change their behavior depending on external stimuli or over temporal dimension (Frank et al., 2010).

viii) Evolutionary Behavior of the Objects.
Objects with evolutionary behavior exhibit properties that are subject to change over time. In remote sensing, most of the objects exhibit evolutionary behavior with age. The Darwinian transformation of species is also an example of evolutionary behavior (Ahsan and Shah, 2008). Evolutionary behavior is also exhibited in domains which are exploratory and research oriented. In these domains, evolution is triggered by new research and discoveries about the domain (Ahsan and Shah, 2008).

ix) Fuzzy Functional Characteristics
These may result from imprecise and incomplete information and data about the application domain. In remote sensing, imprecise data is a result of varying experimentation procedure and equipment used.

x) The Emergent Requirements
The action of developing, delivering and using the product itself create more requirements thorough improved understanding of the problem. This frequently occurs in scientific domains such as remote sensing. With increasing reliance on dry lab experimentation using automated tools, scientists discover new facts about the problem domain even faster. This makes the scientific domains such as remote sensing highly evolutionary (Ahsan and Shah, 2008).

2.3 Limitations of Existing Agile Techniques
In our opinion, existing agile SD techniques despite of being used for the development of computer applications (CA) relating to evolutionary domain, have some limitations because, they are not supported by a suitable framework/methodology consisting of the characteristics of the evolutionary domain (ED), e.g.

i) Explorative/iterative nature

ii) Difficulty in specifying functional requirements
iii) Emergent requirements

The above characteristics necessitate an iterative analysis process. Temporal aspect can be added to trace back the history of changes (Shah, 2008). To the best of our knowledge, the existing Agile SDM’s such as XP, Scrum, Crystal methodologies, FDD and the RUP do not have an iterative analysis process/mechanism to store temporal information of a system. Our proposed methodology, which we design and present in next two sections, have an iterative analysis process/mechanism for storing temporal information. Wirfs-Brock et al and Shah suggested modification to the Spiral model and the Waterfall model to make them suitable frameworks for the object-oriented development methodologies and the prototype-based development methodologies, respectively. The strictly static order working of the phases was replaced by iterative cycles and back cycles. An iterative cycle meant for incorporating additional knowledge into an already developed system resulted in the processing of both design and development phases. For this reason the two phases were merged into a single development phase. A back-cycle is represented as an incorporating process for the revision to an under-development system. Due to these modifications, the modified life cycle model was able to consider, acquire and incorporate both the meta-data knowledge (functional requirements) and data knowledge (data instances) of the system, which were acquired at the start or during the system development of evolutionary domains. Table 2.4 gives a detailed comparison between different software development frameworks and methodologies as discussed earlier.
### Table 2.4: Comparison between different SD frameworks/methodologies over some general parameters

<table>
<thead>
<tr>
<th>General Parameters</th>
<th>Waterfall (Linear)</th>
<th>Prototype (Iterative)</th>
<th>Incremental+ Spiral (Linear+ Iterative)</th>
<th>RAD (Iterative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SADT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yourdon (Structured)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fusion 96</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grady Booch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grady Booch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coad &amp; Yourdon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shlaer &amp; Mellor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scrum</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crystal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Overlap (iterations)        | Some Overlap       | Some Overlap          | No Overlap                           | No Overlap       |
| Project Division            | Allowed            | Allowed               | Allowed                               | Maximum Project Division |
| Documentation               | Compulsory         | Not needed            | Needed                               | Not needed       |
| Team Composition            | Unstable           | Stable                | Unstable                             | Stable          |
| Sequence Control            | Tight control      | Flexible              | Moderate                             | Moderate        |
| Progress                    | Measurable         | Not Measurable        | Measurable                           | Measurable      |
| Resources                   | Heavy              | Low                   | Heavy                                | Heavy           |
| Manageability               | High               | High                  | High                                 | High            |
| Requirement Specification   | Early              | Continuing            | Continuing, sometimes late           | Continuing      |
| Requirement Specification   | Unambiguous        | Vague                 | Vague                                | Vague           |
| Testing and Verification    | Late               | Immediate             | Late, sometimes continuous           | Continuous      |
| Flexibility                 | Inflexible         | Flexible              | Inflexible                           | Flexible        |
| Maintenance cost            | High               | Moderate              | High                                 | Moderate        |
| Communication Overheads     | High               | High                  | High                                 | High            |
| Applications                | Mainframe          | Transaction Online    | Event-driven Systems, Real Time or Safety Critical Systems | Interactive Business Applications |
| Project size                | Large              | Large                 | Large                                | Small-to-Medium |
| Project requirements        | Strict             | Flexible              | Strict                               | Flexible        |
| Domain experts/knowledge    | May be inexperienced| Experienced           | Experienced                          | Experienced     |

Table 2.5 provides a comparison between different software development methodologies over specification parameters.
Table 2.5: Comparison between different SDM over specification parameters

<table>
<thead>
<tr>
<th>Specification Parameters</th>
<th>JSD</th>
<th>SADT</th>
<th>YSM</th>
<th>Booch 91,94 &amp; Coad &amp; Yourdon</th>
<th>Fusion 96</th>
<th>UML</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Communication</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Behavior</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Functions</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conceptual Components/</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decomposition into Objects</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Functions/</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Operations</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Behavior/</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Behavior</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Communication/</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Communication</td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.6 compares different software development methodologies over component based parameters for analysis and design.

Table 2.6: Comparison of different software development methodologies over component based parameters (analysis and design)

<table>
<thead>
<tr>
<th>Component Based Parameters (Analysis)</th>
<th>DeMarco Structured Analysis</th>
<th>Martin Information Engineering (MIE)</th>
<th>YSM</th>
<th>Bailin Object-Oriented Requirements Specification</th>
<th>Coad &amp; Yourdon Object-Oriented Analysis</th>
<th>Shlaer and Mellor OOA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification/Classification of Entities</td>
<td>NS</td>
<td>Data-Model Diagram</td>
<td>ERDiagram</td>
<td>Class and Object Diagram Layer1</td>
<td>Information Structure Diagram (ISD)</td>
<td></td>
</tr>
<tr>
<td>General-to-specific and whole-to-part entity relationship</td>
<td>NS</td>
<td>Data-Model Diagram</td>
<td>ERDiagram</td>
<td>Class and Object Diagram Layer2</td>
<td>(ISD)</td>
<td></td>
</tr>
<tr>
<td>Other Relationship (creates , uses , etc.)</td>
<td>NS</td>
<td>Data-Model Diagram</td>
<td>ERDiagram</td>
<td>Class and Object Diagram Layer4</td>
<td>(ISD)</td>
<td></td>
</tr>
<tr>
<td>Attributes of Entities</td>
<td>Data</td>
<td>Bubble Chart</td>
<td>Data</td>
<td>NS</td>
<td>Class and Object Diagram</td>
<td>(ISD)</td>
</tr>
<tr>
<td>Table 2.7 gives a detailed analysis of different software development frameworks/methodologies, supporting remote sensing data characteristics.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 2.7: Frameworks and methodologies supporting remote sensing data

<table>
<thead>
<tr>
<th>Remote sensing Data characteristics</th>
<th>Methodology Factors</th>
<th>Methodology(s)</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed-Temporally and Spatially Categorized</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tight Control</td>
<td>SADT, JSD, Yourdon</td>
<td></td>
<td>Water Fall</td>
</tr>
<tr>
<td>Measurable Progress</td>
<td>SADT, JSD, Yourdon, Booch, Coad &amp; Yourdon, Shlaer &amp; Mellor</td>
<td></td>
<td>Water Fall, Incremental, RAD</td>
</tr>
<tr>
<td>Resources needs to be conserved</td>
<td>SADT, JSD, Yourdon, Coad &amp; Yourdon, Shlaer &amp; Mellor</td>
<td></td>
<td>Water Fall, Incremental, RAD, Spiral</td>
</tr>
<tr>
<td>Online Requiring Extensive User Dialog</td>
<td>Shlaer &amp; Mellor</td>
<td></td>
<td>Prototype, Incremental, Spiral</td>
</tr>
<tr>
<td>Future Scalability of Design is Critical</td>
<td>Booch, coad &amp; Yourdon</td>
<td></td>
<td>Water Fall, Incremental, Spiral</td>
</tr>
<tr>
<td>Risk Avoidance is at high Priority</td>
<td>Booch, Shlaer &amp; Mellor</td>
<td></td>
<td>Spiral</td>
</tr>
<tr>
<td>High Degree of Accuracy</td>
<td>Booch, Shlaer &amp; Mellor</td>
<td></td>
<td>Spiral</td>
</tr>
<tr>
<td>Might benefit from mix Methodologies</td>
<td>Booch, Shlaer &amp; Mellor</td>
<td></td>
<td>Spiral</td>
</tr>
<tr>
<td>Dramatic savings in Time</td>
<td>XP, Scrum, Crystal</td>
<td></td>
<td>RAD</td>
</tr>
<tr>
<td>Critical Response Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Active user Involvement</td>
<td>Fusion, XP</td>
<td></td>
<td>Prototype, RAD</td>
</tr>
<tr>
<td>Interactive</td>
<td>Yourdon, Coad&amp;Yourdon, Shlaer &amp; Mellor, Fusion</td>
<td></td>
<td>Prototype, Incremental, Spiral, RAD</td>
</tr>
<tr>
<td>Tight fit Between User Requirements and System Specifications</td>
<td>SADT, JSD, XP, Scrum</td>
<td></td>
<td>Water Fall, RAD</td>
</tr>
<tr>
<td>Dramatic Saving in Time</td>
<td>Fusion, XP, Crystal, Scrum</td>
<td></td>
<td>Prototype, RAD</td>
</tr>
<tr>
<td>Highly customized</td>
<td>Fusion, Booch, XP</td>
<td></td>
<td>Spiral, Prototype, RAD</td>
</tr>
<tr>
<td>High Degree of Accuracy</td>
<td>Booch, Shlaer &amp; Mellor</td>
<td></td>
<td>Spiral</td>
</tr>
<tr>
<td>Highly Skilled and experienced Manager required</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, XP,</td>
<td></td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Clear Objectives</td>
<td>SADT, JSD, Yourdon, Booch, Coad &amp; Yourdon</td>
<td></td>
<td>Water Fall, Spiral, RAD</td>
</tr>
<tr>
<td>Temporal in Nature: Volatile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flexible Control</td>
<td>Fusion, Booch, Coad &amp; Yourdon, Shlaer &amp; Mellor, XP, Scrum</td>
<td></td>
<td>Prototype, Spiral, RAD, Incremental</td>
</tr>
<tr>
<td>Requirements may Change Significantly</td>
<td>Fusion, Booch, Shlaer &amp; Mellor</td>
<td></td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Experienced Project Manager</td>
<td>Fusion, Booch, Shlaer &amp; Mellor</td>
<td></td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Risk Avoidance is a high Priority</td>
<td>Fusion, Booch</td>
<td></td>
<td>Spiral</td>
</tr>
<tr>
<td>A High Degree of Accuracy</td>
<td>Fusion, Booch</td>
<td></td>
<td>Spiral</td>
</tr>
<tr>
<td>Progress of System Development is Measurable</td>
<td>SADT, JSD, Yourdon, Booch, Coad &amp; Yourdon, Shlaer &amp; Mellor, XP</td>
<td></td>
<td>Water Fall, Incremental, RAD</td>
</tr>
<tr>
<td>Documentation Required</td>
<td>SADT, JSD, Yourdon, Booch, Coad &amp; Yourdon, Shlaer &amp; Mellor</td>
<td></td>
<td>Water Fall, Incremental, Spiral</td>
</tr>
<tr>
<td>Have a Validity Constraints</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project has Clear Objectives and Solutions</td>
<td>SADT, JSD, Yourdon, Booch, Coad &amp; Yourdon, Shlaer &amp; Mellor</td>
<td></td>
<td>Water Fall, Incremental, Spiral</td>
</tr>
<tr>
<td>Requirements are Stable</td>
<td>SADT, JSD, Yourdon, Booch, Coad &amp; Yourdon, Shlaer &amp; Mellor</td>
<td></td>
<td>Water Fall, Incremental</td>
</tr>
<tr>
<td>User is Involved Throughout the System Development</td>
<td>Fusion, XP</td>
<td></td>
<td>Prototype, RAD</td>
</tr>
<tr>
<td>Pressure Exists for Immediate Implementation of Something</td>
<td>Fusion, XP</td>
<td></td>
<td>Prototype, RAD</td>
</tr>
<tr>
<td>Low Project Risk</td>
<td>Fusion, Booch</td>
<td></td>
<td>Spiral</td>
</tr>
<tr>
<td>A high Degree of Accuracy</td>
<td>Booch, Shlaer &amp; Mellor</td>
<td></td>
<td>Spiral</td>
</tr>
<tr>
<td>Experienced Team Members</td>
<td>Fusion, Booch, Shlaer &amp; Mellor</td>
<td></td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Possible Project Division</td>
<td>SADT, Yourdon, JSD, Fusion, Shlaer &amp; Mellor, Booch, Coad &amp; Yourdon</td>
<td></td>
<td>Water Fall, Prototype, Incremental, Spiral</td>
</tr>
</tbody>
</table>

**Cont.**
<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Methodologies</th>
<th>Frameworks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Evolutionary</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation not needed</td>
<td>Fusion, Booch, XP, Scrum</td>
<td>Prototype RAD</td>
</tr>
<tr>
<td>Flexible Sequence Control</td>
<td>Fusion, Booch, Coad &amp; Yourdon, XP</td>
<td>Prototype, Spiral, RAD, Incremental</td>
</tr>
<tr>
<td>Requirements may Change Significantly</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, Scrum</td>
<td>Prototype, Spiral, RAD, Incremental</td>
</tr>
<tr>
<td>Experienced Team Members</td>
<td>Fusion, Booch, Shlaer &amp; Mellor</td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Encourage Innovation and Flexible Design</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, Coad &amp; Yourdon</td>
<td>Prototype, Incremental, RAD, Spiral</td>
</tr>
<tr>
<td>Project Objectives are Unclear</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, Scrum</td>
<td>Prototype, RAD</td>
</tr>
<tr>
<td>Functional Requirements may change Frequently</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, Scrum</td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Online System Requiring Extensive User Dialog</td>
<td>Yourdon, Coad &amp; Yourdon, Shlaer &amp; Mellor, Fusion</td>
<td>Prototype, Incremental, Spiral</td>
</tr>
<tr>
<td>Flexible Control</td>
<td>Fusion, Booch, Coad &amp; Yourdon, XP</td>
<td>Prototype, Spiral, RAD, Incremental</td>
</tr>
<tr>
<td>Project Manager is Highly Skilled</td>
<td>Fusion, Booch, Shlaer &amp; Mellor</td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Rapidly Change ability to the System Design as</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, Scrum</td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Demanded by users</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Uncertain, Imprecise</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unstable Team Composition is and predictable</td>
<td>SADT, JSD, Yourdon, Booch, Shlaer &amp; Mellor</td>
<td>Water Fall, Incremental</td>
</tr>
<tr>
<td>to Fluctuate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encourage Innovation and Flexible Design</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, Coad &amp; Yourdon</td>
<td>Prototype, Incremental, RAD</td>
</tr>
<tr>
<td>Project Decomposition</td>
<td>SADT, JSD, Yourdon, Booch, Shlaer &amp; Mellor, Fusion</td>
<td>Water Fall, Prototype, Incremental, Spiral</td>
</tr>
<tr>
<td>Flexible Control</td>
<td>Fusion, Booch, Coad &amp; Yourdon, XP</td>
<td>Prototype, Spiral, RAD, Incremental</td>
</tr>
<tr>
<td>Mixture of other Development Methodologies</td>
<td>Booch, Shlaer &amp; Mellor</td>
<td>Spiral</td>
</tr>
<tr>
<td>Requirements of the System are Unknown or</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, Scrum</td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Uncertain</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Heterogeneous</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Division</td>
<td>SADT, JSD, Yourdon, Booch, Shlaer &amp; Mellor, Fusion</td>
<td>Water Fall, Prototype, Incremental, Spiral</td>
</tr>
<tr>
<td>Measurable Progress</td>
<td>SADT, JSD, Yourdon, Booch, Shlaer &amp; Mellor, Coad &amp; Yourdon</td>
<td>Water Fall, Incremental, RAD</td>
</tr>
<tr>
<td>Difficult to Respond Changes</td>
<td>SADT, JSD, Yourdon, Booch, Shlaer &amp; Mellor, Coad &amp; Yourdon</td>
<td>Water Fall, Spiral</td>
</tr>
<tr>
<td>Project is Large, Expensive and Complicated</td>
<td>SADT, JSD, Yourdon, Booch, Shlaer &amp; Mellor, Coad &amp; Yourdon</td>
<td>Water Fall, Prototype, Incremental, Spiral</td>
</tr>
<tr>
<td>Mainframe or Transaction Oriented Batch Systems</td>
<td>SADT, Yourdon, JSD</td>
<td>Water Fall</td>
</tr>
<tr>
<td>Avoid to Solve Wrong Problems</td>
<td>SADT, JSD, Yourdon, Booch, Shlaer &amp; Mellor, Coad &amp; Yourdon</td>
<td>Water Fall, Incremental, Spiral</td>
</tr>
<tr>
<td>Technical Requirements (e.g. Response Time,</td>
<td>XP, Scrum, Crystal</td>
<td>RAD</td>
</tr>
<tr>
<td>Throughput) are Reasonable.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continual Evolution of the Project Requirements</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, Scrum</td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Reduce Project Risk by Breaking Project into</td>
<td>Booch, Shlaer &amp; Mellor</td>
<td>Spiral</td>
</tr>
<tr>
<td>Smaller Segments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encourage Innovations and Flexible Changes</td>
<td>Fusion, Booch, Shlaer &amp; Mellor, Coad &amp; Yourdon</td>
<td>Prototype, Incremental, RAD</td>
</tr>
<tr>
<td>Team Members are Fully Experienced</td>
<td>Fusion, Booch, Shlaer &amp; Mellor</td>
<td>Prototype, Spiral, RAD</td>
</tr>
<tr>
<td>Very Large Infrastructure Projects</td>
<td>SADT, JSD, Yourdon, Booch, Shlaer &amp; Mellor, Coad &amp; Yourdon</td>
<td>Water Fall, Prototype, Incremental, Spiral</td>
</tr>
</tbody>
</table>

31
From the above Table 2.7, there are some frameworks and methodologies which support evolutionary type of data. But no single framework or methodology support the majority of these types of data, so there is a need for a methodology which enables to capture the evolutionary data types. Shah proposed a methodology for bioinformatics which belongs to a scientific and evolutionary domain.

Table 2.8 gives a comparison between the bioinformatics data characteristics and of remote sensing.
Table 2.8: Difference between biological and remote sensing data characteristics

<table>
<thead>
<tr>
<th>Factors/Criteria</th>
<th>Biological Data</th>
<th>Remote Sensing Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volatility of Data</td>
<td>Volatility Results from new findings and environment</td>
<td>Volatility results from need and application</td>
</tr>
<tr>
<td>Source of Change</td>
<td>Evolution is Research based (experiments in the Wet-lab.)</td>
<td>Evolution is Observation and Need based</td>
</tr>
<tr>
<td>Source of Uncertainty</td>
<td>Uncertainty results from different experimental Procedures</td>
<td>Uncertainty results from equipment used</td>
</tr>
<tr>
<td>Interaction with Society</td>
<td>Need not be integrated (indirectly) with society perspectives</td>
<td>Need to be integrated (directly) with society perspectives</td>
</tr>
<tr>
<td>Data Generation Source</td>
<td>Primary Data generation sources are centralized</td>
<td>Primary data sources are Distributed</td>
</tr>
<tr>
<td>Location of data Generation</td>
<td>It is generated in Lab. environment</td>
<td>It is generated through Geospatially (out of the Lab.)</td>
</tr>
<tr>
<td>Decision/Use</td>
<td>It is not used for real time decisions</td>
<td>It is used for real time decisions</td>
</tr>
<tr>
<td>Cause of Imprecision</td>
<td>Imprecision results from faulty or non-standardize Wet-lab. equipment</td>
<td>Imprecision results from faulty or non-standardize Wet-lab. Equipment in addition to Environmental /Weather conditions</td>
</tr>
<tr>
<td>Granularity</td>
<td>Extreme level of Granularity</td>
<td>Moderate level of granularity</td>
</tr>
<tr>
<td>Observations</td>
<td>New observations are unprecedented</td>
<td>New Observations are precedential up-to some extent</td>
</tr>
<tr>
<td>Interval of change</td>
<td>Discrete</td>
<td>Continuous</td>
</tr>
<tr>
<td>Interval of Data Generation</td>
<td>Discrete</td>
<td>Continuous</td>
</tr>
</tbody>
</table>

From the above comparison Table 2.8, it is suggested that there is a need for domain specific methodology for remote sensing applications too like bioinformatics as proposed by Shah and Ahsan (Ahsan and Shah, 2008).
Chapter 3

The Proposed Methodology

Mostly software development (SD) is a messy activity, often renowned by a phrase “code and fix”. SD is carried out without an underlying plan and design of system is done with short term decisions. This situation seems to be well suited when the system is small enough, but for increasingly large systems it is not easy to add-up new features to a system. Furthermore, bugs become increasingly prevalent and increasingly difficult to fix. Hence, a software development methodology is needed to structure, plan and control the process of developing an information system. These methodologies force a disciplined process upon SD with the objective of making SD more expected and more resourceful. This can be done by developing a detailed process with a strong stress on planning like many other engineering disciplines. In this chapter, we propose a methodology for remote sensing applications and argue that there is a need to have a specific methodology for such applications. We propose a remote sensing applications (RSA) methodology to develop models at abstraction level like a code. The benefit of having such methodology is that it evidently defines every step to be taken, forcing the developers to pursue the defined methodology in a customary way. It specifies how to develop the models in a sequence, and how to derive a model from another one at the abstraction level immediately above it. The methodology ensures that developers will know at every moment during the development life cycle what is to be done next for achieving their goals.
3.1 A Domain Specific Methodology for RSAs

The complexity that arises to prevent software developers from reusing software is to be short of software artifacts that are difficult to correlate. Domain-specific software architecture has been proposed. A domain-specific application provides not only a methodology for reusable software components to fit into but also captures the design rational and provide for a level of compliance. The fundamental promises of this research work are:

i) An application that fulfills the needs of user requirements can be defined.

ii) User requirements/needs can be translated into a set of requirements.

iii) Requirements can be assembled in different ways.

iv) Implementation constraints which limit the fulfillment can be addressed.

Our proposed methodology is inspired by the previous work done by Shah "A Framework for the Prototype-based software development Methodologies" and “Proposed Object-Oriented Development Methodology (OODM)”. This framework and methodology provide the foundation on which we have based our proposed methodology.

We decompose our proposed methodology in two steps:

i) When there is no image in the GIS data store i.e. at time \( t_0 \) (at scratch).

ii) The development of the system after time instance \([t_0]\) and change detection.

In the Section 3.5, we discuss the methodology when the system is to be developed at time \( t_0 \) and in Section 3.6, we discuss in detail the methodology after time instance \([t_0]\) and change detection.

Another major contribution of our research work is the introduction of Pre-Analysis Phase in software development methodology cycle. In our opinion, as conventional methodologies for open-ended system modeling/ development are generic therefore Universe of Discourse (UoD) needs to be well defined and delimited. However, in application areas such as Remote Sensing,
most of types of UoD are already known and it makes little sense to input entire Default Knowledge (DK) in Analysis Phase. For this reason we are introducing a Pre-Analysis Phase which reuses well-known, stable and concrete artifacts of the domain. Detailed working and functionality of Pre-Analysis Phase is described in Section 3.6.3.

Our proposed methodology is triggered through inputs which are categorized into two types:

1) Inputs when a system is to be developed at time instance $t_0$

When a system is to be developed at time instance $t_0$, problem statement and DK are input to the Analysis Phase of our proposed methodology, described in the next section.

2) Inputs that may occur during system development after time instance $t_0$

The second category of inputs deals with the changes that may occur during system development. These changes may be [Type I, Type II, Type III and Type IV], depending upon the extent/degree of change.

We briefly introduce these changes here with detailed descriptions, in Section 3.6.1.

Type I: This change occurs when there is a change in parameter, data of the prototype. This type I change can simply be handled/incorporated without affecting other parameters, and to handle/incorporate it we do not need exhaustive development endeavor.

Type II: This change occurs in functional requirements (FR) of a system and will be handled by the analysis phase.

Type III: If structure change occurs in an artifact or layer change occurs then this change will be handled by the pre-analysis phase.

Type IV: If a change occurs due to combination of all of the above three types, then it can be handled by pre-analysis phase for structural change and development phase for data change of our proposed methodology. Type I and Type II changes are handled by the implementation phase.
and analysis phase respectively and will be explained when we describe the detailed working of these two phases. Type III changes will be explained in Section 3.6.3, when we describe pre-analysis phase.

The block (processes) diagram shown in Figure 3.1, gives the whole picture of how new change or raw data/information is to be processed. The process which shows ‘Satellite data acquisition and processing’ receives as an input, the raw data and after processing it completely (described in Section 3.3) produces cleaned data/information to the next process ‘identification processing’ which will also be discussed in detail in Section 3.4). It produces and identifies the four types of changes. These changes can be incorporated by our proposed methodology as shown in Figure 3.19.

The detailed Figures of process (a) and process (b) are shown in Figure 3.2 and Figure 3.12 respectively.

**Figure 3.1:** Block diagram of methodology for GIS application development to capture/identification in data/information change

Before giving a detailed description of our methodology, it is important that we describe the Remote Sensing Method and the processes, techniques and procedures used in it for a thorough understanding of our methodology. These processes, techniques and procedures form the basis of some fundamental methods proposed by us such as Similarity Checking, Metadata Extraction and Layer Determination.
3.2 Remote Sensing Method

The interfacing of GIS and satellite RS provides new and current capabilities to analyze the land-use changes dynamics (Eiumnoh, 2000). The unique feature of satellite RS is comparable to the data collecting tools used for baseline record and prospect monitoring process (William, 2005).

3.2.1 Remote Sensing

RS provides an instrument-based technique in the possession and measurement of spatially controlled (geographically distributed) data/information on some property(ies) (spectral, spatial, physical) (Radhika, et al., 2010).

3.2.2 Remote Sensing Satellites

A satellite with remote sensors to observe the earth is called a remote-sensing satellite, or earth observation satellite. Remote-Sensing Satellites are characterized by their altitude, orbit and sensor (Mahmood, et al., 2007).

TRIOS (1960-1965) Series

Television/Infrared_Observation_Satellites.

NOAA being the first generation of National_Oceanic and Atmospheric_Administration satellites and it was the first operational RS system (Philippe, et al., 2007).

GMS, Geo_Synchronous_Meteorological_Satellite having an altitude of 36,000 km, and meteorological observations is its main purpose.

Landsat have an altitude of 700 km [polar orbit] and land area surveillance is it main purpose.

Other RS satellite series which are in operation are: SPOT, MOS, JERS, ESR, RADARSAT, IRS etc (Sivakumar, et al., 2003).

Now we give the details of processes (process (a) and process (b)) as shown in Figure 3.1.
Figure 3.2 gives the detailed picture of how data is captured through satellite and needs certain processes to be performed to make it suitable for GIS applications development. These processes include: Radiometric Pre-processing Calibration, Geo-Referencing, Image Enhancement, Image Classification and Image Interpretation etc.
Figure 3.2: Flow chart showing the complete research method for GIS application development for process (a) shown in Figure 3.1
3.3 Satellite Data Acquisition and processing (process (a))

We give detail of process (a) in Figure 3.2. This process gets the inputs from the satellite in the form of a ground image or raw data/information stored as a raster file format. This data/information needs certain pre-processing steps to make this data useable for many remote sensing applications.

3.3.1 Satellite Imagery

Satellite images applications are in the field of agriculture, geology, forestry, biodiversity conservation, regional planning, education, intelligence and warfare.

The basic data structures for storing and manipulating computerized images and graphical data are Raster and Vector (Shi, et al., 2006).

3.3.1.1 Raster Images (RI)

Raster images are in the form of individual pixels. A pixel value is associated with each spatial position/resolution, which is called the coordinate, elevation, and any pertinent data, such as color, ID Number (John, et al., 2006).

The ArcGIS image server adds Raster data to image service definition by specifying Raster type. The Raster type helps the server identify Metadata, such as Geo-Referencing, acquisition data or sensor type, along with raster format (McGarava, et al., 2009).

A Raster format defines how pixels are stored such as number of Rows and Columns, number of Bands, actual pixel values and other Raster format specifications. However, by adding Raster data according to a Raster type, the ArcGIS image server reads the appropriate metadata and uses it to define any processing that has to be applied (Nafaa et al., 2009).
3.3.1.2 Vector data

Points and lines are the form of Vector data that are geometrically/mathematically associated. The coordinates are being used for storing point, and the lines stored in the form of series of point_pairs. Vector file formats are: DXF, DWG, ASV (ArcGIS 9.2 Help Desk).

3.3.2 Processing

Several processing procedures can be adopted for the use of digital satellite/image data in spatial database. These procedures may be:

Radiometric correction, Geometric correction, Terrain correction, Image enhancement and feature selection. The aim of digital image processing (DIP) is to boost both the interpretability and accuracy of the digital data during the Image Procession (IP) phase given in Figure 3.2.

3.3.3 Radiometric Correction (RC) and Calibration

RC (referred to as ‘pre-processing and restoration’) is used to alter Digital Number(DN) values in order to report for noise, i.e., assistance to the DN that are not a function of the sensed attribute but are of:

i) The intervening_atmosphere

ii) The sun_sensor_geometry or

iii) The sensor itself

In case, if the majority of the scene is exaggerated, we might ignore this result. We have to correct otherwise.

Land use/land cover changes can be monitored by using satellite images. But the key problems with images below clouds are not covered by sensor. The distortion in the images due to cloud_cover is the classical problem of classical band of RS imagery, especially for non_stationary satellites which are commonly found in the earth resource observation
application. Removing cloud_cover for satellite imagery is very useful for assisting image_interpretation (Yang et al., 2000). Hence cloud detection and removal is very vital in processing of satellite imagery. Further, it is quite hard to compute and interpret changes on multi_temporal images under different elucidation, atmospheric or sensor conditions without RC. The relative approach to RC is known as relative_radiometric_normalization (Yang et al., 2000). Many methods for radiometric correction and calibration are available in the literature.

Image_normalization given in literature can be classified into 3 categories: statistical methods (i.e. Standard_Deviation_Method); the Histogram_Matching_Methods; the Linear_Regression_Methods.

For Normalization Methods, a target that meets the following criteria is selected as an ideal target for normalization.

- The targets should be approximately at the same elevation so that the thickness of the atmosphere over each target is approximately the same.
- When viewed on the image display screen, the pattern on the normalization targets should not change over time.
- A set of targets must have a wide range of grey values for the regression model to be reliable.

Image normalization can be divided into two steps:

i) Selecting normalization targets and

ii) Determining normalized coefficients

The methods reported in the literature are:

- Haze_Correction (HC) normalization
- Minimum_Maximum (MM) normalization
• Mean Standard Deviation (MS) Normalization
• Simple Regression Normalization (SR)
• Dark Set Bright (DB) normalization
• Pseudo Invariant (PI) normalization
• No Change Regression (NC) normalization

(Wong et al., 2002).

3.3.3.1 Cloud Removal

It is normally hard to get a cloud free image of even a small-sized land area. With these imperfect cloud contaminated image sources, we may still interpret images by referring to data from same geographic area where image data of the core source is tainted by the cloud cover. Bright areas generally are assumed as clouds (Ho, 2002). Detection of cloud is achieved by Average Brightness Shareholding (ABT) algorithm. It is based on three observations (Ho, 2002). Radiometric correction Process can be shown in Figure 3.2.

3.3.4 Geo-Referencing

It is the process (shown in Figure 3.2) to Scale, Rotate, Translate and Deskew the image for matching it with certain position and size.

It involves the reorientation of the image data to selected parameters. This consists of selection of a Map Projection (MP) system and co-registration of satellite image data with other data layers that may be used in a GIS (Spies et al., 2007). This will permit for precise spatial assessments/measurements of the data generated from the satellite imagery.

3.3.5 Image Enhancement (IE)

Image enhancement technique can be used in order to help Visual interpretation, Visual appearance of the objects can be improved by Grey level stretching to get better contrast
and Spatial_filtering for enhancing the edges (Liew, 2001). Example given in Figure 3.3 represents an enhancement procedure.

**Figure 3.3:** Satellite image without image enhancement

Radiometric (RC) and Geometric Corrections (GC) have been made. The transformation has been done to conform up to certain MP. This image is represented without any more enhancements (Liew, 2001).

The image given in Figure 3.3 is an un-enhanced image with bluish_tint in all images which produces a hazy appearance. This is due to sunlight scattered by the atmosphere into the field_of_view of the sensor. (Liew, 2001).

It is valuable to observe the image _Histograms_ before performing any IE. The x-axis of the histogram is the array of the available numbers, [i.e. 0_ 255]. The y-axis represents number of pixels in the image with given DN (Liew, 2001). The histograms 3_bands are shown in Figure 3.4, 3.5 and 3.6.
Figure 3.4: X_S3 Histogram (red in color)(near infrared) band

Figure 3.5: X_S2 histogram(green) band
Figure 3.6: X_S1 histogram (green band represented in blue)

Note: The minimum DN for any band is not zero. Any histogram can be shifted to the right will be by a certain number (Liew, 2001).

The Grey-Level _ Transformation (GLT) is shown through graph in Figure 3.7 (Liew, 2001).
Figure 3.7: Grey_LEVEL_Transformation to perform linear grey_level stretching of the 3_bands of the image. RedLine: X_S3 band; Green_Line: X_S2 band; Blue_Line: X_S1 band.

The resultant after applying the linear stretch is shown in Figure 3.8. The hazy_appearance has been removed, except for some parts near to the top of the image. (Liew, 2001).
**Figure 3.8:** Multi_spectral SPOT image after enhancement by a simple linear_grey_level stretching

### 3.3.6 Image Classification (IC)

Different landcovers in the image can be discriminated by applying some IC algorithms using spectral_features, e.g., the brightness and color information enclosed in each pixel (John, et al., 2000). The classification_procedures may be supervised or un-supervised.

In supervised_classification, the Spectral_Features (SF) of some areas of known landcover types are extracted from an image, known as training area. Every image pixel is then classified as the one which belongs to one of the classes depending on how close its SFs are to the SFs of the training areas as shown in Figure 3.9 (John, et al., 2000). In unsupervised classification, the computer program automatically groups the pixels in the image into separate clusters, depending on their spectral features shown in Figure 3.10. Each cluster will then be assigned a land cover type by the Analyst (Jaffery, 2009).
Figure 3.9: SPOT multi_spectral image belongs to the test area

Figure 3.10: Thematic_map resulting from the SPOT satellite image using an unsupervised_classification_algorithm.

A reasonable assignment of land-cover types to the thematic_classes is shown in the Figure 3.11. The accuracy of the thematic_map derivative from RS images should be verified by field_observation (Jaffery, 2009).
The SF of these Land-cover classes can be exhibited in the graphs shown in Figure 3.11. The 1st graph shows the mean pixel values of the \texttt{x_s3} (near infrared) band versus the \texttt{x_s2} (red) band for each class. The 2nd is a plot of the mean pixel values of the \texttt{x_s2} (red) versus \texttt{x_s1} bands (Jaffery, 2009). The SD of the pixel values for each class is also shown in Figure 3.11.

![Scatter Plot showing mean pixel values for every land-cover class](image)

**Figure 3.11:** Scatter Plot showing mean pixel values for every land-cover class

### 3.4 Identification Process

This process is shown in Figure 3.1 (process (b)) as “identification process”. It gets a cleaned data/information as an input and outputs the types of changes that may occur in the input data. Figure 3.12 gives a detailed picture of how cleaned data/information from the above mentioned processes is to be used for change detection in input data. To capture and handle these types of changes, we have proposed a methodology shown in Figure 3.19.
3.4.1 Image Interpretation and Characterization

It refers to the labeling of image into one of a number of predefined categories.

Image interpretation may be defined as: qualitative and quantitative extraction of information in a map form related to its shape, structure, location, function, condition, quality and relationship between objects by using human knowledge/experience. Image interpretation in satellite RS can be made using a scene of a satellite image.

Interpretation Elements (IE): Eight IE are frequently used:
(i) Size  (ii) Shape  (iii) Shadow  (iv) Tone  (v) Color  (vi) Texture  (vii) Pattern  (viii) Color

Associated Relationships

At this stage we have identified pixel color values for different objects using object classification/categorization techniques as discussed above. By using these pixel values we can identify the area covered by that object as shown in Figure 3.13.

Now we can identify and incorporate the four types of changes that may occur during the remote sensing software development and maintenance.

Type I: Change occurs to the data

Type II: Change occurs to the functional requirements of an application

Type III: Change occurs to structure of an application

Type IV: Change is a combination of above three types

Referring to Figure 3.2; after image enhancement process, the image is classified and verified with the ground truth values. Any change in the image data can be detected with reference to above four mentioned types, where as the change in data and structure can be captured and handled by the similarity check method while functional requirements may be handled manually with the help of user as shown the Figure 3.2.
If there is change in number of pixel colors then there may be a data change. If there is an addition of new color to an object it means new layer is being added. Then there is a structure change. If there is a change in number of pixel colors as well as an addition of new color to the object then there is a combination of both type of changes and it can be decided with the help of similarity check process. The functional requirements may be implicit/internal or explicit/external which can cause both data and structural change. These functional requirements are from the users/stake holders and have to be incorporated during the system development or after the system development as shown in proposed methodology in Figure 3.14.

The interpreted image is to be checked for similarity (algorithm for similarly check is given in Section 3.6.2) with the GIS database. If they are 100% similar then it can be stored back to the GIS database with the addition of new dates otherwise it goes to change detection process for detection of the type change. So the type of change can be investigated with the help of change in color or addition of new color or both as described above and can be handled by the proposed methodology as shown in Figure 3.14.

The model which is being used for change detection can be useful for urban growth monitoring. It takes into account the process to convert vacant landscape into a wide-ranging living area. The updated GIS database can be used to update land cover map, land use map, digital map, deforestation, urban area and urban rural fringe, etc.

A satellite image represents a picture of the earth in digital form and is collection of a pixel’s grid. A pixel, which is a square in shape, can be considered as the smallest unit. The pixel brightness shows the reflected energy that can be detected by the satellite sensor over the area of land-covered by the pixel. The values of the pixels in the image are gray shades and darker shades are assigned lower pixel values of gray. Total shades are 256 of gray in these images.
To handle the above mentioned changes in the data/information during and after application development, we proposed a methodology for remote sensing application development as shown in Figure 3.14.

### 3.4.2 Data Dissemination Issues

Remotely sensed data is either constantly acquired or may be acquired on the request for specific purposes. In both the cases, it is organized in some host primary machine using a conventional DBMS (Ahmad and Shah, 2009).

However, fetching fresh data directly from primary site is not efficient. This is because datasets of remotely sensed data may be organized into multiple files containing a subset of data elements categorized temporally and spatially. The response time of such data bases may be adversely affected as multiple queries access these datasets for the same temporal or spatial information (Ahmad and Shah, 2009). The effect of write operations may even be more adverse as consistency is compromised resulting in conflicting results. Remotely sensed data can be categorized into temporal data and non-temporal data. Real time sensor data representing the physical world is temporal in nature (Ahmad and Shah, 2009). Temporal data objects are difficult to manage and analyze as they have validity intervals and are updated periodically. As compared to temporal data, non-temporal data do not change dynamically with time. Thus they do not have validity intervals and they do not need to be updated by periodic system updates. However, as most of these data and applications are hosted in a distributed environment, the issues of data dissemination relating, availability, consistency and response time affect non-temporal data also (Ahmad and Shah, 2009).
3.4.3 Data Provenance Issues

With the increased capability and usage of the Internet, various research groups with common interests have moved towards collaborative research (Ahmad and Shah, 2010).

Due to heterogeneity and distribution of data resources, the usability of data resource for a particular domain depends upon the provenance information attached to the data resource. More generally, Data Provenance is the information about data, describing its origin and sequence of tasks (workflows/processes) that were responsible for its transformation; structurally, logically, physically and/or geographically. It provides a qualitative and quantitative metric to analyze the quality and dependability of the data based on consumer trust of the source of creation and the sources that were responsible for modification. Therefore, data provenance has a significant role to play for remote sensing domain in addressing the concerns of trust, quality and copyright (Ahmad and Shah, 2010).

In Geographical Information Systems (GIS), Provenance information includes description of the lineage of the data product including description of the data source, the transformations used to derive it, references to the control information and mathematical transformations of the coordinates used (Ahmad and Shah, 2010). Spatial Data Transfer Standard (STDS) is one such lineage recording system that helps in using the data product by deciding whether the data meets the requirements of the domain or not. Lineage Information Program (LIP) is for GIS and is used for informational purposes, update stale data, regenerate and compare data (Ahmad and Shah, 2010), LIP follows a data-oriented provenance technique. Provenance data relates to spatial layers, the basic dataset of GIS, the transformation algorithm and intended use of the data (Ahmad and Shah, 2010). Semantic Information is not included. To be useful, the data to be transferred must also be meaningful in terms of data content and data quality We feel that if this
lineage information is stored and recorded in the machine readable from using Resource Description Framework (RDF) and Ontologies, an efficient automated validation of results is possible with the potential for increased access to and sharing of spatial data, the reduction of information loss in data exchange, the elimination of the duplication of data acquisition, and the increase in the quality and integrity of spatial data. Annotating data from one organization with enough metadata will enable other organizations to use the purposefully and in more meaningful ways (Ahmad and Shah, 2010). However, there are only a few data provenance systems catering for remote sensing domain which primarily focus data provenance as an issue (Ahmad and Shah, 2010). We have discussed the potential of replication and provenance techniques to solve the problems of remote sensing data dissemination and management in (Ahmad and Shah, 2010).

3.4.4 Potential of Replication and Provenance Techniques

E-Systems are real-time distributed applications and they need real-time data accessibility for prompt and timely decision making. Providing such real-time data accessibility is, however, a challenging task is due to long remote data accessing delays, network hops and inflexible temporal requirements of real-time transactions. Replication is a technique that has been used to ensure the availability of data in a distributed environment to improve the performance and meet the stringent response time requirements of real-time applications (Ahmad and Shah, 2010). As discussed earlier, a real-time data intensive application should be supported by provenance data to establish its authenticity, trust and reliability (Ahmad and Shah, 2010). More importantly, the provenance data can also be used to avoid duplication of the data replication efforts. We feel that real-time data intensive remote sensing applications can get benefit from the existing replication and provenance techniques. Based on this argument, we advocate the need of applying replication and provenance techniques in remote sensing applications. In our opinion,
applying the replication and provenance techniques, we can address the issues of dissemination of RS data, and achieve the following objectives: i) Data availability enhancement of data during disk-failure in a multi-disk system,  
ii) Speed up I/O performance of read-intensive applications, 
iii) Maintain consistency of temporal data,  
iv) Maintain reliability of transactions accessing temporal data, 
v) Efficient execution of batch-shared I/O oriented data-intensive tasks in web based distributed environment.

RS data is usually “write-once” and “read-many” type of the data and the read performance of an application becomes an important factor (Ahmad and Shah, 2010). This type of stored data is generally is accessed through range type of queries which retrieve a range of key values from a distributed high dimensional data set (Ahmad and Shah, 2010). To minimize the disk retrieval time for range queries various non-replication based schemes have been used that distribute data blocks among parallel disks (Ahmad and Shah, 2010). However, to achieve objectives i) and ii) using the data replication technique, two problems need to be addressed (Ahmad and Shah, 2010). These two problems are listed below.

a) The data placement problem of determining the optimal scheme for storing copies of each data block on a particular disk, and  
b) The scheduling problem, i.e., to minimize the retrieval time from the disk for a data block.
3.5 System development at time instance $t_0$

First we discuss our proposed methodology when the system is to be developed at time instance $t_0$. Our methodology works in two phases; Analysis phase and Design phase.

3.5.1 Analysis Phase

In the analysis phase, the problem statement of remote sensing applications is studied. In the initial steps, layers are identified and all the information about the metadata is extracted. This is achieved by the two sub phases of our proposed methodology which are:

a) Metadata Extraction/Annotation

b) Layer Determination
Figure 3.14: Methodology for remote sensing applications development (when the system is to be developed at time instance $t_0$)

Legends used: LP $\rightarrow$ Layer Prototype, DK $\rightarrow$ Default Knowledge, FR $\rightarrow$ Functional Requirements.
When a system is to be developed at time instance $t_0$, DK and problem statement is input to the analysis phase of our proposed methodology. The function of analysis phase can be decomposed into the processing units shown in Figure 3.15.

![Figure 3.15: The processing unit of analysis phase](image)

### 3.5.1.1 Metadata Extraction/Annotation

Different types of layers, for example, vegetation layer, urban area layer and soil layer represent different colors in an image. Vegetation layer is represented by green color, urban layer by a pink color and soil as a cyan color in a digital thematic satellite image. From the quantitative analysis of these images we can also find out the area covered by these types of layers with the help of quantitative analysis of the number of pixels present in each image. Vegetation layer can further be classified at granularity level shown in Table 3.1.
Table 3.1: Layer identification for vegetation layers

<table>
<thead>
<tr>
<th>Green</th>
<th>Parrot green</th>
<th>Prussian green</th>
<th>Plant green</th>
<th>Yellowish green</th>
<th>Dull grassy green</th>
</tr>
</thead>
<tbody>
<tr>
<td>New vegetation</td>
<td></td>
<td>A complete pattern of trees</td>
<td>Fields</td>
<td>Vegetation reduced</td>
<td>Herbs, shrubs and bushes</td>
</tr>
</tbody>
</table>

If there is a problem in determining some layer as described above, metadata is to be annotated with the help of stakeholders/users of the system. Hence, proper layer determination is carried out for further design and implementation.

Metadata is stored in a document containing quality, type, content, creation and spatial information about a set of data. It can be stored in any format such as a text file, Extensible Markup Language (XML), or database record.

It may be broadly classified into logical and machine-interpretable representations. Metadata is primarily used as a mechanism for communication between disparate components or systems. It is also used for representation at different level of granularity and perspectives.

There are three levels of metadata:

i) Discovery metadata

ii) Exploration metadata

iii) Exploitation metadata.

i) **Discovery Metadata (DM)** provides the minimum amount of information that is needed to convey the inquirer about the nature and content of the data source. It answers about “what, why, when, who, where and how” about spatial data.
What represents the title and explanation of the data set.

Why abstract detailing reasons to collect and its use.

When the data set was created? And the cycle to update.

Who originator, data supplier, and possible intended audience.

Where the geographic extent based on latitude/longitude, coordinates, geographical names or administrative areas.

How how was it built to access the data.

ii) **Exploration metadata** gives adequate information which facilitates an inquirer to discover that data is fit for a given purpose, to assess its properties, and to reference some point of contact for more information.

iii) **Exploitation Metadata** is to have those properties which are necessary to access, load, transfer, interpret, and apply the data in the end application where it is exploited. These roles form a continuum in which a user cascades through a pyramid of choices to determine what data are available, to evaluate the fitness of the data for use, to access the data, and to transfer and process the data. The exact order in which data elements are evaluated, and the relative importance of data elements, will not be the same for all users.

Data and information resources are generated by diverse user communities and the metadata required to describe these resources are equally diverse.

**3.5.1.1.1 Building Metadata Model**

This model is a combination of the following processes:

i) Identification information

ii) Metadata reference information

iii) Spatial reference information
iv) **Entity_and_attribute_information.**

It contains the vital information about the data set. The detailed diagram of the building metadata model is shown in the Figure 3.16.

![Diagram of Building Metadata Model](image)

**Figure 3.16** Decomposition of processing unit” Building Metadata Model”
3.5.1.2 Layer Determination Model

This model tries to identify the types of layers present in an image. This can be done with the help of the following eight types of image interpretation elements: Size, shape, shadow, tone, color, texture, pattern, and associated relationship or context.

Figure 3.17 shows the process of Layer determination with the help of image interpretation elements which are described above. When the layers are successfully determined, they are passed onto the design phase.

**Figure 3.17:** Decomposition of sub-phase “Building Layer Model”
3.5.1.3 Analysis Report

The analysis report thus contains different types of identified layers, for example, vegetation layer, water body layer, buildings layer etc. These layers are then passed onto the design phase for subsequent designing of layers so that implementation can be done in a proper way. The analysis report also depends upon the functional requirements of the user. User is an important part throughout the process of analysis.

3.5.2 Design Phase

The processing of design phase of our proposed methodology mainly uses the report generated by the analysis phase. The analysis phase works as three processing units; namely, sub-layers decomposition, data quality information model and spatial data organization model, as shown in Figure 3.18.

Figure 3.18: Output of the design phase (design report)
Water body layer can be classified into sub-layers as: lakes, ponds, rivers, streams and drainage.

The design phase of our proposed methodology gets its input from the analysis phase as layer prototypes (LP) for example; vegetation layer, water body layer, land cover layer etc. These layers are designated as $L_1P$, $L_2P$, ..., $L_nP$.

Now we give the details of the sub-processes of the design phase as described in Figure 3.18.

This phase designs the type of layers (vegetation layer, buildings layer, water body layer, etc.) to be input to the image incorporation and GIS storage phase. Vegetation layer can be further classified into sub-layers as: tree stands, individual landmark trees and large hedgerows, etc.

Structure layer can be classified into sub-layers as: labs, offices, school and libraries, housing, warehouses, public buildings (fires/police stations, government offices), etc. Water body layer can be classified into sub-layers as: lakes, ponds, rivers, streams, drainage, etc. Similarly, transportation layer can be further classified as: paved roads, unpaved roads and trails, bridges, railroads, runway/helipads, parking lots paved and unpaved, roads centerlines, etc. Now we will further discuss the sub-processes generated by the design phase.

**Sub-Layers Decomposition Model**

This process further decomposes the determined layers into subsequent sub-layers having the similar class, as described below.

Structure layer can be classified into sub-layers as: Labs, offices, school and libraries, housing, warehouses, public buildings (fires/police stations, government offices) etc.

Water body layer can be classified into sub-layers as: lakes, ponds, rivers, streams, drainage channels and culverts.

Environmental layer can be classified into sub-layers as: wetlands and marshes, sensitive habitat, wells, remediation/removal area, landfills, water table contours, etc.
Utilities layer is classified into sub-layers as: water, sewer, electricity, telephone, cable, natural gas, steam, etc. There may be other layers or sub-layers as: golf courses, recreational fields, cemeteries, power stations, incinerators and sewerage treatment plants.

The layers as described above are to be designed at this phase and these layers can migrate from one level to another depending upon the changes that can occur in the functional requirements by the user at the analysis phase.

3.5.2.1 Design Report

The design report consists of number of layer as described above and the sub-layers under each category of the layers.

These layers and sub-layers are further passed to the implementation phase.

3.5.3 Image incorporation and GIS Storage Phase

Designed layers are to be input to the image incorporation and GIS database phase. In our proposed methodology, the design and implementation phases work together for a high-productivity programming environment, as the designed layers can be manipulate by the programmer directly in the image incorporation and GIS database phase.

3.6 System Development after Time Instance $t_0$

Now, we discuss in detail our proposed methodology during system development and maintenance. This system development phase starts just after the first iteration is over at time $t_0$. We consider this case in our proposed methodology by introducing a pre-analysis phase in the traditional waterfall model. The reason for introducing the pre-analysis phase in our proposed methodology is given below.
Conventional methodologies for system modeling/development are generic therefore Universe of Discourse (UoD) needs to be well defined and delimited. Recently, Domain Specific Modeling has been finding favor by system designers to model specific domains such as mobile applications, VLSI design, etc. However, majority of business application areas are open-ended and hence are not suitable for Domain Specific Modeling. Also certain scientific and technological application areas such as Bioinformatics (in which new knowledge is being generated as a result of research and new discoveries) are less suitable to be modeled by Domain Specific Modeling. Scientific application areas such as Remote Sensing in which most of UoD artifacts are well-known, stable and concrete are better modeled by Domain Specific Modeling. For open-ended application areas and areas where new knowledge is constantly being generated, default knowledge (DK) is always input into analysis phase. In analysis phase these open-ended systems are delimited and their scope is then precisely defined. However, in application areas such as Remote Sensing, most of types of UoD are already known and it makes little sense to input entire DK in analysis phase. For this reason, we are introducing a pre-analysis phase in our proposed methodology.

There are different modes/situations when software is developed and maintained after the software development. These software development and maintenance situations except development of software from scratch are influenced by the types of changes that occur to software during and after the software development.

All possible changes that can occur during or after software development are given as follows:

3.6.1 Types of Changes

Type I: Change occurs to data of software (data change).

Type II: Change occurs to metadata/functional requirements of software.
Type III: Change occurs to both data and metadata (structural change).

Type IV: Change occurs due to both (data change and structural change).

As we have said earlier that these four (4) types of changes can affect the development and maintenance processes, the fact is also pointed out by Shah in (Shah, 2001). The water fall software development life-cycle is modified by Shah and based on this modified water fall software development life-cycle, a new framework is proposed for the prototype-based software development methodologies. Here for remote sensing software/applications, we extend the list of changes that is suggested by Shah. We note that this extended list influences the development and maintenance process as it has already been pointed out by Shah in (Shah, 2001).

The detail of the above types of changes is given below:

To elaborate the four (4) types of changes and modifications in an existing/developed application, we take an example of a canal passing through a mass of trees. Changes may occur to the geographical features of the application, which may affect the development and maintenance processes of this remote sensing application.

i) Type I: Change occurs to data of the application

This change occurs when there is change to data parameter of an already developed remote sensing application. This change type can simply be handled/ incorporated to the application without affecting FR or structure of the application and to incorporate it, we need less development effort.

In our example of canal passing through a mass of trees, if there is a change in the number of trees or flow rate of water in the canal, this change can be easily detected and incorporated directly by the application.

ii) Type II: Change occurs to functional requirements (FR) of the application.
Changes/modifications that occur to functional requirements of the application and they are categorized as Type II changes.

FR captures the planned behavior/functionality of an application. This behavior may be termed as service/tasks or functions that the application is needed to perform, whereas, the structure of an application provides both the structural and behavioral capabilities of an application. Structural requirements are defined by a set of instance variables and methods (or operations). In other words, the main difference between FR and structure (structural requirements) of an application is that, the set of FR may be a subset of the structural requirements.

Type II changes occur to a remote sensing application due to lack of communication between clients and developers, insufficient domain knowledge, change in UoD or changes in the requirements of clients.

Due to the change (Type II change) in the FR to a remote sensing application, the following two cases may arise.

**Case I:** Change to FR may not cause a structural change:

For example, in the running example, change to the FR is: Finding buildings of area greater than 100 square feet along the canal bank. This new requirement may be needed to incorporate in the cases such as urban planning, telecommunication, or environment monitoring, etc.

**Case II:** A functional Requirement may cause a structural change:

In the example, a new FR is: Capture all features in a 20 m corridor along the canal. The addition of new requirements (or the change) to the existing (or already developed) application, resulted in a structural change.

iii) **Type III:** Change occurs to structure of the application:
Every fresh data item is to be input for similarity check process; either known or a new artifact. The similarity check stage of our proposed methodology compares its similarity with the core elements of the existing artifacts.

Theorem given in Section 3.6.2 works on the data item for similarity check. If similarity exists in the input data, it will be treated as mature or immature artifact (depend upon the degree of similarity with the help of stake holders) and if, similarity does exist, it will be treated as a new artifact and then classify its type.

A structural change, depending upon its degree, may significantly alter the structure of an existing object to warrant its similarity check. This means that a Type III change will be handled in a way similar to when a new artifact is input to the system. The object which has undergone a structural change will be input to our similar check module as described in Section 3.6.2.

3.6.2 Similarity Check

Definition: Let ‘U’ is a set of classical elements denoted by ‘µ’ so, a Fuzzy subset B can be defined as:

\[
\{(\mu, \mu B(\mu))/ \mu \in U\} \tag{1}
\]

Where, \( \mu B(\mu) \) is known as the association of elements; describes the degree of associations between \( \mu \) and Set B; \( 0 \leq \mu B(\mu) \leq 1 \)

The logic represented here is based on a closed interval \([0, 1]\) rather to have either 0 or 1. Fuzzy Theory (FT) showed its authority in managing the uncertain and non-linear problems because of its elasticity. It can help with data classification problems (Yihua, et. al., 2009).

**Similarity:** In FT, similarity is a vital concept. It lies more or less within any two elements. We can define similarity as: \( S(a,c,r) \)
Which means ‘c’ is similar to ‘a’ in the property ‘r’. When relating to similarity, the object ‘c’,
the sample ‘a’ and the property ‘r’ should be dealt with.

Hence, similarity function can be defined (Yihua et. al, 2009).

\[
S(a,c)=\omega f(A \cap C) - \alpha f(A-C) - \beta f(C-A) \quad \text{----------(2)}
\]

Where

- \( S \rightarrow \) represents similarity within source_object ‘a’ and target_object ‘c’
- \( A \) and \( C \) are the feature set of source_object ‘a’ and the target_object ‘c’
- \( A \cap C \) represents the public_features of ‘a’ and ‘b’, while \( (A-C) \) and \( (C-A) \)
  shows the distinct_features of ‘a’ and ‘b’, respectively;
- Parameter \( \omega, \alpha, \beta \) are weight coefficients.

The function given above represents that similarity increases as public_features of the
distinct_features increase or decrease.

Tversky, similarity measure algorithm states that:

i) \( A \cap C, \) when \( \alpha=\beta=0 \)

We are only interested that object A and C share some features. So more the features are
common, more similar objects are.

ii) \( A-C, \) when \( \alpha=1 \) and \( \beta=0 \)

We compare the features common to object A and C with those unique to object A.

iii) \( C-A, \) when \( \alpha=0 \) and \( \beta=1 \)

We compare the features common to object C and A with those unique to object B.

**Type IV:** Change is a combination of the above three (3) types of changes:

In Type IV, it can be a change to data, functional requirements and structure of a remote sensing
application. It is to be noted that if we go from the changes Type I to Type IV, the complexity of
incorporating these changers is in increasing order. The increased complexity from Type I to Type IV results in corresponding increase in development effort. Type II and Type III changes in which functional requirements changes and structural change need to be incorporated, result in more development effort as compared to Type I change in which only new data needs to be incorporated. Also, a change in already existing system needs more development effort as compared to a corresponding change in a system under development.
**Figure 3.19:** Methodology for remote sensing Applications development after time instance $t_0$ and change detection

Legend used: LP $\rightarrow$ Layer Prototype, FR $\rightarrow$ Functional Requirements, P $\rightarrow$ Prototype
3.6.3 Functionality of Pre-Analysis Phase

The pre-analysis phase consists of two sub phases:

i) Classification of Prototypes

ii) Metadata Extraction

The pre-analysis phase is triggered with the new facts which arrive during system development after scratch and the changes are occurred during system development. First we will discuss when the system is to be developed from scratch.

i) Classification of prototypes

This phase classifies artifacts into:

- New Prototypes
- Immature Prototypes
- Mature Prototypes

**New Prototypes** are previously known artifacts. These artifacts are passed onto Meta Data Annotation Sub-phase for subsequent input into Analysis Phase.

**Immature Prototypes** are partially recognized artifacts and needs further clarification and recognition, and are passed onto Analysis Phase to be known artifact with the help of stakeholders and domain experts.

**Mature Prototypes** are known to the system and are directly input to the design phase.

We have categorized the changes that may occur during RS Application development into [Type I, Type II, Type III and Type IV], already discussed in Section 3.6.

Type I can be handled by implementation phase, while pre-analysis phase handles Type II (when causes structural change) and Type III changes.

Figure 3.20 sub modules of “classification of prototypes” phase.
Figure 3.20: Decomposition of sub-phase "classification of prototypes"

These three types (mature, immature and new prototypes) are passed onto the analysis phase for subsequent processing on it.

i) Immature Prototypes: these are inputs to the analysis phase when classification of prototypes is to be done at pre-analysis phase and prototypes are not completely identified. These prototypes need further meta-data annotation for proper recognition, details of which is given in section 3.5.1.1.

ii) New prototypes: These prototypes are identified by the pre-analysis phase as a new prototype and will be handled by the analysis phase as described in Section 3.5.1.
The rest of the phases (analysis, design and implementation) in this stage (maintenance/after development) of the proposed methodology work in the same manner as discussed in Sections 3.5.1, 3.5.2 and 3.5.3, respectively.
Chapter 4

OPERATORS FOR FUNCTIONAL REQUIREMENTS

In this Chapter, we propose a set of operators that represents most of the Functional Requirements (FR) of remote sensing applications such as Geographical Information System (GIS) (Ahmad and Shah, 2012). This set of operators captures the behavior/functions of the applications, and they can be considered as the services provided by the applications. Generally, that software is considered as a good software/application which meets maximum user requirements (Sommerville, I., 2010). In the following sections, the proposed operators which define the functional requirements of remote sensing applications (such as GIS) are given. The algorithms of these operators are defined in Appendix II of our Thesis.

We define a Remote Sensing Application (RSA) by a set of images that have been created at different time instances, and each image further consists of a set of different layers.

Assume that we have total number of N elements (or components) in a RSA, as defined in Equation (1).

\[ RSA = \{(I_i, t_i) \mid 1 \leq i \leq N\} \]

In Equation (1), the 2-tuple \((I_i, t_i)\) is referred to as the \(i^{th}\) element (or component) of the RSA (Ahmad and Shah, 2012). Note that in Equation (1), \(I_i\) is the \(i^{th}\) image that has been created at the time instance \(t_i\). The time instance \(t_i\) satisfies the inequality, \(1 \leq t_i \leq \text{Now}\) where \(1 \leq i \leq N\). An image of a RSA, e.g., \(I_i\), has been created at the time instance \(t_i\), and it can further be written in term of its layer as follows. For example, we write layers of the \(i^{th}\) image as follows:

\[ I_i = \{L_{i,j} \mid 1 \leq j \leq n_j \leq 5\} \]
Note that in Equation (2) the image, $I_i$, has been created at the time instance, $t_i$, and $n_j \leq 5$ denotes the total number of layers in the image $I_i$. There can be at the maximum five different types of layers in an image but time instance associated with all these layers is the same, that is, the time instance when the corresponding image was created. For example, since the image $I_i$ was created at the time instance $t_i$, therefore, the time instance associated with all the layers of the image $I_i$ is $t_i$. Note that each image does not necessarily contain all five types of layers (Ahmad and Shah, 2012).

Now we specify the same type of layer in different images of a RSA. Assume $L_k$, is a set of layers of the $k^{th}$ type, and they are created at different time instances $t_{1k}$, $t_{2k}$, ……$t_{Nk}$, in different images of the RSA as it is denoted by Equation(1) (Ahmad and Shah, 2012).

Five different types of layers are available in RSAs. The physical representation of these five layers is water body, vegetation, urban area, railway line, and soil.

The set of layers, $L$, in RSA can be written as follows:

$$L = \{L_{j,k} \mid 1 \leq j \leq N, \ 1 \leq k \leq 5 \} \quad \text{.............(3)}$$

Note that in Equation (3), the first subscript, $j$, denotes total number of images in the RSA and the second subscript, $k$, denotes type of the layer. After formally defining a RSA and its components, now we define the proposed operators for the RSA (Ahmad and Shah, 2012).

Equation (2) contains the set of all available layers in an image at the time instant $t_i$.

The members of the set, $L_i$, in Equation (2) can be water body, vegetation, urban area, railway line, or soil.

Note that in this thesis, a time instance $t_i$ always lies in the time interval $[t_i, \text{now}]$, and in this time interval the time instance now represents the current time instance on the time line.
4.1 Distance Operator

The first operator $D$ computes the distance between two given locations (or points) at the time instance $t_i$, and it is defined as follows:

$$D_{ti} = (d(p_j, p_k), t_i) \quad \text{--------------------------}(4)$$

or

$$D_{ti} = (d, t_i) \quad \text{--------------------------}(4)'$$

In Equation (4), $p_j$ and $p_k$ are the two different point locations in two different layers belonging to the image $I_i$ as it has been defined in Equation (2). In the above equation, the time instance, $t_i$, represents the time when the image, $I_i$, has been created. In Equation (4)$'$ $d$ represents the Euclidean distance between two locations in two layers, and it can be computed by the following formula:

$$d((b_1, y_1), (b_2, c_2)) = \sqrt{(b_1 - b_2)^2 + (c_1 - c_2)^2} \quad \text{--------------------------}(5)$$

Figure 4.1: An image showing pixels of different colors, representing different layers
In this paragraph, we give physical interpretation/explanation of the above defined operator. In Figure 4.1, $d$ is actually the distance between two pixel locations in two different layers (represented in yellow and blue colors). The distance between two given locations can be computed by using the Euclidean distance formula given in Equation (5). These pixel locations (or coordinates) belong to two layers of the image, $I_i$.

### 4.2 Area Operators

In this section, we define a set of operators to find areas of different types of layers in an image, $I_i$ (as defined in Equation (2)) created at the time instance $t_i$. We refer to such type of area as a **temporal area**. This type of area can be useful and helpful in making some important decisions such as forest estimation, resource management, etc. over the period of time (Ahmad and Shah, 2012).

The temporal area operator is defined as $A_{t_i}(a, t_i)$, where $a$ represents the area at the time instance $t_i$, and it can be computed by using the Offset Method for irregular shapes (Mysore et al., 2009). Note that the time instance, $t_i$, used in the operator is the time when the image, $I_i$, was created and we are computing the area $a$, of a layer in the image.

![Figure 4.2: An irregular shaped area of a layer in an image, divided into segments](image)
There are many methods available in the literature to compute the area of irregular shapes (Mysore et al., 2009). The method we have selected to use here is called Offset Method. This method calculates the length of the line by measuring the length of the longest axis of the area line \((AB)\) (see Figure 4.2). In the next step, the method divides the length line \((AB)\) into equal segments (at points \(C, D, E, F\) and \(G\)) (see Figure 4.2). At each of these points, the distance across the area in a line perpendicular to the length line at each point (line \(C\) through \(G\)) is measured. These lines are called offset lines. Finally, lengths of all offset lines are added and the result is multiplied by the distance that separates these lines. The area \(a\) in definition of the operator can be computed by using Equation (6) (Ahmad and Shah, 2012).

\[
a = d \sum_{i=1}^{n} x_i
\]

(6)

In Equation (6), the terms \(d\) and \(x_i\) represent the distances between offset lines, and lengths of offset lines of any layer belonging to image, \(I_i\), in Equation (2), as shown in Figure 4.2. In Equation (6), \(n\) is the number of offset lines, \(d\) is the distance between offset lines, and \(x_i\)s are the lengths of offset lines (Mysore et al., 2009).

### 4.2.1 Maximum Area and Minimum Area Operators

Two operators to find maximum and minimum areas of layers in an image are proposed in this section. We identify two cases of these two operators, and they are given as follows:

**Case I**: Find a layer having maximum area and minimum area amongst all layers in a given image \(I_i\) which was created at the time instance \(t_i\) in the 2-tuple \((I_i, t_i)\).

**Case II**: Find maximum area and minimum area amongst all layer of same type in all images of an application (see Equation (1)), created at different time instances, \(t_1, t_2, \ldots, t_N\). Note that in this work, we have taken at maximum five (5) numbers of layers in an image but there can be more
than five. In the following section, consider both of these cases, respectively (Ahmad and Shah, 2012).

**Case I**

Now we propose two operators to find a layer having maximum area and minimum area amongst all layers in an image $I_i$ which was created at the time instance $t_i$.

The maximum area operator defined over a set of layers in an image $I_i$ (see Equation (2)) at the time instance $t_i$ is given in Expression (E1).

\[ \text{Max}_{t_i} \left( \{(L_{i,j}) \mid 1 \leq j \leq n_j \} \right) \text{-----------------------------(E1)} \]

In Expression (E1), the component of the 2-tuple $(L_{i,j}, j)$, $L_{i,j}$ denotes $j^{th}$ type layer of the image that is created at the time instance $t_i$. This operator works in two steps. In the first step it computes temporal areas, of each layer by of the image using Equation (6) and in the second step it computes maximum area among all computed areas of all layers of the image. The operator returns two things:

i) maximum area  ii) corresponding type of the layer. This operator can be useful to categorize an image. For example, if at the time instance *now*, the maximum area among all layers is the *urban* area, then we can say that this *urban area* is a city or a town. Similarly, if at the time instance *now* the maximum area among all layers is the *water body*, then one should plan for building a dam or planning for irrigation purpose and other utilization of water (Ahmad and Shah, 2012).

The operator in Expression (E2) finds a layer having minimum area amongst all layers in an image $I_i$ at any time instant $t_i$.

\[ \text{Min}_{t_i} \left( \{(L_{i,j}) \mid 1 \leq j \leq n_j \} \right) \text{-----------------------------(E2)} \]

In Expression (E2), the component of the 2-tuple $(L_{i,j}, j)$, $L_{i,j}$ denotes $j^{th}$ type layer of the image that is created at the time instance $t_i$. This operator works in two steps. In the first step it
computes temporal areas, of each layer by of the image using Equation (6) and in the second step it computes minimum area among all computed areas of all layers of the image (Ahmad and Shah, 2012). The operator returns two things: i) minimum area, ii) corresponding type of the layer. This operator can be useful to categorize an image.

**Case II**

As we have already mentioned that in this case, the two operators find maximum area and minimum area of a given type of layer (say \(k^{th}\)) in all images (that has been created at different time instances) of an application.

We assume that \(L_{i,k}\) is a \(k^{th}\) type layer of the \(i^{th}\) image as defined in Equation (3), and it has been created at the time instance \(t_i\) (Ahmad and Shah, 2012).

By using this definition of the layer, i.e., \(L_{i,k}\), now we define the following two operators:

\[
\text{Max}-k ((l,k) : 1 \leq i \leq N \leq \text{now} \quad \text{and} \quad 1 \leq k \leq 5) \text{--------------------}(E_3)
\]

In Expression (E_3), the first component of the 2-tuple \((l,k)\), that is, \(l\) denotes the set of images, and the second component, \(k\), denotes the type of layer. This operator works in two steps. In the first step it computes temporal areas, of each \(k^{th}\) type layers of all images of an application by using the formula given in Equation (6) and in the second step it computes maximum area among all computed areas of \(k^{th}\) layers for each image. The operator returns maximum area of a given type of layer (Ahmad and Shah, 2012).

This operator helps to find out a layer \((k^{th})\) having the maximum area among all \(k^{th}\) layers taken at different time instances \(t_i\) \((1 \leq i \leq N \leq \text{now})\). In this case, we can predict the growth rate of a particular layer (represented by an area in an application, e.g., growth profile of a city/town) in the future based on its growth rate in the past (Ahmad and Shah, 2012).

Similarly, we define the operator to compute minimum area as follows:
Min-\(k\) \((\{I_i, k\} \mid 1 \leq i \leq N \leq \text{now} \text{ and } 1 \leq k \leq 5)\)  

This operator also works in two steps. In the first step it computes temporal areas, of the given \(k^{th}\) type layer of all images of an application by using the formula given in Equation (6) and in the second step it computes minimum area among all computed areas of \(k^{th}\) layers of all images. The operator returns minimum area of the given type of layer.

This operator helps in finding a layer of a given \(k^{th}\) type layer having minimum area among all images of an application. Using this operator, we can predict the depletion rate of a particular layer (represented by an area in an application, e.g., vegetation) in the future.

The operators defined in Expression (E3) and Expression (E4) can be used in a number of planning and management activities such as in the future planning of the land. It can also be helpful maintaining harmony among sustainable resources and socio-economic needs of a region (Ahmad and Shah, 2012).

**4.3 Operators for Change Tracing**

These operators are useful in tracing the following three types of changes:

i) Change in Data

ii) Change in Structure

iii) Change due to combination of both (data change and structural change)

In the following sections, we define the operators to trace the above two types of changes.

**4.3.1 Operator to Trace Change in Data** We take two images, i.e., \(I_i\) and \(I_j\) of a Remote Sensing Application (RSA). These two images are created at two different time instances, i.e., \(t_i\) and \(t_j\), respectively. Assume that the temporal areas of two layers of the same type (say \(k^{th}\) type) in these two images are: \(A_{i,k}\) and \(A_{j,k}\), respectively. The subscripts \(k\) \((1 \leq k \leq 5)\) represents the type
of layer (such as water body, vegetation, soil, urban area or railway line) (Ahmad and Shah, 2012).

We define the operator to trace a data change in the areas $A_{i,k}$ and $A_{j,k}$ of the same type of layer, belonging to two different images as follows:

$$DC (A_{i,k}, A_{j,k})$$  \hspace{2cm} (E5)

This operator gives the difference between the two areas: $A_{i,k}$ and $A_{j,k}$ (i.e., $A_{i,k} - A_{j,k}$), and it traces a change in data that has occurred during a time interval $[t_i, t_j]$ (where $t_i < t_j$), in a given type of layer and belonging to two different images. In this operator, three results can occur and they are listed and discussed in the following three paragraphs (Ahmad and Shah, 2012).

i) This case can arise if the operator gives zero (0) as a result, (i.e., $DC (A_{i,k}, A_{j,k})=0$). It means that no change has occurred in the temporal areas of the two layers of the same type and belonging to two different images during the time interval $[t_i, t_j]$. In other words, the area of the $k^{th}$ type layer in these two different images remains unchanged during the time interval $[t_i, t_j]$.

ii) This case occurs if the operator returns a negative value, (i.e., $DC (A_{i,k}, A_{j,k})<0$), then it means that growth has occurred in the area of the $k^{th}$ type of layer during the time interval $[t_i, t_j]$ (Ahmad and Shah, 2012).

iv) This third result occurs if the operator defined in Expression (7), returns a positive value (i.e., $DC (A_{i,k}, A_{j,k})>0$), then it means that depletion in the temporal area has occurred in the $k^{th}$ type layer during the during the time interval $[t_i, t_j]$.

4.3.2 Operator to Trace Structural Change (SC)

The following operator finds a structural change that has occurred in the images $I_i$ and $I_j$ of a Remote Sensing Application (RSA) in a given time interval.
These two images are created at two different time instances, i.e., \( t_i \) and \( t_j \), respectively. Assume that the *temporal areas* of two layers of the same type (\( k^{th} \) type) in these two images are: \( A_{i,k} \) and \( A_{j,k} \), respectively. The subscripts \( k (1 \leq k \leq 5) \) represents the type of layer (such as water body, vegetation, soil, urban area or railway line).

We define the operator to trace *structural change* in these two different images as follows:

\[
SC \left( A_{i,k}, A_{j,k} \right) \quad \text{------------------------(E6)}
\]

This operator gives the difference between two areas \( A_{i,k} \) and \( A_{j,k} \) (i.e., \( A_{i,k} - A_{j,k} \)), and it traces a *structural change* that has occurred during the time interval \([t_i, t_j]\), (where \( t_i < t_j \)), in a given type of layer and belonging to two different images (Ahmad and Shah, 2012).

A *structural change* occurs, if any of the following conditions exists:

i) \[ SC \left( A_{i,k}, A_{j,k} \right) = |A_{i,k}| \quad \forall \ (1 \leq k \leq 5) \]

ii) \[ SC \left( A_{i,k}, A_{j,k} \right) = |A_{j,k}| \quad \forall \ (1 \leq k \leq 5) \]

In Expression (E6), there will be a *structural change* (SC), if there exists a particular layer in any image taken at the time instance \( t_i \) and is absent in an image taken at some other time instance \( t_j \) and vice versa.

For example, if there exits water body layer, represented by an area, \( A_{i,k} \), in an image taken at any time instance, \( t_i \) and that layer (water body), represented by an area, \( A_{j,k} \) does not exist in any image taken at any time instance \( t_j \). So, a *structural change* (layer deletion) has occurred during the time interval \([t_i, t_j]\), (where \( t_i < t_j \)). And, if, there exist a water body layer in an image taken at any time instance \( t_j \), and was not present in an image taken at time instance \( t_i \), hence, structural change has occurred due to addition of a new layer (water body) (Ahmad and Shah, 2012).
4.3.3 Operators to Trace Combined Data and Structural Changes

In this section we defined the operators those can be used to trace back the combined data and structural changes in two images of an application. Note that these two images are taken at the two different time instances, e.g., \( t_i \) and \( t_j \), and they have been recorded in the application.

To define these operators, we identify the following five (5) different cases of this type of change and are shown in Figure 4.3.

![Structured chart showing all five (5) cases of the fourth type of change (data change and structural change)](image)


**Figure 4.3:** Structured chart showing all five (5) cases of the fourth type of change (data change and structural change)

The structured chart given in Figure 4.3 shows five (5) different modules/blocks, labeled as Module (a), Module (b), Module (c), Module (d) and Module (e), and they represent five cases (i.e., Case I - Case V) of the change, respectively. In the figure, Module (a) (Case I) is the case of the change when depletion occurs in layer(s) and new layer(s) are added to an application. Note that the depletion in layer(s) causes a data change, whereas addition of new layer(s) causes a structural change to an application, and the combined effect of both changes is referred to as
the fourth type of change (see Section 4.3.3). Module (b) handles the case of the changes when depletion and growth occurs in the existing layer(s) and addition of new layer(s) in an application. This change due to depletion and growth in layer(s) causes a data change and addition of new layer(s) causes a structural change in the application. Similarly, Module (c) handles a data change when both depletion and growth occur in layer(s), and a structural change occurs due to deletion of layer(s). In Figure 4.3, Module (d) represents a data change that occurs due to growth in layer(s) and a structural change that occurs due to deletion of layer(s) in an application. We identify this type of change as the fourth type of change (combination of data change and structural change) i.e., Case IV of the fourth type of change (data change and structural change), shown in Figure 4.3. Module (e) handles the case when a data change occurs due to depletion and growth in the existing layer(s), a structural change due to deletion of existing layer(s) and addition of new layer(s) in an application, shown in Figure 4.3 (Ahmad and Shah, 2012).

The details of this type of change (type four) and its cases (Case I - Case V) are given in the next sections (Section 4.3.3.1 – 4.3.3.5).

4.3.3.1 Case I

This is the case of fourth type of change (see Section 4.3.3) when data and structural changes occur simultaneously, and it is shown as Module (a) in Figure 4.3.

This case occurs when it contains the following combination of changes:

i) One or more layer(s) are depleted (partial deletion). This change triggers a data change.

ii) One or more new layer(s) are added to an application. This change triggers a structural change in an application.

Depletion in layer(s) triggers the addition of one or more layers in the application.
This case (Case I) can further be categorized into four (4) sub-cases shown in Figure 4.4.

**Figure 4.4**: Sub-cases of case I, represented by module (a) in Figure 4.3

In this paragraph, we give details about each sub-case (Module (1) – Module (4)) of Case I and combination of changes (data change and structural change) included in them.

Sub-case which is represented by Module (1) in Figure 4.4, have the following combination of changes:

Module (1):  
- (a_{i1}) Depletion in one layer  
- (a_{i2}) Addition of one new layer  

In the above combination of changes (i.e., a_{i1}), first subscript (i) represents case number and the 2\textsuperscript{nd} subscript (i.e., 1) represents the change number. The same notations are used in other cases and their sub-cases as well.

Note that, as mentioned earlier in this section, the depletion in a layer causes a data change, whereas the addition of a new layer causes a structural change.

Similarly, the remaining sub-cases of Case I are represented by Module (2) – Module (4) in Figure 4.4 and combinations of these changes are given below:

Module (2):  
- (b_{i1}) Depletion in one layer
(b₁₂) Addition of more than one new layer

Module (3): (c₁₁) Depletion in more than one layer
(c₁₂) Addition of one new layer

Module (4): (d₁₁) Depletion in more than one layer
(d₁₂) Addition of more one new layer

We illustrate this case (Case I) through an example shown in Figure 4.5.

Suppose an image \( I_i \), taken at any time instance, \( t_i \) and it contains four (4) types of layers (vegetation, soil, water body and railway line). We take another image \( I_j \), taken at the time instance, \( t_j \). During the time interval \([t_i, t_j]\), (where \( t_i < t_j \)), if some vegetation area is to be cut down (partial deletion) (which is data change) and urban area have been constructed, it means new layer (urban has been added) (which is a structural change). This situation causes, depletion in the vegetation layer (data change) and addition of a new layer (urban) (structural change), as shown in images \( I_i \) and \( I_j \) in Figure 4.5. Figure 4.5 depicts sub-case (Module (1)) of Case I, when there is depletion in one layer and addition of one new layer in an application.

![Figure 4.5: Graphical representation of fourth type of change (data and structural change) in images taken at \( t_i \) and \( t_j \) (Case I)](image-url)
4.3.3.2 Case II

This Case (Case II) is different from Case I by an extra change included in the combination of changes i.e., the growth in layer(s) in addition to depletion of layer(s) and addition of new layer(s) in an application. This case occurs when depletion in layer(s) (data change) triggers growth in layer(s) (data change) and addition of new layer(s) (structural changes), as shown in Figure 4.3 (see Module (b)).

Seven (7) Modules (Module (1) –Module (7)) of Module (b) have been identified as shown in Figure 4.6, depending on different combinations of data and structural changes.

Module (b)  

Figure 4.6: Representation of sub-cases of Case II (Module (b) in Figure 4.3)

Three (3) different combinations of these two changes (data change and structural change) are given as follows:

i) It is the case if the depletion occurs in one or more layers of an application, then this change triggers a data change (Ahmad and Shah, 2012).

ii) If growth occurs in one or more layers of an application (i.e., partial addition of layer(s)), then this change triggers a data change (Ahmad and Shah, 2012).
iii) If addition occurs in one or more layers of an application, then this change triggers a structural change in the application (Ahmad and Shah, 2012).

Depletion (i.e., i) described above) and growth (i.e., ii) described above) in layer(s) causes a data change whereas addition (i.e., iii) as mentioned above) of new layer(s) causes a structural change, and the whole effect of these changes, we mentioned it as a fourth type of change (see Section 4.3.3).

Depletion in layer(s) triggers the growth and addition of layer(s). Note that the depletion triggers both growth and addition of new layers.

The details of the sub-cases (i.e., Module (1) – Module (7)) shown in Figure 4.6 are given in the following paragraphs.

Module (1): \((a_{1})\) Depletion occurs in one layer

\(((a_{2})\) Growth occurs in one layer

\(((a_{3})\) Addition of one new layer

Module (2): \((b_{1})\) Depletion occurs in more than one layer

\(((b_{2})\) Growth occurs in one layer

\(((b_{3})\) Addition of one new layer

Module (3): \((c_{1})\) Depletion occurs in one layer

\(((c_{2})\) Growth occurs in one layer

\(((c_{3})\) Addition of more than one new layer

Module (4): \((d_{1})\) Depletion occurs in one layer

\(((d_{2})\) Growth occurs in more than one layer

\(((d_{3})\) Addition of one new layer

Module (5): \((e_{1})\) Depletion occurs in more than one layer
\((e_{II2})\) Growth occurs in more than one layer

\((e_{II3})\) Addition of one new layer

Module (6): \((f_{II1})\) Depletion occurs in one layer

\((f_{II2})\) Growth occurs in more than one layer

\((f_{II3})\) Addition of more than one new layer

Module (7): \((g_{II1})\) Depletion occurs in more than one layer

\((g_{II2})\) Growth occurs in one layer

\((g_{II3})\) Addition of more than one new layer

We illustrate this case (Case II) through an example shown in Figure 4.7.

Suppose an image \(I_i\), taken at any time instance, \(t_i\), and it contains four (4) types of layers (vegetation, soil, water body, and railway line). We take an image \(I_j\), taken at any time instance, \(t_j\). During the time interval \([t_i, t_j]\), (where \(t_i < t_j\)), if some area of soil is depleted (which causes data change) and vegetation area is grown up (data change), covering some area from depleted soil and a new layer (urban) is added (structural change) (which was not present in an image \(I_i\)). This situation occurs due to depletion in the soil layer (data change), growth in vegetation layer (data change) and addition of new layer (urban) (structural change) and is shown by images \(I_i\) and \(I_j\) in Figure 4.7.

Figure 4.7 represents the sub-case (Module (1) of Case II of the fourth type of change when there is depletion and growth in one layer each and addition of one new layer.
4.3.3.3 Case III

This case occurs when depletion and deletion of layer(s) triggers growth in layer(s). Depletion and growth in layer(s) causes data change whereas deletion of layer(s) causes structural change. In Figure 4.3 this case is represented by Module (c) and detailed structured chart of this Module and its sub-modules (i.e., Module (1) -Module (7)) are shown in Figure 4.8.
In this paragraph, we represent different combination of changes included in sub-cases of Case III of fourth type of change (Ahmad and Shah, 2012).

Seven modules (Module (1) - (7)) of Case III deal the following seven situations and they are given below.

Module (1):  \((a_{III1})\) Depletion occurs in one layer

\((a_{III2})\) Deletion of one layer occurs

\((a_{III3})\) Growth occurs in one layer

Module (2):  \((b_{III1})\) Depletion occurs in more than one layer

\((b_{III2})\) Deletion of one layer occurs

\((b_{III3})\) Growth occurs in one layer

Module (3):  \((c_{III1})\) Depletion occurs in one layer

\((c_{III2})\) Deletion of more than one layer occurs

\((c_{III3})\) Growth occurs in one layer

Module (4):  \((d_{III1})\) Depletion occurs in one layer

\((d_{III2})\) Deletion of one layer occurs

\((d_{III3})\) Growth occurs in more than one layer

Module (5):  \((e_{III1})\) Depletion occurs in more than one layer

\((e_{III2})\) Deletion of more than one layer occurs

\((e_{III3})\) Growth occurs in one layer

Module (6):  \((f_{III1})\) Depletion occurs in one layer

\((f_{III2})\) Deletion of more than one layer occurs

\((f_{III3})\) Growth occurs in more than one layer

Module (7):  \((g_{III1})\) Depletion occurs in more than one layer
(g_{III2}) Deletion of one layer occurs

(g_{III3}) Growth occurs in more than one layer

We illustrate this case through an example shown in Figure 4.9.

Suppose an image $I_i$, taken at any time instance, $t_i$, and it contains five (5) types of layers (vegetation, soil, urban, water body and railway line). We take a second image $I_j$, at any time instance, $t_j$. During the time interval $[t_i, t_j]$, (where $t_i < t_j$), if soil layer is deleted (this case causes structural change), vegetation layer is depleted (partial deletion) (data change) and urban area is developed more (growth) covering area of deleted (soil layer) layer and depleted (vegetation layer).

This case shown in Figure 4.9 explains the growth of urban layer (data change), depletion in vegetation layer (data change) and deletion of soil layer (structural change).

Figure 4.9 represent sub-case (Module (1)) of Case III when there occur deletion of one layer, depletion of one layer and growth of one layer (Ahmad and Shah, 2012).

\[ I_i \]

Vegetation

Soil

Urban

Railway Line

Water Body

\[ I_j \]

Vegetation

Urban

Railway Line

Water Body

**Figure 4.9:** Graphical representation of fourth type of change (data and structural change) in images taken at $t_i$ and $t_j$ (Case III)
4.3.3.4 Case IV

This case occurs when there is deletion and growth of the existing layer(s). Deletion (structural change) triggers the growth (data change) of layer(s). In Figure 4.3 this case is shown as Module (d). Four (4) sub-cases of this case are shown by Module (1) – Module (4)) in Figure 4.10.

**Module (d)**

![Diagram](image)

**Figure 4.10:** Sub cases of Case IV, represented by Module (d) of structured chart shown in Figure 4.3

Details of these sub-cases represented by Module (1) – Module (4), are given in the following paragraphs. Case IV deals with the following combination of changes in the images taken at two different time instances, \(t_i\) and \(t_j\).

i) One or more layers are deleted from an application.

ii) Growth occurs in one or more layers of an application.

Deletion of layer(s) triggers the growth in layer(s) of an application.

We can further categorize Case IV into the following four sub-cases, represented as Module (1) - Module (4), shown in Figure 4.10. In the figure these four sub-cases includes the following combination of changes:

Module (1): \((a_{IV1})\) Deletion of one layer

\((a_{IV2})\) Growth in one layer
Module (2):  \( (b_{IV_1}) \) Deletion of one layer

\( (b_{IV_2}) \) Growth occurs in more than one layer

Module (3):  \( (c_{IV_1}) \) Deletion of more than one layer

\( (c_{IV_2}) \) Growth occurs in one layer

Module (4):  \( (d_{IV_1}) \) Deletion of more than one layer

\( (d_{IV_2}) \) Growth occurs in more than one layer

We illustrate this case (Case IV) through an example shown in Figure 4.11.

Suppose an image \( I_i \), taken at the time instance \( t_i \), and it contains five (5) types of layers (vegetation, soil, urban, water body and railway line). We take another image \( I_j \), taken at the time instance, \( t_j \). During the time interval \([t_i, t_j]\) (where \( t_i < t_j \)), if vegetation layer is to be cut down and there is no more vegetation (which is a structural change) area shown in the image \( I_j \), and soil layer has been increased (growth occurred) (i.e., data change).

Figure 4.11 shows sub-case (Module (1) of Case IV when one layer is deleted and growth in on layer occurs.

\[\text{Figure 4.11: Graphical representation of fourth type of change (data and structural change) in images taken at } t_i \text{ and } t_j \text{ (Case IV)}\]
### 4.3.3.5 Case V

This case is shown as Module (e) in Figure 4.3 and occurs when the following combinations of changes occur:

i) One or more layers are deleted from an application.

ii) One or more layers are depleted from an application.

iii) One or more layers are added to the application.

iv) Growth occurs in one or more layers

Note that the depletion and growth in layer(s) cause data change whereas the deletion and addition of new layer(s) cause the structural change. The combined result of these two changes is referred to as the fourth type of change as mentioned in Section 4.3.3.

The sub-cases (Module (1) – Module (5)) of this case (Case V) are shown in Figure 4.12.

---

**Module (e)**

[Diagram showing sub-cases of Case V]

**Figure 4.12:** Sub-categories of Case V into five (Module (1), (2), (3), (4) and (5))

We give details about sub-cases of Case V in this paragraph. The combinations of changes included in these sub-cases are as follows:

Module (1):  
(\text{av}_1) \text{ Depletion in one layer} 

(\text{av}_2) \text{ Deletion of one layer}
(aV3) Growth occurs in one layer

(aV4) Addition of one new layer

Module (2):  (bV1) Depletion occurs in one layer

(bV2) Deletion of one layer

(bV3) Growth occurs in more than one layer

(bV4) Addition of one new layer

Module (3):  (cV1) Depletion occurs in more than one layer

(cV2) Deletion occurs in one layer

(cV3) Growth occurs in one layer

(cV4) Addition of one new layer

Module (4):  (dV1) Depletion occurs in one layer

(dV2) Deletion of more than one layer

(dV3) Growth occurs in one layer

(dV4) Addition of one new layer

Module (5):  (eV1) Depletion occurs in one layer

(eV2) Deletion of one layer

(eV3) Growth occurs in one layer

(eV4) Addition of more than one new layer

For better understanding of this case (Case V), we take an example shown in Figure 4.13.

Suppose an image $I_i$ of an application taken at the time instance, $t_i$, containing four types of layers (vegetation, soil, water body and railway line). We consider another image $I_j$ of the application taken at the time instance, $t_j$. During the time interval $[t_i, t_j]$ (where $t_i < t_j$), if some part of vegetation layer is to be cut down, by increasing the soil area (which causes a data
change) and *railway line* is to be put off, and *urban area* has been developed at the same place, by replacing *railway line*. Therefore, deletion of layer (*railway line*) and addition of a new layer (*urban layer*), causes a structural change in the application (see Figure 4.13).

Figure 4.13 represents sub-case (Module (1) of Case V).

**Figure 4.13:** Graphical representation of fourth type of change (data and structural change) in images taken at time instances, $t_i$ and $t_j$ (Case V)

To have a better and quick understating of all cases (Case I – Case V), which are discussed in Section 4.3.3.1 - Section 4.3.3.5, we presented here a comparative analysis Table 4.1 which shows different combinations of changes and their corresponding cases.
Table 4.1: Effects of different types of changes on different cases (Case I- Case V)

<table>
<thead>
<tr>
<th>Parameter (change type)</th>
<th>Case I</th>
<th>Case II</th>
<th>Case III</th>
<th>Case IV</th>
<th>Case V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer(s) Depletion</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>(data change)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer(s) Growth</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>(data change)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer(s) Deletion</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>(structural change)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer(s) Addition</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>(structural change)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1 clearly represents that Case I occurs due to layer(s) depletion and layer(s) addition whereas in Case II there is layer(s) growth too in addition to layer(s) depletion and layer(s) addition. In Table 4.1, Case III occurs due to layer(s) depletion, layer(s) growth and layer(s) deletion. Similarly, Case IV occurs when there is layer(s) deletion and layer(s) growth. In Case V, we have all types of changes as mentioned above (Ahmad and Shah, 2012).

**Operator for Type IV Change**

In this section we define operator to trace back the fourth type of change, i.e., combined data and structural change. This operator, we defined here is a combination of the two operators that have defined earlier in Section 4.3.1 and Section 4.3.2 to trace data change and structural change, respectively.

We define the operator in Expression (E7) for the fourth type of change (see Section 4.3.3) and derivation of its five cases (see Section 4.3.3.1- Section 4.3.3.5) from the defined operator.
In Expression (E7), the first part of the operator, i.e., $DC(A_{i,k}, A_{j,k})$, handles the data change if it occurs otherwise there is no change if it is equal to zero (i.e., $DC(A_{i,k}, A_{j,k}) = 0$, then no data change) (Ahmad and Shah, 2012). This part identifies the change by computing the difference between the two temporal areas, $A_{i,k}$ and $A_{j,k}$ (i.e., $A_{i,k} - A_{j,k}$) of two layers of the same type (say $k^{th}$ type) but they belong to two different images, i.e., $I_i$ and $I_j$, of a Remote Sensing Application (RSA), taken at the two time instances $t_i$ and $t_j$, respectively. The subscripts $k$ ($1 \leq k \leq 5$) represents the type of layer (such as water body, vegetation, soil, urban area or railway line) (Ahmad and Shah, 2012).

Note that this change has occurred during the time interval $[t_i, t_j]$, (where $t_i < t_j$), in a same type of layer, but the layer belongs to two different images. The second part of the operator, i.e., $SC(A_{i,k}, A_{j,k})$ if $\exists A_{i,k} \land A_{j,k} = 0)$ of the expression gives the difference between the two temporal areas, $A_{i,k}$ and $A_{j,k}$ (i.e., $A_{i,k} - A_{j,k}$). This part traces a structural change that has occurred during the time interval $[t_i, t_j]$, (where $t_i < t_j$), in a same type of layers but they belong to the two different images. A structural change is identified if the operator finds that a particular layer (say $k^{th}$ type) is present in one image but absent in the other image and vice versa (in other words, $\exists A_{i,k} \land A_{j,k} = 0)$, in the two images $I_i$ and $I_j$, taken at two different time instances $t_i$ and $t_j$.

In this paragraph we present that how type I change (i.e., data change) and type II change (i.e., structural change) can be traced back from the operator given in Expression (E7).

If the first part of the operator, i.e., $DC(A_{i,k}, A_{j,k})$, gives value greater ($>0$) than zero ($0$) (i.e., $(DC(A_{i,k}, A_{j,k})>0))$ for two layers of the same type (say $k-th$), it means that $k$-th layer of an image $I_i$, represented by a temporal area $A_{i,k}$, having greater temporal area as compared to the temporal
area in an image taken at time instance \( t_j \). Hence, depletion (data change) has occurred in the temporal areas of two layers of the same type (say \( k-th \) type) belonging to two different images during the time interval \([t_i, t_j]\), and if in the second part of the operator, i.e., if \( \exists (A_{ik} \land A_{jk} \neq 0) \), then it means that no structural change has occurred. In other words, this is a case of Type I change (data change) as discussed in Section 4.3.1, because only data change has occurred and no structured change has been found (Ahmad and Shah, 2012).

Note that if there occurs a depletion in the \( k-th \) layer and no structured change has been occurred then there must also be a growth in some layer(s) other than the \( k-th \) layer, as we have five (5) layers \((1 \leq k \leq 5)\) in our case study.

Hence, depletion causes growth and vice versa (depletion \( \rightarrow \) growth) if there exists no structural change. This can be explained with the help of a graphical example given in Figure 4.14, where only data can be seen.

**Figure 4.14:** Representing data (depletion \( \rightarrow \) growth) change in the images \( I_i \) and \( I_j \), taken at time instances \( t_i \) and \( t_j \), respectively.
In Figure 4.14, there can be seen depletion in the soil layer and growth in the urban layer but no structural change has been found (i.e., number of layers in both the images, $I_i$ and $I_j$ are same) (Ahmad and Shah, 2012).

Similarly, if first part of the operator, $DC (A_{i,k}, A_{j,k})$, of Expression (E7) gives value equal to (=0) zero (0) (i.e., $(DC (A_{i,k}, A_{j,k})=0)$). It means, there occurs no data change but if the second part of the operator, i.e., $SC (A_{i,k}, A_{j,k})$, returns value equal to the absolute value of $A_{j,k}$ (i.e., $SC (A_{i,k}, A_{j,k}) = Abs(A_{j,k})$), then it means that a new layer $A_{j,k}$, has been added to the application during the time interval $[t_i, t_j]$ which was not present at the time instance $t_i$. Hence, only structural change has been occurred which is the case of type II change as discussed in Section 4.3.2.

As we have already discussed that the fourth type of change have further five cases (Case I – Case V) (Section 4.3.3.1 – Section 4.3.3.5) and their sub-cases. Now we derive these five cases from the operator given in Expression (E7), in the following five paragraphs:

i) If the first part of the operator, $DC (A_{i,k}, A_{j,k})$, of Expression (E7) gives value greater (>0) than zero (0) (i.e., $(DC (A_{i,k}, A_{j,k})>0)$). It means that depletion (data change) has occurred in the temporal area in the $k$-th type of layer during the time interval $[t_i, t_j]$, and if the second part of the operator, i.e., $SC (A_{i,k}, A_{j,k})$, returns value equal to the absolute value of $A_{j,k}$ (i.e., $SC (A_{i,k}, A_{j,k}) = Abs(A_{j,k})$), then it means that a new layer $A_{j,k}$, has been added to the application during the time interval $[t_i, t_j]$ which was not present at the time instance $t_i$. The combined effect due to the depletion and new layer addition, we define it as Case I (see section 4.3.3.1) of fourth type of change (i.e., data change and structural change). This type of change can also be shown in Table 4.1. The sub cases of Case I can be derived by investigating the number of layers to be depleted with the number of layer to be added. For example, if one layer is depleted and one new layer is added to the application, we refer it as a sub case (represented by Module (1) in Figure 4.4,
Section 4.3.3.1) of Case I. Other sub cases of Case I (Module (2) - Module (4) in Figure 4.4) can also be extracted in the same manner by tracing down different combinations of depleted layers and added new layers, as total number of layers in our case are five ($1 \leq k \leq 5$).

ii) If first part of Operator ($DC (A_{ik}, A_{jk})$ in Expression (E7) gives value greater ($>0$) than zero ($0$) (i.e., $DC (A_{ik}, A_{jk}) > 0$) for any $k$-th type layer (depletion) and also gives value less than ($<0$) zero ($0$) (growth) for any other layer ($1 \leq k \leq 5$ ) during the time interval $[t_i, t_j]$. And if the 2$^\text{nd}$ part of Operator in Expression (E7) ($SC (A_{ik}, A_{jk})$), returns value equal to the Absolute $A_{jk}$ (i.e., $SC (A_{ik}, A_{jk}) = \text{Abs}(A_{jk})$), it means that new layer $A_{jk}$, has been added to the application during the time interval $[t_i, t_j]$ which was not present at time instance $t_i$. The combined effect due to depletion, growth and new layer addition, we define it as Case II of fourth type of change (i.e., data change and structural change) shown in Table 4.1. The sub cases of Case II can also be derived by investigating the number of layers to be depleted, number of layers to be partially added (growth) and number of layers to be added. For example, if depletion and growth occurs in one layer each and one new layer is added to the application, we refer it as a sub case of Case II represented by Module (1) in Figure 4.6, Section 4.3.3.2. Other sub cases (Module (2) – Module (7) in Figure 4.6) of Case II can also be extracted in the same manner by tracing down different combinations of depleted layers, growth in layers and added new layers, as total number of layers in our case are five ($1 \leq k \leq 5$) (Ahmad and Shah, 2012).

iii) If first part of Operator ($DC (A_{ik}, A_{jk})$ in Expression (E7) gives value greater ($>0$) than zero ($0$) (i.e., $DC (A_{ik}, A_{jk}) > 0$) for any $k$-th type layer and also gives value less than ($<0$) zero ($0$) for any other layer ($1 \leq k \leq 5$ ) during the time interval $[t_i, t_j]$. And if the 2$^\text{nd}$ part of Operator in Expression (E7) ($SC (A_{ik}, A_{jk})$), returns value equal to the Absolute $A_{ik}$ (i.e., $SC (A_{ik}, A_{jk}) = \text{Abs}(A_{ik})$), it means that layer $A_{ik}$, has been deleted from the application during time interval $[t_i,$
which was present at time instance \( t_i \). The combined effect due to depletion, growth and deletion of layer(s), we define it as Case III of fourth type of change (i.e., data change and structural change) shown in Table 4.1. The sub cases of Case III can be derived by investigating the number of layers to be depleted, number of layers to be partially added (growth) and number of layers to be deleted. For example, if one layer is depleted, and there is a growth in one layer and one layer is deleted from the application, we refer it as a sub case of Case III represented by Module (1) in Figure 4.8, Section 4.3.3.3. Other sub cases (Module (2) – Module (7) in Figure 4.8) of Case III case can also be extracted in the same manner by tracing down different combinations of depleted layers, growth in layers and deleted layers, as total number of layers in our case are five (\( 1 \leq k \leq 5 \)).

iv) If first part of Operator \( DC(A_{ik}, A_{jk}) \) in Expression (E7) gives value less (<0) than zero (0) (i.e., \( DC(A_{ik}, A_{jk}) < 0 \)) for any \( k \)-th type layer during the time interval \([t_i, t_j]\). And if the 2\textsuperscript{nd} part of Operator in Expression (E7) \( SC(A_{ik}, A_{jk}) \), returns value equal to the Absolute \( A_{ik} \) (i.e., \( SC(A_{ik}, A_{jk}) = Abs(A_{ik}) \)), it means that layer \( A_{ik} \), has been deleted from the application during the time interval \([t_i, t_j]\) which was present at time instance \( t_i \). The combined effect due to growth in layer(s) and deletion of layer(s), we define it as Case IV of fourth type of change (i.e., data change and structural change) shown in Table 4.1. The sub cases of Case IV can be derived by investigating the number of layers to be partially added (growth) and numbers of layers to be deleted. For example, if there is a growth in one layer and deletion of one layer from the application, we refer it as a sub case (represented by Module (1) in Figure 4.10, Section 4.3.3.4) of Case IV. Other sub cases (Module(2) – Module (4)) shown in Figure 4.10 of Case IV case can also be derived in the same manner by tracing down different combinations of growth in layer(s) deleted layers, as total number of layers in our case are five (\( 1 \leq k \leq 5 \)).
v) If first part of Operator \((DC(A_{i,k}, A_{j,k}))\) in Expression (E7) gives value greater \((>0)\) than zero \((0)\) (i.e., \(DC(A_{i,k}, A_{j,k}) > 0\)) for any \(k\)-th type layer and also gives value less than \((<0)\) zero \((0)\) for any other layer \((1 \leq k \leq 5)\) during the time interval \([t_i, t_j]\). And if the 2\(^{nd}\) part of Operator in Expression (E7) \((SC(A_{i,k}, A_{j,k}))\), returns value equal to the Absolute \(A_{i,k}\) (i.e., \(SC(A_{i,k}, A_{j,k}) = Abs(A_{i,k})\)) for any \(k\)-th type layer and also returns value equal to Absolute \(A_{j,k}\) (i.e., \(SC(A_{i,k}, A_{j,k}) = Abs(A_{j,k})\)) for any other layer it means that layer \(A_{i,k}\), has been deleted from the application and also new layer \(A_{j,k}\) has been added to the application during the time interval \([t_i, t_j]\). The combined effect due to depletion, growth, addition and deletion of layer(s), we define it as Case V of fourth type of change (i.e., data change and structural change) shown in Table 4.1. The sub cases of Case V can be derived by investigating the number of layers to be depleted, partially added (growth) and number of layers to be added, deleted. For example, if one layer is depleted, growth in one layer and one layer is added and one layer is deleted from the application, we refer it as a sub case of Case V represented by Module (1) in Figure 4.12, Section 4.3.3.5. Other sub cases (Module (2) – Module (5) in Figure 4.12) of Case V case can also be extracted by adopting the same procedure, that is by tracing down different combinations of depleted layers, growth in layers and added and deleted layers, as total number of layers in our case are five \((1 \leq k \leq 5)\).
Chapter 5

CASE STUDIES

In this chapter, we present three (3) case studies to illustrate the main steps (pre-analysis, analysis and design phases) of our proposed methodology. To achieve this goal, we took cases to develop a remote sensing system when very first image encounters and then to incorporate the changes that occur afterward images. For this purpose, we decided to take case study I of rapidly developed site with the passage of time, along Islamabad highway.

Islamabad is the 10th largest city Pakistan, having a population of 1.74 million (2009). It is situated in the Pothohar_Plateau in the north of Pakistan. Islamabad was built during 1960s to replace capital of Pakistan (Karachi). Historically it was a part of the crossroads of Punjab and NWFP.

The dynamic and complex entities belong to the cities which represent a heterogeneous mixture of residue nature, semi natural, modified. The components of natural and cultural realms determine the mixture of introduced and indigenous of urbanized areas. The result of urbanization is the destruction of natural ecosystems is the resultant of urbanization, followed by exchange of land into built-up structures and other man_made logical habitats i.e., gardens, lawns and parks etc.

5.1 Case Study I

In this section, we present Case Study I on urbanization along Islamabad highway and monitoring the changes that occur with the passage of time by using satellite images and GIS system. The settlement pattern and land_use around the suburban has affected badly by the
growth of the urban area and the population both. Especially, the area around the Islamabad highway is changing rapidly into urban rural fringe. This is because of the facility of transportation near highway. At Rawat junction the Grant Trunk (GT) road also joins the highway.

The methodical approach of satellite RS and GIS evolves as a basis for data consolidation across purposeful and organizational boundaries at site so the research was conducted to emphasize the urbanization trend.

The satellite RS gives the opportunity for effective assessment of the deforestation along the highway.

For this reason three year images of 1992, 2000 and 2009 are being studied to monitor the changes that occur along the Islamabad highway such change detection is essential for the improved supervision of the natural resources.

Because of the common facilities like employment, health and education, urbanization is increasing day by day.

The rural cites are combined in urban area producing urban rural fringe.

The main objectives of this case study are to analyze and design an application which may help achieve the following objectives:

i) To observe the growth in urban area and changes in urban settlements with the help of RS and GIS.

ii) To observe the deforestation phenomenon with the urban growth.

iii) To observe the landcover/landuse.

iv) To observe locally, the users of physical and financial resources.
In this case study, we will try to prove our proposed methodology with its custom/specialized/inherent features for example, metadata extraction, metadata annotation, classification of prototypes and layer identification etc. Hence is better suited for remote sensing applications as compared to the conventional methodologies.

The subject of our case study is the SPOT sub-scene of Islamabad and Rawalpindi.

The total land area of Islamabad is 906 sq.km including 220 sq.km area belong to administrative purposes, Margilla Hills National park 220 sq.km and rural Islamabad is 446 sq.km.

The Loi_Bher forest is located along Islamabad_highway and about 16 km far away from Rawalpindi. It is an un-classed forest having an area of 1087 acres and a big nullah is passing through it which provides water to the livestock and the local people. Some part of the forest is under the direct control of RDA.

To develop and maintain a remote sensing GIS system, we apply our proposed methodology to satellite images of the above mentioned location.

Following is the detailed description of the various phases of the modeling processes.

First we give the detailed description when the system is to be developed at first image at time $t_0$.

**5.1.1 Development of Islamabad Geographical System (IGIS) at $t_0$**

When we develop the IGIS at very first image taken at time instance $t_0$, our methodology works in two phases:

i) Analysis phase

ii) Design phase

i) **Analysis Phase**

The analysis phase of our proposed methodology consists of two sub-phases:

i) Metadata Extraction/Annotation
ii) Layer Determination

Our analysis phase started with the satellite images having the following information:

Years of image acquisition: 1992, 2000, 2009

Source of acquiring image: SPOT satellite (panchromatic mode)

Location of satellite image: Loi Bher forest along Islamabad highway

Purpose of acquiring image: To develop and maintain a remote sensing application and change detection with the passage of time

No. of satellite images: 3

Area covered by a satellite image: 201 Sq.km

Figure 5.1: SPOT image of the forest along Islamabad highway in 1992

Figure 5.2: SPOT image of the forest along Islamabad highway in 2000
We apply the first sub-phase (metadata extraction/annotation) of our proposed methodology to the above satellite images.

At time $t_0$, the image of year 1992 has been selected. We trace the processing of this image through our proposed methodology.

Stakeholders and developers consult and agree upon the feasibility study.

Based on these functional qualifications, the following schema was approved:
Table 5.1: Analysis of satellite image by using our proposed methodology

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Metadata1</th>
<th>Identification information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Metadata2</td>
<td>Metadata reference information</td>
</tr>
<tr>
<td></td>
<td>Metadata3</td>
<td>Spatial-reference-information</td>
</tr>
<tr>
<td></td>
<td>Metadata4</td>
<td>Entity- and attribute- information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer Determination</th>
<th>Type of Layers</th>
<th>vegetation</th>
<th>soil</th>
<th>railway</th>
<th>water</th>
</tr>
</thead>
<tbody>
<tr>
<td>(The Layers which are needed by the stake holders)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1 shows the analysis of the satellite image based upon our proposed methodology. The image is analyzed and find out the metadata necessary for the proper identification of the layers. These layers are identified by consulting the stake holders and problems statement of the remote sensing system. The functional requirements of the users are to be incorporated for the type of layers to be identified. For example the user is interested in finding the vegetation layer, soil layer, Railway line and water body layer as described in Table 5.1.

In the image shown in Figure 5.1, green color represents vegetation, cyan color represents the soil, railway line and water body have special meandering nature. These colors and special other features can further be sub-classified to identify the sub-categories of the vegetation, soil railway line and water bodies. For example, there is seen some quantity of parrot green color.
representing new vegetation, Prussian green color as a complete pattern of tree, dull grassy green as herbs, shrubs or bushes.

Metadata extraction and its types can be further explained by the following tables:

Table 5.2(a) shows the syntax and semantics of the metadata, identification information.

**Table 5.2 (a):** Detail of metadata (identification information) generated at Analysis phase, its syntax and semantics

<table>
<thead>
<tr>
<th>Metadata Type</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identification information</strong></td>
<td>Citation</td>
<td>Satellite image of SPOT panchromatic mode of year 1992 covering an area of 116 sq.km along Islamabad highway</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>The purpose to acquire the image is to find out the changes which occur along Islamabad highway with the passage of time e.g. urbanization. So topographic maps, guided maps and regional maps are also used for this purpose.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This satellite image was taken in 1992 and validate</td>
</tr>
<tr>
<td><strong>Time period of contact</strong></td>
<td>with the ground truth values and GIS maps</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Raster data set was captured and after pre-processing (radiometric correction, geo-referencing, image enhancement and image classification), it can be used for remote sensing application development</td>
<td></td>
</tr>
<tr>
<td><strong>Spatial domain</strong></td>
<td>The image was taken 16 km away from Islamabad at Loi Bher forest along highway. Bounding coordinates (BC):</td>
<td></td>
</tr>
<tr>
<td><strong>Keywords</strong></td>
<td>Satellite image, Loi Bher Forest, Islamabad highway, urbanization</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2(a), provides the basic information about the data set. It provides the information about the intension and purpose to acquire the data set. It provides the maintenance and update frequency to find out the changes that occur to the data set after the initial data set is completed. The geographic domain provides the boundary
coordinates that limit the coverage of a data set expressed by latitude values and longitude values.

Now we give the syntax and semantics values of the metadata “metadata reference information”.

**Table 5.2(b):** Detail of metadata (reference information) generated at Analysis phase, its syntax and semantics

<table>
<thead>
<tr>
<th>Metadata Type</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata Reference Information</td>
<td>Metadata date</td>
<td>The date that the metadata were created</td>
</tr>
<tr>
<td></td>
<td>Metadata review data</td>
<td>The data of the last review of the metadata(1992)</td>
</tr>
<tr>
<td></td>
<td>Metadata future review data</td>
<td>The data at which the next review should be taken(2000 and 2009)</td>
</tr>
<tr>
<td></td>
<td>Metadata contact</td>
<td>The party responsible for the metadata information</td>
</tr>
<tr>
<td></td>
<td>Metadata standard name</td>
<td>Spatial Data Transfer Standard(STDS)</td>
</tr>
</tbody>
</table>

The Table 5.2(b) gives information about the currnetness of the metadata information and the responsible party.
Table 5.2(c): Detail of metadata (spatial reference information) generated at Analysis phase, its syntax and semantics

<table>
<thead>
<tr>
<th>Metadata Type</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial Reference</td>
<td>Horizontal and vertical coordinate system definition</td>
<td>Quantity of latitude and longitude, defining the position of a point on the earth surface with respect to a reference spheroid.</td>
</tr>
<tr>
<td>Information</td>
<td>Map projection</td>
<td>It is the systematic representation of all or part of the surface of the earth on a plane or developable surface. Map projection parameters are a complete parameter set of the projection that was used for the dataset. The information provided shall include the names of the parameters and values used for the dataset that describes the mathematical relationship between the earth and the dataset.</td>
</tr>
</tbody>
</table>
Table 5.2(c) provides a description about the reference frame and means of encoding coordinates in the dataset.
Table 5.2(d): Detail of metadata (entity and attribute information) generated at Analysis phase, its syntax and semantics

<table>
<thead>
<tr>
<th>Metadata Type</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity and attribute information</td>
<td>Detailed description/Overview</td>
<td>Gives the detailed information about, entity, attributes and its values and characteristics encoded in the data set.</td>
</tr>
<tr>
<td>Entity type</td>
<td>Name of entity type</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Characteristics of an entity</td>
<td></td>
</tr>
<tr>
<td>Attribute label</td>
<td>Name of an attribute</td>
<td></td>
</tr>
<tr>
<td>Attribute domain values</td>
<td>These are the valid values that can be assigned for an attribute</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2(d) gives detail about the content of the data set, including the entity type and its attributes and domain values which can be assigned to them for example, names and definitions of features, attributes and attribute values.

This metadata is stored in text format. Because of its small size as compared to the data it describes, can easily be shareable. It can be easily available to anyone seeking it. This metadata makes data discovery easy and reduces the chances of data duplication. It can be used to provide mechanism for communicating between disparate components or systems.
Now we trace the satellite image through the 2\textsuperscript{nd} sub-phase (layer determination) of our proposed methodology.

By using our proposed methodology, we can find out the layers which are present the satellite snapshot. These layers can be determined by using the metadata described above and with the help of image interpretation elements given below in Table 8.3.

**Table 5.3:** Detail of layer identification process at analysis phase (syntax/semantics)

<table>
<thead>
<tr>
<th>Process</th>
<th>Syntax(element)</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Determination</td>
<td>Size</td>
<td>Helps to identify the object. For example, size of water bodies will help to differentiate lakes, tanks and rivers. For vegetation, trees and bushes and herbs etc.</td>
</tr>
<tr>
<td></td>
<td>Shape</td>
<td>It helps to identify different type of forest, vegetation, buildings etc.</td>
</tr>
<tr>
<td></td>
<td>Shadow</td>
<td>It provides the height of trees, buildings etc.</td>
</tr>
<tr>
<td></td>
<td>Tone</td>
<td>An object can be identify by its tone which is the nature of reflectance. For example, new vegetation, grownup vegetation etc.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The most important element to identify a layer in an object. Green patches will indicate vegetation whereas brown will</td>
</tr>
</tbody>
</table>
Color | indicate settlement. In our methodology, number of pixels represent different types of layers, shown in figure 3.13. In the SPOT image of 1992, there are three types of main layers are found, urban area, vegetation and soil, with different colors.

Texture | Repetitive small patches can be helpful to identify the exact type and nature of the layer. For example, homogeneous grassland.

Pattern | It represents regular pattern, for example, rows of houses can provide valuable information.

Relationships between the objects | All of the above mentioned elements can provide sufficient information to identity the relationship between layers with the earth surface.

Table 5.3 shows how different layer are to be identified with the help of metadata extracted and by evaluating different features of the layers present in that image. All these identified layers are passes to the design phase as an analysis report.
The analysis report can contain the metadata and the three types of layers (vegetation, urban and soil) as identified at this analysis phase.

A satellite image shown in Figure 5.1 is digital picture of forest along Islamabad highway, taken in 1992. The image used in this case study is of square shape consist of grid of pixels (a smallest unit of an image). The pixel values are the brightness level based on the reflected energy detected by the satellite sensors over the area of a land. Each pixel value can be interpreted as a gray shade, with lower pixel values assigned darker shades of gray. Total gray shades are 256 in an image. Image interpretation techniques are used for detection, identification, and measurement of specific features for further layer determination process.

Table below gives a detailed picture of the satellite image of year 1992 after analysis phase.

**Table 5.4:** Results of satellite image of year 1992 after analysis phase

<table>
<thead>
<tr>
<th>Year of image taken</th>
<th>No. of layers/prototypes</th>
<th>Name of Layer</th>
<th>Area covered (sq.km)</th>
<th>No. of Pixels present in each layer</th>
<th>Percentage of pixels in an image</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td>4</td>
<td>Vegetation</td>
<td>66.1240</td>
<td>671158</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Soil</td>
<td>90.2417</td>
<td>915953</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Railway line</td>
<td>20.3532</td>
<td>206584</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Water body</td>
<td>24.8757</td>
<td>252488</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 5.4 provides information about the layers present in an image taken in 1992. Four layers are shown with no. of pixels of each layer. 33 percent of pixels represent the vegetation, 45
percent represents the soil, 10, is to railway line and 12 percent of pixel colors represent the water body.

There may be other layers too in an image but identification of these layers depends upon the functional requirements of the user/stake holders. These layers at this stage are labeled with layer prototypes (LP).

Now we trace the processing of satellite image of year 1992 on design phase of our proposed methodology.

ii) Design Phase

The design phase of our methodology works on the input image in the form of layers which are identified at analysis level. The analysis report then further be processed and sub-layers are determined at design phase.

Table below shows how design phase works to find out the sub-layers and extraction of further metadata.

**Table 5.5(a): Metadata (data quality information) extraction at design phase**

<table>
<thead>
<tr>
<th>Process</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Quality Information</td>
<td>Attribute accuracy</td>
<td>An assessment of the accuracy of the identification of entities</td>
</tr>
<tr>
<td></td>
<td>Completeness report</td>
<td>Report about selection criteria, generalization and definition used for the derivation of a data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Accuracy report about the</td>
</tr>
</tbody>
</table>
Positional accuracy | position of satellite objects. It contains horizontal and vertical positional accuracies of spatial objects and positions of the data set.

Lineage | Information about events, parameters source data set and information about the responsible parties

Cloud cover | The extent of the cloud cover data set

The above Table 5.5(a) shows the metadata values extracted at design phase for further sub-layer classifications and affiliation of these sub-layers with the parent layers.

**Table 5.5(b):** Metadata (spatial data organization information) extraction at design phase

<table>
<thead>
<tr>
<th>Process</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial data organization information</td>
<td>Indirect Spatial data reference</td>
<td>Types and names of geographic features, addressing schemes or means of data set referencing</td>
</tr>
</tbody>
</table>
Direct spatial reference

Raster object information

Types and the number of spatial raster objects in the data set. For example, two or three dimensional locations in the data set.

**Table 5.5(c):** Sub-layers decomposition at design phase

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Sub-Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vegetation</td>
<td>New vegetation</td>
</tr>
<tr>
<td></td>
<td>A Complete pattern of trees</td>
</tr>
<tr>
<td></td>
<td>Herbs, bushes, shrubs etc</td>
</tr>
<tr>
<td></td>
<td>Fields</td>
</tr>
<tr>
<td>Soil</td>
<td>Plane</td>
</tr>
<tr>
<td></td>
<td>Pothohar</td>
</tr>
<tr>
<td>Railway line</td>
<td>Tram track</td>
</tr>
<tr>
<td></td>
<td>Rail track for local carrier</td>
</tr>
<tr>
<td>Water body</td>
<td>River</td>
</tr>
<tr>
<td></td>
<td>Lake</td>
</tr>
<tr>
<td></td>
<td>Sea</td>
</tr>
<tr>
<td></td>
<td>Pond</td>
</tr>
<tr>
<td></td>
<td>Dam</td>
</tr>
</tbody>
</table>
Table 5.5(c) shows the types of main layers present in an image and the decomposition of these layers into further sub-layers. All these can be done by using the image interpretation elements like shape, texture, tone, color. Shadow etc. these sub-layers and the corresponding layers are to be input to the implementation phase for coding and implementation. The design report contains all these layers and the metadata extracted at design phase.

**Implementation phase**

The designed layers are implemented one-to one at this phase.

Now we discuss the case study when the system is to be developed after implementation.

### 5.1.2 System development after implementation/maintenance

To discuss the case after implementation, we took the satellite image of Islamabad along highway of years 2000 and 2009 as shown in Figures 5.2 and 5.3 respectively. System development after implementation phase is started just after the first iteration at time $t_0$.

At time $t_1$, we have introduced the pre-analysis phase in our proposed methodology as mentioned in chapter 3 and the reason for the introduction of pre-analysis phase is also given there in chapter 3.

At time $t_1$, when the satellite image of year 2000 and 2009 are input to the pre-analysis phase of our proposed methodology, the similarity check procedure as given in section 3.1.3 is activated.

This procedure checks the similarity of the newly arrived information/data and finds out the four types of changes. These types of changes are given below:

**Type I change:** This change is present in the satellite image of year 2000 and 2009 as there seen some changes in pixel colors of year 1992. So, to incorporate such types of change, our
methodology, handles them by sending these changes to implementation phase. Also, there seen some new colors too in the image taken in years 2000 and 2009 of the same location along Islamabad highway. Hence, according to our proposed methodology, this type of change can also be incorporated by the pre-analysis phase of our proposed methodology. The detailed function of pre-analysis phase is given in chapter 3 section 3.6.3 of our proposed methodology. The satellite images of years 2000 and 2009 are to be compared at change identification module and check the similarity with the already developed IGIS database. We find four types of changes in the images. Type II change is due to data change in the image of years 2000 and 2009. As the colors of pixels of layers are slightly changed, some part of green color pixels are turned out to be brownish pink color pixels and the number of pixels of the green color are reduced and brown color pixels are increased. This type of change can be incorporated by the implementation phase of our proposed methodology. Type II change can be handled at analysis phase when are some functional requirement from the user. Type III changes are seen in the image as we have seen some new color to the satellite image of years 2000 and 2009. So, structural changes are also identified. The type IV change can occur due to all above mentioned type of changes.

The output of the pre-analysis phase is of three types of prototypes:

i) Mature prototypes

ii) Immature prototypes and

iii) New prototypes

*Mature prototypes* are those having some type of data change and already present in the IGIS database. *Immature* are those types of prototypes when there are some structural changes and need to be identified again. *New* are those which were not there in the IGIS database and we take care of those prototypes as the system is to be developed at time instance $t_0$. These new
prototypes are handled at analysis phase and proper layer determination is to be carried out there. The analysis and design phases work in a similar manner as in a case when the system is to be developed for the first image. The following Table shows the extent and type of changes in the three years satellite images along the Islamabad highway.

**Table 5.6: Comparison Table for change detection in satellite images of three years**

<table>
<thead>
<tr>
<th>Year of image taken</th>
<th>No. of layers/prototypes</th>
<th>Name of Layer</th>
<th>Area covered (sq.km)</th>
<th>No. of Pixels present in each layer</th>
<th>Percentage of pixels in an image</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td>4</td>
<td>Vegetation</td>
<td>66.1240</td>
<td>671158</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Soil</td>
<td>90.2417</td>
<td>915953</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Railway line</td>
<td>20.3532</td>
<td>206584</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Water body</td>
<td>24.8757</td>
<td>252488</td>
<td>12</td>
</tr>
<tr>
<td>2000</td>
<td>5</td>
<td>Vegetation</td>
<td>54.4325</td>
<td>552489</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Soil</td>
<td>94.2341</td>
<td>956476</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Railway line</td>
<td>20.3532</td>
<td>206584</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Water body</td>
<td>23.2347</td>
<td>235832</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Urban</td>
<td>10.7648</td>
<td>109262</td>
<td>5</td>
</tr>
<tr>
<td>2009</td>
<td>5</td>
<td>Vegetation</td>
<td>41.7683</td>
<td>423948</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Soil</td>
<td>97.3421</td>
<td>988022</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Railway line</td>
<td>20.3532</td>
<td>206584</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Water body</td>
<td>20.4532</td>
<td>207599</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Urban</td>
<td>22.3214</td>
<td>226562</td>
<td>11</td>
</tr>
</tbody>
</table>
Table 5.6 shows a comparison of the satellite image of different years taken along Islamabad highway. This Table shows a trend of urbanization along Islamabad highway. Our proposed methodology works on the input image of year 1992 and after image interpretation, finds out four types of layers (vegetation, soil, railway line and water body) present in the image. After then the satellite image of year 2000 are to be input to the pre-analysis phase of our proposed methodology. This image is to be compared with the existing IGIS system database. The change detection module of our proposed methodology compares its similarity with the existing prototypes and finds out the following four types of changes as mentioned in chapter 3.

Type I: change occur to the data of an application
This change is seen in vegetation, soil layer and water body layer of the previously taken image of the same site. There is seen some decrement of the percentage of pixels of the vegetation layer from 33 to 27 percent. Similarly, the soil layer is increased from 45 to 46 percent. This type of change is categorized as data change and can be incorporated by the implementation phase of our proposed methodology.

Type II: This type of change can occur from the functional requirements of the user and can be handled by the analysis phase of our proposed methodology.

Type III: change occur to the structure of an application
This type of change is seen in the satellite image of year 2000. New layer (urban) is seen in the image which was not present in the image of year 1992. Hence, there is structural change in the remote sensing application and can be incorporated by the pre-analysis phase of our proposed methodology.

Type IV: change is a combination of all above three types
Table 5.6 represents all above three types of changes. No change is seen in the railway line layer represented in the images of years 1992, 2000 and 2009. There is data change in the layers of urban, vegetation, water body and the soil layer. And the structural change is due to new layer added in the image of 2000 as shown in the Table 5.6.

Table 5.6 shows that, the number of pixels of vegetation layer are continuously decreasing with the passage of time. Hence, there is a trend of deforestation and due to introduction of new (urban) layer and continuously increasing with the passage of time, hence shows the trend of urbanization.

The total area covered by the Islamabad territory, the capital Pakistan, is about 906 square kilometer. And the total population density is 890 persons per square kilometer. The Table 5.6 shows the extent of urbanization from 1992 to 2009. The total number of colonies has been increased showing the demand of urban settlement. The vegetation which is situated along the Islamabad highway is known as Loi Bher vegetation and is about 16 km away from Islamabad. Deforestation is seen from the temporal data of different years.

The rural area settlement is seen on the right side of the highway in satellite image of year 1992. This rural location is seen about 5-10 km away from the Islamabad highway. The Islamabad highway consist of vegetation, grass, herbs and shrubs, soil, the set pattern of trees are also present there on the right side of the highway. A big current Nullah is passing through it. A railway line is crossing the highway and runs between Sihala and Rawat.

There are seen hell of differences between the results of interpretation of satellite image of year 1992, 2000 and 2009. These satellite (SPOT) images are of size (6miles x 6miles) and are captured at summer season because it enhances the spectral reparability and brightness of the area under study. The right side of the Islamabad highway is developed in year 2000.
The recreational park is also developed in the Loi Bher vegetation provide the pollution free environment for the people. The patches of the Loi Bher forest, which also present between the sides of the judicial colony and Koran town shown in the image of year 1992, are diminished. The deforestation is observed by comparing both images.

Bahria town housing scheme has been developed along the Islamabad highway and its boundaries touches the Loi Bher forest.

Urbanization has some positive and negative impacts on environment but there is always a negative effect due to unplanned urban growth. Developed and underdeveloped countries are facing the same environmental problems due to urbanization.

The major problem with the city of Islamabad due to urbanization along Islamabad highway is the deforestation and the loss of habitat.

5.2 Case Study II

In this section, we present a 2nd case study to prove our proposed design methodology and its phases. This case study aims to detect land use changes between years 2005 and 2010, using satellite images of Land SAT 7.

To achieve this objective, digital topographic maps have been used to see the landuse/landcover changes in urban areas and identifying hotspots of land cover changes using multi-temporal satellite data and also studying relationship between human pressure on landuse/landcover and its impacts in the vital urban habitats

Study Area

The Villivakkam block (study area) of Thiruvallur district is located in between 13° 1’ 25” N to 13° 12’ 24” N Latitudes and 80° 1’ 40” E to 80° 11’ 30” E Longitudes.
Total area covered under study is 177 square kilometer and found in the west of Chennai. Villivakkam which is one of the most rapidly populated and changes regions of Thiruwallur district between the time interval 2005 and 2010.

5.2.1 Development of System at very first image (time $t_0$)

When we develop the system at very first image taken at time instance $t_0$, our methodology works in two phases:

i) Analysis Phase   ii) Design Phase

i) Analysis Phase

The analysis phase of our proposed methodology consists of two sub-phases:

i) Metadata Extraction/Annotation

ii) Layer Determination

Our analysis phase started with the satellite images having the following information:

Years of image acquisition: 2005-2010

Source of acquiring image: Land SAT 7

Location of satellite image: Thiruwallur district, covering area of 177 square kilometer in the west of Chennai.

Purpose of acquiring image: To develop and maintain a remote sensing application and change detection between time interval 2005-2010.

No. of satellite images: 2

Area covered by a satellite image: 177 sq. km
Table 5.7: Analysis report generated by the analysis phase of our proposed methodology

<table>
<thead>
<tr>
<th>Metadata Extraction/Annotation</th>
<th>Metadata1</th>
<th>Identification information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Metadata2</td>
<td>Metadata reference information</td>
</tr>
<tr>
<td></td>
<td>Metadata3</td>
<td>Spatial reference information</td>
</tr>
<tr>
<td></td>
<td>Metadata4</td>
<td>Entity and attribute information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer Determination</th>
<th>Type of Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>(The Layers which are needed by the stake holders)</td>
<td>Built-up</td>
</tr>
<tr>
<td></td>
<td>Agriculture land</td>
</tr>
<tr>
<td></td>
<td>Fallow land</td>
</tr>
<tr>
<td></td>
<td>Wasteland</td>
</tr>
<tr>
<td></td>
<td>Perennial water bodies</td>
</tr>
<tr>
<td></td>
<td>Dry water bodies</td>
</tr>
<tr>
<td></td>
<td>River/stream</td>
</tr>
<tr>
<td></td>
<td>Plantation</td>
</tr>
</tbody>
</table>

Table 5.7 provides information about the layers present in an image taken in 2005. Eight layers are identified at analysis phase and are input to the design phase for representation of their storage structure as hash table in terms of area and number of pixels for each layer.

There may be other layers too in an image but identification of these layers depends upon the functional requirements of the user/stake holders. These layers at this stage are labeled with layer prototypes (LP).
Figure 5.4 represents land SAT map generated from image taken in year 2005.

**Figure 5.4:** Land SAT 7 map of image taken in year 2005

**ii) Design Phase**

The design phase of our methodology takes input as analysis report in the form of layers which are identified at analysis level.

Table 5.8 is generated at design phase with number of pixels and area covered by each layer. These layers are stored by using a hash table storage structure.
### Table 5.8: Area under different layers in the image of year 2005

<table>
<thead>
<tr>
<th>Name of layers in the image taken in 2005</th>
<th>Area covered by a layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of area of each layer in the image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-up</td>
<td>65.1329</td>
<td>661238</td>
<td>36.99</td>
</tr>
<tr>
<td>Agriculture land</td>
<td>29.2117</td>
<td>302005</td>
<td>16.59</td>
</tr>
<tr>
<td>Fallow land</td>
<td>31.1963</td>
<td>327086</td>
<td>17.72</td>
</tr>
<tr>
<td>Wasteland</td>
<td>31.5077</td>
<td>324078</td>
<td>17.89</td>
</tr>
<tr>
<td>Industry</td>
<td>Nil</td>
<td>Nil</td>
<td>Nil</td>
</tr>
<tr>
<td>Perennial watrbdies</td>
<td>6.1853</td>
<td>62006</td>
<td>3.51</td>
</tr>
<tr>
<td>Dry waterbodies</td>
<td>9.8220</td>
<td>106432</td>
<td>5.58</td>
</tr>
<tr>
<td>River/streams</td>
<td>1.8572</td>
<td>20272</td>
<td>1.05</td>
</tr>
<tr>
<td>Plantation</td>
<td>1.1860</td>
<td>20101</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Table 5.8, represents eight layers in the image taken in 2005, as desired by the stake holder. Built-up layer covers most of the area under observation. Plantation is the least occurring area in the image taken in 2005.

This image is stored in the database in the form of layers by using hash table storage structure and can be used for comparison of subsequent images. Now we take image of year 2010 and this image is compared with already stored image in the database. This image is handled by incorporation and change detection module of our proposed design methodology as shown in Figure 3.19. Figure 5.5 represents the Land SAT map generated by satellite image taken in years 2010.
Figure 5.5: Land use map taken from satellite image of year 2010

Figure 5.5, represents nine layers with an additional layer of industry (causing structural change) which was absent in the previous image taken in 2005.

Table 5.9 shows the classification of each layer with area covered by layers in the image.

Table 5.9: Area under different layers in the image taken in 2010

<table>
<thead>
<tr>
<th>Name of layers in the image taken in 2010</th>
<th>Area covered by a layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of area of each layer in the image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-up</td>
<td>93.0097</td>
<td>942356</td>
<td>52.82</td>
</tr>
<tr>
<td>Agriculture land</td>
<td>24.8418</td>
<td>253830</td>
<td>14.11</td>
</tr>
<tr>
<td>Fallow land</td>
<td>31.48494</td>
<td>327086</td>
<td>12.9</td>
</tr>
<tr>
<td>Wasteland</td>
<td>22.7201</td>
<td>232512</td>
<td>12.9</td>
</tr>
<tr>
<td>Industry</td>
<td>1.5773</td>
<td>20034</td>
<td>0.9</td>
</tr>
<tr>
<td>Perennial water bodies</td>
<td>12.6342</td>
<td>132256</td>
<td>7.17</td>
</tr>
<tr>
<td>Dry water bodies</td>
<td>3.5207</td>
<td>40432</td>
<td>2.0</td>
</tr>
<tr>
<td>River/streams</td>
<td>1.8642</td>
<td>20543</td>
<td>1.06</td>
</tr>
<tr>
<td>Plantation</td>
<td>1.0842</td>
<td>24321</td>
<td>0.61</td>
</tr>
</tbody>
</table>
Table 5.9 represents a new layer, industry in the image taken in 2010. There have seen a decrease in agriculture land, fallow land wasteland and dry water body layers and increase in river/steam, perennial water bodies and built-up layers. Increase and decrease in layers cause data change whereas new layer addition causes structural change and handled by pre-analysis phase of our proposed design methodology as discussed in Chapter 3. New layer is input to the analysis phase for further metadata annotation and layer identification and design phase provides the design and storage structure.

5.3 Case Study III

We took a case study III of Lahore District with the collaboration of Punjab Government (Urban Unit Department).

Three years Land Sat TM images (1992, 1999 and 2003) have been taken for this purpose.

Total Area covered by each image is 1762 sq. km.

First image shown in Figure 5.6 (year 1992) is being used for application development at time $t_0$. 
The Analysis phase of our proposed design methodology works in two phases over this image given in Figure 5.6.

i) Metadata extraction/annotation

ii) Layers determination

We analyzed through our propose design methodology, the first image taken in 1992 at time instance $t_0$.

The analysis report generated by the analysis phase of our proposed design methodology is given in Table 5.10.
Table 5.10: Analysis report containing extracted metadata and determined layers

<table>
<thead>
<tr>
<th>Metadata Extraction/Annotation</th>
<th>Metadata1 Identification information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata2 Metadata reference information</td>
<td></td>
</tr>
<tr>
<td>Metadata3 Spatial reference information</td>
<td></td>
</tr>
<tr>
<td>Metadata4 Entity and attribute information</td>
<td></td>
</tr>
</tbody>
</table>

Layer Determination

(The Layers which are needed by the stake holders)

<table>
<thead>
<tr>
<th>Type of Layers</th>
<th>Built-up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vegetation</td>
</tr>
<tr>
<td></td>
<td>Soil</td>
</tr>
<tr>
<td></td>
<td>Water body</td>
</tr>
</tbody>
</table>

The analysis report given in Table 5.10 represents four layers; built-up, vegetation, soil and water body. This analysis report is passed to the design phase of our proposed methodology for determining the area covered by each layer and storage structure (Hash table) representation.

Table 5.11: Area covered (sq.km) by each layer with number of pixels

<table>
<thead>
<tr>
<th>Name of layers in the image taken in 1992 (1762 sq.km)</th>
<th>Area covered by a layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of area of each layer in the image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-up</td>
<td>302</td>
<td>3065296</td>
<td>17.14</td>
</tr>
<tr>
<td>Vegetation</td>
<td>668</td>
<td>6780193</td>
<td>37.91</td>
</tr>
<tr>
<td>Soil</td>
<td>767</td>
<td>7785042</td>
<td>43.53</td>
</tr>
<tr>
<td>Water bodies</td>
<td>25</td>
<td>253750</td>
<td>01.42</td>
</tr>
</tbody>
</table>
Table 5.11 shows that in year 1992, the most of area under district Lahore was covered with vegetation layer. The least covering area was by the water body layer (canal, river, lake etc.) while the built-up layer (buildings, houses, offices etc.) covers almost 37% of the total area under observation in district Lahore. This image is stored in the database as a hash table for subsequent comparisons with the upcoming images and to find out the changes that may occur after a certain period of time.

Figure 5.7 represents Landsat TM image taken in year 1999. This image (1999) is being passed to our proposed design methodology at time instance $t_1$. This image is handled by the pre-analysis phase of our proposed design methodology. The image (992) taken and stored in the database is to be compared with the upcoming image (1999) and find out the types of changes that may occur during time interval 1992-1999 in the district Lahore. Table 5.12 represents the number of layers in the image taken in 1999 and the respective area covered by each layer.
**Figure 5.7:** Landsat TM image taken in 1999 of district Lahore

**Table 5.12:** Landsat TM image layers covering area in sq.km in year 1999

<table>
<thead>
<tr>
<th>Name of layers in the image taken in 1999 (1762 sq.km)</th>
<th>Area covered by each layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of area of each layer in the image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-up</td>
<td>323</td>
<td>3278447</td>
<td>18.33</td>
</tr>
<tr>
<td>Vegetation</td>
<td>873</td>
<td>8860941</td>
<td>49.55</td>
</tr>
<tr>
<td>Soil</td>
<td>446</td>
<td>4526896</td>
<td>25.31</td>
</tr>
<tr>
<td>Water bodies</td>
<td>120</td>
<td>1217999</td>
<td>06.81</td>
</tr>
</tbody>
</table>

Table 5.12 represents some facts about the images taken in years 1992 and 1999.

Both images contain the same number of layers hence, there have seen, no structural change in the image taken in 1999. Only data change has been seen due to increase in built-up layer,
vegetation layer, water bodies layers and decrease in the soil layer. So, there is no need for this
image taken in year 1999 to be passed through the analysis and design phases of our proposed
methodology. This image is directly stored as a hash table storage structure in the database with
new a data. This can be done by the “Image Incorporation and Change Detection Module” of our
proposed design methodology.

Figure 5.8 represents Landsat TM image taken in year 2003. This image is passed to our
proposed design methodology at time instance \( t_2 \).

This image is processed by the “Image Incorporation and Change Detection Module” of our
propose design methodology by retrieving the latest image (1999) about district Lahore which
was previously stored in the database.
Figure 5.8: Landsat TM image taken in year 2003 of district Lahore

This image shows four layers and their respective area are given in Table 5.13

Table 5.13: Area covered by each layer in the image of year 2003

<table>
<thead>
<tr>
<th>Name of layers in the image taken in 2003</th>
<th>Area covered by a layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of area of each layer in the image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-up</td>
<td>374</td>
<td>3796096</td>
<td>21.23</td>
</tr>
<tr>
<td>Vegetation</td>
<td>1183</td>
<td>12007438</td>
<td>67.14</td>
</tr>
<tr>
<td>Soil</td>
<td>147</td>
<td>1492049</td>
<td>8.34</td>
</tr>
<tr>
<td>Water bodies</td>
<td>58</td>
<td>588699</td>
<td>03.30</td>
</tr>
</tbody>
</table>

This image is compared with the image taken in 1999. The results which we concluded depicts that there have seen no structural change while data change is present. Data change is there due
to increase in built-up and vegetation layers while corresponding decrease in soil and water bodies layers. So, this image is stored in the database as a temporal image.

Figure 5.14, Figure 5.15 and Figure 5.16 represents the hash table storage structure of images taken in 1992, 1999 and 2003 respective.
5.4 Storage Structure (Hash Table) for Images Taken for all three Case Studies

In this section we present a storage structure (hash tables) for case study I, case study II and case study III. The algorithmic representations for these storage structures are given in appendix II.

5.4.1 Storage Structure for Case Study I

**Figure 5.9:** Image taken in 1992 (Case study I) is stored as Hash Table

- **Vegetation**
  - Total No. of pixels: 671158
  - Storage Representation:
    - SR 2 6x10^3
    - SC 6x10^3
    - TNP

- **Soil**
  - Total No. of pixels: 915953
  - Storage Representation:
    - SR 40 2 8x10^3
    - SC 8x10^3
    - TNP

- **Railway Line**
  - Total No. of pixels: 206584
  - Storage Representation:
    - SR 10 6x10^3
    - SC 6x10^3
    - TNP

- **Water Body**
  - Total No. of pixels: 252488
  - Storage Representation:
    - SR 40 2 8x10^3
    - SC 8x10^3
    - TNP

- **Urban Area**
  - Total No. of pixels: 0
  - Storage Representation:
    - SR 0 2 6x10^3
    - SC 6x10^3
    - TNP

**Figure 5.9:** Image taken in 1992 (Case study I) is stored as Hash Table
<table>
<thead>
<tr>
<th>Vegetation</th>
<th>Total No. of pixels: 552,489</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>20</td>
<td>5000</td>
</tr>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>220</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Soil</th>
<th>Total No. of pixels: 956,476</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>30</td>
<td>4000</td>
</tr>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Railway Line</th>
<th>Total No. of pixels: 206,584</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>80</td>
<td>400</td>
</tr>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>100</td>
<td>4000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Water Body</th>
<th>Total No. of pixels: 235,832</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>70</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Urban Area</th>
<th>Total No. of pixels: 109,262</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>SR</td>
<td>SC</td>
</tr>
<tr>
<td>30</td>
<td>40</td>
</tr>
</tbody>
</table>

**Figure 5.10:** Image taken in 2000 (Case study I), stored as Hash Table
### Hash Table 3 \((t_2)\)

#### Vegetation
Total Number of Pixels: 552489

#### Soil
Total Number of Pixels: 956476

#### Railway Line
Total Number of Pixels: 206584

#### Water Body
Total Number of Pixels: 235832

#### Urban
Total Number of Pixels: 109262

---

**Figure 5.11:** Image taken in 2009 (Case study I), stored as Hash Table
5.4.2 Storage Structure for Case Study II

<table>
<thead>
<tr>
<th>Hash Table 4 ($t_0$)</th>
<th>Built-up</th>
<th>Total No. of pixels: 661238</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>SR</strong></td>
<td><strong>2</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Agricultural land</strong></td>
<td>Total No. of pixels: 302005</td>
</tr>
<tr>
<td></td>
<td><strong>SR</strong></td>
<td><strong>30</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Fallow land</strong></td>
<td>Total No. of pixels: 327086</td>
</tr>
<tr>
<td></td>
<td><strong>SR</strong></td>
<td><strong>50</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Wasteland</strong></td>
<td>Total No. of pixels: 324078</td>
</tr>
<tr>
<td></td>
<td><strong>SR</strong></td>
<td><strong>20</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Industry</strong></td>
<td>Total No. of pixels: nil</td>
</tr>
<tr>
<td></td>
<td><strong>Perennial waterbodies</strong></td>
<td>Total No. of pixels: 62006</td>
</tr>
<tr>
<td></td>
<td><strong>SR</strong></td>
<td><strong>30</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Dry waterbodies</strong></td>
<td>Total No. of pixels: 106432</td>
</tr>
<tr>
<td></td>
<td><strong>SR</strong></td>
<td><strong>10</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Rivers/streams</strong></td>
<td>Total No. of pixels: 20272</td>
</tr>
<tr>
<td></td>
<td><strong>SR</strong></td>
<td><strong>70</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Plantation</strong></td>
<td>Total No. of pixels: 20101</td>
</tr>
<tr>
<td></td>
<td><strong>SR</strong></td>
<td><strong>90</strong></td>
</tr>
</tbody>
</table>

**Figure 5.12:** Hash Table storage structure of image taken in 2005 (Case Study II)
Hash Table 5 ($t_1$)

- **Built-up** Total No. of pixels: 942356
- **Agricultural land** Total No. of pixels: 253830
- **Fallow land** Total No. of pixels: 327086
- **Wasteland** Total No. of pixels: 232512
- **Industry** Total No. of pixels: 20034
- **Perennial water bodies** Total No. of pixels: 132256
- **Dry water bodies** Total No. of pixels: 40432
- **Rivers/streams** Total No. of pixels: 20543
- **Plantation** Total No. of pixels: 24321

![Hash Table Storage Structure](image)

**Figure 5.13**: Hash Table storage structure of image taken in 2010 (Case Study II)
5.4.3 Storage Structure (images of years 1992, 1999 and 2003) for Case Study III

**Hash table** $(t_0)$

<table>
<thead>
<tr>
<th>Built-up</th>
<th>Total No. of pixels:</th>
<th>3065296</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vegetation</td>
<td>Total No. of pixels:</td>
<td>6780193</td>
</tr>
<tr>
<td>Soil</td>
<td>Total No. of pixels:</td>
<td>7785042</td>
</tr>
<tr>
<td>Water Body</td>
<td>Total No. of pixels:</td>
<td>253750</td>
</tr>
</tbody>
</table>

**Figure 5.14:** Image taken in 1992 (Case Study III) is stored as Hash Table
Figure 5.15: Image taken in 1999 (Case Study III) is stored as Hash Table
Figure 5.16: Image taken in 2003 (Case Study III), stored as Hash Table
As we have already mentioned that our proposed methodology works in two phases, i.e., i) Application development when there is no image in the database/application (First Phase); ii) and when the changes occurred to the application and they are incorporated in the application after its development (second phase). The proposed methodology suggests that all information and data (images) and changes to applications should be stored with time dimension (temporally) so that later they can be traced and retrieved temporally. For this purpose, images and changes to the applications are stored with time instances. The granularity of the time instance is decision of the designer which is done when a remote sensing application is being designed. Storing and maintaining all types of data temporally, is generally referred to as the history management, and it allow to retrieve important information and data that can be helpful in making essential and sensitive decisions. Another important feature of this methodology is that it allows to incorporate frequent changes occurred to the requirement specifications and its four different cases that have been identifies earlier.

Three case studies have been used to illustrate the working of the proposed methodology.

The first case study is taken from Islamabad highway, at Loi Bher forest. In this case study, there is no change or insignificant changes have been observed in the images of the Islamabad highway in the same year but different types of changes have been observed in the images taken in the later years.

The image that is taken in the year 1992 is being used in the initial development of the application at the time instance $t_0$ (First Phase), and the information and data that is extracted
about the layers present in the image is stored for the subsequent comparison use in the upcoming images. The images that are taken in the years 2000 and 2009 are used to incorporate the changes that occurred during the years 2000-2009. We have analyzed the three satellite images that are taken in three (3) years (1992, 2000 and 2009, respectively. In these images, we have detected three types of changes, i.e., Type I, Type III and Type IV (for details see Chapter 3), whereas, Type II is the type of change that occurred in the functional requirements of the application. Note that in this case study we have not considered this type of change to the application.

6.1 Results of Case Study I and their Analysis

In this case study, there are only four layers (vegetation layer, soil layer, water body layer and railway line layer) (see Chapter 5) in the first image taken in the year 1992 (see Figure 5.1). The pixels percentage of each layer in the image shows that vegetation layer, covers most of the land after soil layer (see Table 5.1). The Water body and railway line layer have the least number of pixels in the image taken in the year 1992 (see Table 5.1). This image is used for the comparison of the subsequent upcoming images. We get three base tables using the storage structure (Hash Table) given in Figure 5.9. The image taken in the year 1992 is used for the application development as the first image in the database of the application. Table 6.1 is used as a base table for the comparison with the base tables of the other images for the change detection purposes in the application.
Table 6.1: Results taken (using Hash Table) from Case study I of image taken in year 1992

Total area of the image taken = 203 sq.km

<table>
<thead>
<tr>
<th>No. of layers/Prototypes</th>
<th>Name of each layer in the image</th>
<th>Area covered by each layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of layer pixels in the image</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Vegetation</td>
<td>66.1240</td>
<td>671158</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>Soil</td>
<td>90.2417</td>
<td>915953</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>Railway line</td>
<td>20.3532</td>
<td>206584</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Water body</td>
<td>24.8757</td>
<td>252488</td>
<td>12</td>
</tr>
</tbody>
</table>

Similarly, we other base tables have been constructed using the hash table storage structure given in Figure 5.9 and Figure 5.10, and the tables are given in Table 6.2 and Table 6.3, respectively. The area covered by each layer and their corresponding number of pixels in the images taken in the years 2000 and 2009, as it has been observed through this case study are given in Table 6.2 and Table 6.3, respectively.
Table 6.2: Results extracted from the image taken in year 2000 through Case Study I

<table>
<thead>
<tr>
<th>No. of layers/prototypes</th>
<th>Name of each layer</th>
<th>Area covered by each layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of pixels of each layer in the image</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Vegetation</td>
<td>54.4325</td>
<td>552489</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Soil</td>
<td>94.2341</td>
<td>956476</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Railway line</td>
<td>20.3532</td>
<td>206584</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Water body</td>
<td>23.2347</td>
<td>235832</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Urban</td>
<td>10.7648</td>
<td>109262</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 6.3: Results taken (using hash table storage structure in Figure 5.10) through our Case Study I of the image taken in year 2009

<table>
<thead>
<tr>
<th>No. of layers/prototypes</th>
<th>Name of each layer</th>
<th>Area covered by each layer (sq.km)</th>
<th>No. of Pixels present in each layer</th>
<th>Percentage of pixels of each layer in the image</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Vegetation</td>
<td>41.7683</td>
<td>423948</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Soil</td>
<td>97.3421</td>
<td>988022</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>Railway line</td>
<td>20.3532</td>
<td>206584</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Water body</td>
<td>20.4532</td>
<td>207599</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Urban</td>
<td>22.3214</td>
<td>226562</td>
<td>11</td>
</tr>
</tbody>
</table>

We use the base tables, Table 6.1 and Table 6.2 for a comparative analysis between the images taken in the years 1992 and 2000. These tables can also be used for temporal change detection between different images as we can observe in Case Study I.
Table 6.4: Change detection between images taken in 1992-2000

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Area covered by each layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of pixels of each layer in the image</th>
<th>Change Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vegetation</td>
<td>66.1240</td>
<td>54.4325</td>
<td>671158</td>
<td>552489</td>
</tr>
<tr>
<td>Soil</td>
<td>90.2417</td>
<td>94.2341</td>
<td>915953</td>
<td>956476</td>
</tr>
<tr>
<td>Railway line</td>
<td>20.3532</td>
<td>20.3532</td>
<td>206584</td>
<td>206584</td>
</tr>
<tr>
<td>Water body</td>
<td>24.8757</td>
<td>23.2347</td>
<td>252488</td>
<td>235832</td>
</tr>
<tr>
<td>Urban</td>
<td>Nil</td>
<td>10.7648</td>
<td>Nil</td>
<td>109262</td>
</tr>
</tbody>
</table>

Overall Change is of Type IV (i.e., structural change and data change combined)

Table 6.4 shows a comparative analysis between the two images taken in the years 1992 and 2000. From this analysis it has been observed that the image taken in year 2000 contains an extra layer (urban), in addition to other four layers (see Table 6.4). This addition of the fifth layer (urban) nabs that a structural change has occurred in the area. This change causes depletion in the vegetation layer (which is a data change). Table 8.4 also reflects slight increase in the number of pixels of soil layer in the image taken in year 2000 as compared to the image taken in...
year 1992, and it causes a data change (Type I), whereas railway line and water body layer depicts no change in the area.

Now, we present a comparison between the two images taken in years 1992 and 2009 by using base Tables 6.1 and Table 6.3, and give the comparison in Table 6.5. This comparison is helpful in detecting the temporal changes that have been occurred during the time interval $[1992, 2009]$. 
Table 6.5: Change detection between images taken in 1992-2009

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Area covered by each layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of pixels of each layer in the image</th>
<th>Change Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vegetation</td>
<td>66.1240</td>
<td>41.7683</td>
<td>671158</td>
<td>423948</td>
</tr>
<tr>
<td>Soil</td>
<td>90.2417</td>
<td>97.3421</td>
<td>915953</td>
<td>988022</td>
</tr>
<tr>
<td>Railway line</td>
<td>20.3532</td>
<td>20.3532</td>
<td>206584</td>
<td>206584</td>
</tr>
<tr>
<td>Water body</td>
<td>24.8757</td>
<td>20.4532</td>
<td>252488</td>
<td>207599</td>
</tr>
<tr>
<td>Urban</td>
<td>Nil</td>
<td>22.3214</td>
<td>Nil</td>
<td>226562</td>
</tr>
</tbody>
</table>

Overall change in the images taken in 1992 and 2009 is of Type IV (structural and data change combined).

Table 6.5 shows a layer (*urban*) in the image taken in the year 2009, which is not present in the image taken in the year 1992. Hence, this addition of a new layer (*urban*) causes a structural change at the cost of decrease in the *vegetation layer* (see Table 6.5). Other observations from this comparison table are given as follows: i) there is a slight increase in the *soil layer* and slight
decrease in the water body layer, and this change occurred due to overhead bridge constructed on nullah for Behria Town as seen through this case study.

ii) These two types of changes that gas occurred (i.e., increase and decrease in the layers), causes data change (type I). Table 6.5 also shows that there is no change in the railway line layer. We analyze and compare the images taken in the years 2000, 2009 by using their respective base tables and their comparison that is given in Table 6.6.

This comparison table which is being extracted from the case study shows a relative temporal change within the time interval \([2000, 2009]\), by using base Table 6.2 and Table 6.3. Our proposed methodology is capable of detecting and incorporating these types of changes successfully (see Figure 3.19).
Table 6.6: Change detection between images taken in 2000-2009

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Area covered by each layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of pixels of each layer in the image</th>
<th>Change Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vegetation</td>
<td>54.4325</td>
<td>41.7683</td>
<td>552489</td>
<td>423948</td>
</tr>
<tr>
<td>Soil</td>
<td>94.2341</td>
<td>97.3421</td>
<td>956476</td>
<td>988022</td>
</tr>
<tr>
<td>Railway line</td>
<td>20.3532</td>
<td>20.3532</td>
<td>206584</td>
<td>206584</td>
</tr>
<tr>
<td>Water body</td>
<td>23.2347</td>
<td>20.4532</td>
<td>235832</td>
<td>207599</td>
</tr>
<tr>
<td>Urban</td>
<td>10.7648</td>
<td>22.3214</td>
<td>109262</td>
<td>226562</td>
</tr>
</tbody>
</table>

Over all change in the images taken in 2000 and 2009 is of Type I only (i.e., data change)

Table 6.6 shows that both the images taken in the years 2000 and 2009 have an equal number of layers, i.e., five so, no structural change has been observed. Only data change (Type I) has been observed due to increase in the urban layer and the corresponding decrease in the vegetation layer. A slight increase in the soil layer and decrease in the water body layer has also been seen.
(see Table 6.6). Our proposed methodology can incorporate and handle this type of change (data change) as mentioned in Case Study I.

We present a comparative analysis from our Case Study I given in Table 6.7.

**Table 6.7:** Trend of change in three year images taken in 1992, 2000 and 2009

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Area covered by each layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of pixels of each layer in the image</th>
<th>Trend of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vegetation</td>
<td>66.1240</td>
<td>54.4325</td>
<td>41.7683</td>
<td>671158</td>
</tr>
<tr>
<td>Soil</td>
<td>90.2417</td>
<td>94.2341</td>
<td>97.3421</td>
<td>915953</td>
</tr>
<tr>
<td>Railway line</td>
<td>20.3532</td>
<td>20.3532</td>
<td>20.3532</td>
<td>206584</td>
</tr>
<tr>
<td>Water body</td>
<td>24.8757</td>
<td>23.2347</td>
<td>20.4532</td>
<td>252488</td>
</tr>
<tr>
<td>Urban</td>
<td>Nil</td>
<td>10.7648</td>
<td>22.3214</td>
<td>Nil</td>
</tr>
</tbody>
</table>

The results of the case study are analyzed and summarized in Table 6.7 for all three images. They show a continuous decrease in the vegetation layer due to deforestation and continuous increase in the urban layer, and we refer to these phenomena as the urbanization. The urbanization has been observed in the case study due to construction of new residential colonies like Bahria Town, Doctors Town, Korang Town, Commoners Town, Pakistan Town, PWD colony, Site for Swan Garden, River Garden, Police Foundation, Jubilee Town, Sihala, CBR Housing schemes, Bankers Town, and Awan-e-Sadar Housing schemes. Table 6.7 also shows a
trend of slight but continuous increase in the soil layer and small decrease in the water body layer whereas the railway line layer remains unchanged.

The findings of this case study related to Loi Beher forest along the Islamabad highway, reveals that there is a continuous process of urbanization and its corresponding continues effect of deforestation.

Further, to analyze the types of changes that have occurred in different images during the time intervals, and the frequency of occurrence of the changes as they have been observed in the case study, we present them using different graphs. The graph shown in Figure 6.1 represents the number of pixels of each layer (area) in the corresponding image. This graph can help us for a better and quick understanding of the temporal changes that have occurred in the layers belonging to different years in the images (i.e., 1992, 2000 and 2009).

![Bar chart showing the number of pixels of each layer in different images](image)

**Figure 6.1:** Area covered by each layer in different years (1992, 2000 and 2009)

We have considered five (5) different layers (i.e., vegetation, soil, urban, water body and railway line) in our Case Study I. These five layers have different number of pixels in the images, taken
at different time instances (e.g., 1992, 2000, 2009), and they are shown in Figure 6.1. The image of the year 1992 has only four layers, and the *urban layer* as it is shown in Figure 6.1. The second image that is taken in the year 2000 has five layers, including an extra layer, i.e., *urban*. The *urban layer* has been introduced due to some construction which has been made using some area of the *vegetation* and *soil* layers. This depletion occurred in the *vegetation* and *soil layers*, as it has been shown in Figure 6.1. Therefore, the inclusion of a new layer (*urban*), causes Type III change (i.e., the structural change), and the depletion causes Type I change (i.e., the data change).

Now we present graphically in Figure 6.2 the trends of the types of change that have occurred in the case study.

**Figure 6.2:** Trend (frequency) of change (Type I, Type II, Type III and Type IV)
In the case study, we observe that at the early stage, the frequency of occurrences of Type I changes is more than any other change which increases gradually and becomes maximum at the time instance $t_3$, and then it decreases as the time passes. The Type II changes decrease gradually, the Type III change first increases, then it becomes almost constant and same is the case with the Type IV change as shown in Figure 6.2.

The Type I change occurs due to the cutting down of the vegetation layer and the Type III change occurs due to inclusion of a new layer (urban).

As the remote sensing application development completes, the reusability of the software becomes maximum, as shown in Figure 6.3.

Figure 6.3: Graph showing reusability trend
Reusability of software components a Time after application development/maintenance

Figure 6.3 shows the reusability trend of different components with the passage of time as the systems grow. The reusability of the components has minimum value in the beginning but it grows over the time, and this trend is shown in Figure 6.3.

6.2 Results of Case Study II and their Analysis

We have used Case Study II in Chapter 5 to illustrate the working of the proposed methodology. As we have mentioned earlier that this case study belongs to the region of Thiruvallur District India, in the West Chennai.

In this case study, there are two images which are taken at the time instances 2005 and 2010, respectively.

As we know already from Chapter 5 that the first image is taken in the year 2005, it has eight (8) layers as they are shown in Table 5.8. This table shows that the built-up layer covers most of the area in the image while the plantation layer covers the smallest area among all layers in the image. The Industry layer is not present in the first image but it is the desire of the users/stack holders. The image is stored by using the hash table storage structure and shown in Figure 5.12.

We have constructed Table 5.8 as the base table in Chapter 5 for the image shown in Figure 5.4. On the arrival of the second image at the time instance 2010, as our proposed design methodology compares the newly arrived image with the previously stored image in the data store of the application. This comparison is done by the image incorporation and change detection modules of the proposed design methodology.

The base table, Table 5.9, is also built for the second image that is taken at the time instance 2010 (see Case Study II in Chapter 5).
In this section, we give an analysis of the second case using the base tables, Table 5.8 and Table 5.9.

The comparison of both tables is carried out by image incorporation and change detection modules of the proposed methodology (see Chapter 3).

**Table 6.8:** Change detection in images (layers) during (2005-2010)

<table>
<thead>
<tr>
<th>Name of layer/prototype</th>
<th>Area covered by a layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>% of area of each layer in the image</th>
<th>Difference %</th>
<th>Change type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>2010</td>
<td>2005</td>
<td>2010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Built-up</td>
<td>65.1329</td>
<td>93.0097</td>
<td>661238</td>
<td>942356</td>
<td>+15.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>36.99</td>
<td>52.82</td>
<td>Data change</td>
</tr>
<tr>
<td>Agriculture land</td>
<td>29.2117</td>
<td>24.8418</td>
<td>302005</td>
<td>253830</td>
<td>-2.48</td>
</tr>
<tr>
<td>Fallow land</td>
<td>31.1963</td>
<td>31.484944</td>
<td>327086</td>
<td>327086</td>
<td>-9.29</td>
</tr>
<tr>
<td>Wasteland</td>
<td>31.5077</td>
<td>22.7201</td>
<td>324078</td>
<td>232512</td>
<td>-4.82</td>
</tr>
<tr>
<td>Industry</td>
<td>Nil</td>
<td>1.5773</td>
<td>Nil</td>
<td>20034</td>
<td>+1</td>
</tr>
<tr>
<td>Perennial water bodies</td>
<td>6.1853</td>
<td>12.6342</td>
<td>62006</td>
<td>132256</td>
<td>+4.02</td>
</tr>
<tr>
<td>Dry water bodies</td>
<td>9.8220</td>
<td>3.5207</td>
<td>106432</td>
<td>40432</td>
<td>-3.58</td>
</tr>
<tr>
<td>River/streams</td>
<td>1.8572</td>
<td>1.8642</td>
<td>20272</td>
<td>20543</td>
<td>+0.01</td>
</tr>
<tr>
<td>Plantation</td>
<td>1.1860</td>
<td>1.0842</td>
<td>20101</td>
<td>24321</td>
<td>-0.06</td>
</tr>
</tbody>
</table>

*Note:* Overall Change is of Type IV (Combination of data change and structural change)

Table 6.8 shows that the built-up layer occupies 36.99% of the total area in the first image and 52.82% area in the second image. The agricultural land, fellow land and wasteland layers have been decreased by 2.48%, 9.29% and 4.82%, respectively. From these statistics, it can be
concluded that the people have utilized the agricultural land, waste land and fallow land for their residential and other building construction purposes. This type of change (increase/decrease in area of a layer(s)), it means that a data change (see Chapter 3) has occurred in the region during the time interval; [2005, 2010]. This change has occurred due to use of the agricultural land for the purpose of urbanization. Table 6.8 also shows the appearance of a new layer, i.e., industry, in the second image. In other words, a structural change has occurred and detected in the second image. This table also shows slight seasonal changes in dry water bodies, rivers and plantation, which are the data changes.

We present graphically the two images and their layers, in Figure 6.4.

Figure 6.4: Graph showing change in layers during time interval 2005-2010

Figure 6.4 shows that there is a remarkable increase in the built-up layer and a new layer, Industry, has also appeared in the second image. There are slight variations in all layers, either they have increased or decreased, as shown in Table 6.8.
6.3 Results of Case Study III and their Analysis

Case Study III belongs to the district Lahore, covering an area of 1762 square kilometers.

In this case study, three are three satellite (Landsat TM) images, which are taken at the time instances 1992, 1999 and 2003.

As we already know from Chapter 5 that the first image taken in the year 1992, has four (4) layers as shown in Table 5.11. This table shows that the soil layer (43.53 %) covers most of the area of the image while the vegetation layer covers the 2nd largest area under observation following the built-up layer. The water body layer has the smallest area just covering 01.42 % of the total area of the image. The image (of the year 1992) is stored by using the hash table storage structure and shown in Figure 5.14. We have constructed the base table, Table 5.11, in Chapter 5 for the image shown in Figure 5.6.

On the arrival of the second image at the time instance 1999, our proposed methodology suggests to compare the newly arrived image with the previous stored image in the data store of the application. This comparison is done by the image incorporation and change detection modules of the proposed methodology.

The base table, Table 5.12, is also constructed for the second image (shown in Figure 5.7) that is taken at the time instance 1999 (see Case Study III in Chapter 5).

The base Table 5.12 reflects that vegetation layer covers most of the area (49.55 %) and soil layer covers 2nd largest area (25.31 %) in the image.

The third image is taken at the time instance 2003 shown in Figure 5.8 and its base table, Table 5.13, is constructed by using the proposed methodology given in Case Study III (see Chapter 5).
The results from Table 5.13 show that the vegetation layer covers most part of area of the image while the built-up layer covers the 2nd largest area under consideration (i.e., of the district Lahore).

In this section, we give an analysis of these three images of the Case Study III, by using the base tables, Tables 5.11, 5.12 and 5.13 respectively.

First, we compare the results of Tables 5.11 and 5.12. This comparison is carried out by the image incorporation and change detection modules of the proposed methodology.

**Table 6.9:** Change detection in images (layers) during (1992-1999)

<table>
<thead>
<tr>
<th>Name of layer/prototype</th>
<th>Area covered by a layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>% of area of each layer in the image</th>
<th>Difference %</th>
<th>Change type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-up</td>
<td>302 323</td>
<td>3065296 3278447</td>
<td>17.14 18.33</td>
<td>+01.19</td>
<td>Data change</td>
</tr>
<tr>
<td>Vegetation</td>
<td>668 873</td>
<td>6780193 8860941</td>
<td>37.91 49.55</td>
<td>+11.64</td>
<td>Data change</td>
</tr>
<tr>
<td>Soil</td>
<td>767 446</td>
<td>7785042 4526896</td>
<td>43.53 25.31</td>
<td>-18.22</td>
<td>Data change</td>
</tr>
<tr>
<td>Water body</td>
<td>25 120</td>
<td>253750 1217999</td>
<td>01.42 06.81</td>
<td>+05.39</td>
<td>Data change</td>
</tr>
</tbody>
</table>

*Note: Overall Change is of Type (data change only)*

Table 6.9 shows that the built-up layer occupies 17.14 % of the total area in the first image (1992) and 18.33% area in the second image (1999). So, 01.91 % area of built-up layer has been increased during the time interval [1992, 1999], and causing a data change. Similarly, the vegetation layer has increased by 11.64 % during this interval and the water body layer has increased by 05.39 %. The only layer in which decrease has occurred during this interval is the soil layer, and the decrease is 18.22 % as shown in Table 6.9.
From these statistics, it can be concluded that people have utilized the soil or waste land for residential and agricultural purposes during the time interval [1992, 1999]. This type of change (increase/decrease in some layer(s)) means that a data change (see Chapter 3) has occurred in the region during the time interval. This table also shows slight seasonal changes in water body, which has also caused data changes. The above mentioned changes can also be seen graphically in Figure 6.5.

![Figure 6.5: Graph showing change in layers during time interval 1992-1999](image)

Figure 6.5 shows that there is a remarkable increase in the vegetation layer and decrease in the soil layer. There are slight variations in the built-up (increase) and water body layers (increase).

Now, we present a comparison between images taken in the years 1992 and 2003.

Table 6.10 reflects changes in the images (layers) during the time interval [1992, 2003].
Table 6.10: Change detection in images (layers) during (1992-2003)

<table>
<thead>
<tr>
<th>Name of layer/prototype</th>
<th>Area covered by a layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>% of area of each layer in the image</th>
<th>Difference %</th>
<th>Change type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-up</td>
<td>302</td>
<td>374</td>
<td>3065296</td>
<td>3796096</td>
<td>17.14</td>
</tr>
<tr>
<td>Vegetation</td>
<td>668</td>
<td>1183</td>
<td>6780193</td>
<td>12007438</td>
<td>37.91</td>
</tr>
<tr>
<td>Soil</td>
<td>767</td>
<td>147</td>
<td>7785042</td>
<td>1492049</td>
<td>43.53</td>
</tr>
<tr>
<td>Water body</td>
<td>25</td>
<td>58</td>
<td>253750</td>
<td>588699</td>
<td>01.42</td>
</tr>
</tbody>
</table>

Note: Overall Change is of Type (data change only)

Table 6.10 shows that the built-up layer occupies 17.14 % of the total area in the first image (1992) and 21.23% area in the second image (2003). It means that 04.09 % area of the built-up layer has been increased during the time interval [1992, 2003], and causing a data change. Similarly, the vegetation layer has increased 29.23 % during this time interval and the water body layer by 01.88 %. The only layer which causes drastic decrease during this interval is the soil layer, having decrease of 35.19 % as shown in Table 6.10.

From these comparisons, it can be concluded that the soil or waste land has been utilized for the agricultural/green belts or form land purposes during the time interval [1992, 2003]. This type of change (increase/decrease in some layer(s)) means that a data change (see Chapter 3) has occurred in the region during the time interval; [1992, 2003]. This table also shows slight seasonal changes in water body, which also causes data changes.

The change detection as discussed in Table 6.10 can also be presented graphically in Figure 6.6.
Figure 6.6: Graph showing change in layers during time interval 1992-2003

Figure 6.6 shows that there is a remarkable increase in the vegetation layer and decrease in the soil layer. There are slight variations in the built-up (increase) and water body layers (increase).

In this section we give a comparison between the images taken in the years 1999 and 2003.

Table 6.11 shows the comparative analysis between these two images taken during time interval; [1999, 2003].
Table 6.11: Change detection in images (layers) during (1999-2003)

<table>
<thead>
<tr>
<th>Name of layer/prototype</th>
<th>Area covered by a layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>% of area of each layer in the image</th>
<th>Difference</th>
<th>Change type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1999</td>
<td>2003</td>
<td>1999</td>
<td>2003</td>
<td>%</td>
</tr>
<tr>
<td>Built-up</td>
<td>323</td>
<td>374</td>
<td>3278447</td>
<td>3796096</td>
<td>18.33</td>
</tr>
<tr>
<td>Vegetation</td>
<td>873</td>
<td>1183</td>
<td>8860941</td>
<td>12007438</td>
<td>49.55</td>
</tr>
<tr>
<td>Soil</td>
<td>446</td>
<td>147</td>
<td>4526896</td>
<td>1492049</td>
<td>25.31</td>
</tr>
<tr>
<td>Water body</td>
<td>120</td>
<td>58</td>
<td>1217999</td>
<td>588699</td>
<td>06.81</td>
</tr>
</tbody>
</table>

*Note: Overall Change is of Type (data change only)*

Table 6.11 shows that the *built-up layer* occupies 18.23 % of the total area in the first image (1999) and 21.23% area in the second image (2003). It means that the 02.90 % area of the *built-up layer* has been increased during the interval [1999, 2003], thus causing a data change. Similarly, the *vegetation layer* has increased 17.59 % during this time interval and the *soil, water body layers* have decreased by 16.97 % and 03.51, respectively.

From these comparisons, it can be concluded that people have utilized the *soil* or *waste land* for agricultural/green belts or form land purposes during the time interval. This type of change (increase/decrease in some layer(s)) means that a data change (see Chapter 3) has occurred in the region during the time interval; [1999, 2003]. This table also shows slight seasonal changes (decrease) in the *water body*, which also causes data changes.

We present graphically, for better overview of changes mentioned in Table 6.11 in Figure 6.7.
Figure 6.7: Graph showing change in layers during time interval 1999-2003

Figure 6.7 shows that there is a remarkable increase in the vegetation layer and decrease in the soil layer. There are slight variations in the built-up (increase) and water body layers (decrease).

We present a comparative analysis of Case Study III in Table 6.12. The table shows the change trend over time (rate of change) over all three years.
**Table 6.12:** Trend of change in three year images taken in 1992, 2000 and 2009

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Area covered by each layer in the image (sq.km)</th>
<th>No. of Pixels of each layer present in the image</th>
<th>Percentage of pixels of each layer in the image</th>
<th>Trend of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-up</td>
<td>302</td>
<td>323</td>
<td>374</td>
<td>3065296</td>
</tr>
<tr>
<td>Vegetation</td>
<td>668</td>
<td>873</td>
<td>1183</td>
<td>6780193</td>
</tr>
<tr>
<td>Soil</td>
<td>767</td>
<td>446</td>
<td>147</td>
<td>7785042</td>
</tr>
<tr>
<td>Water body</td>
<td>25</td>
<td>120</td>
<td>58</td>
<td>253750</td>
</tr>
</tbody>
</table>

Table 6.12 gives a comparative analysis of three images taken in three different years. The **Built-up** and **vegetation layers** are increasing continuously over the time while the **soil layer** is decreasing continuously. Table 6.12 shows this trend of change as the over the time.

We can also observe this change trend in Figure 6.8.
Figure 6.8: Trend of change in layers during time interval 1992-1999-2003

Figure 6.8 shows that the built-up layer increases gradually, it means that housing facilities and commercial building are constructed more over the time interval; at the cost of soil or waste land. Similarly, the vegetation layer increases rapidly and the soil layer decreases at the same pace.

In the next section, we give analysis of all three case studies.

6.4 Analysis of Three Case Studies

All case studies belong to the different physical locations and time instances. They have different satellite spot images containing different number of layers. Therefore, we argue that the proposed methodology is capable to handle and incorporate spot images of different types having different number of layers. The number and types of layers in these three case studies depend upon the functional requirements of the users/stack holders at the time of requirement analysis. For example, if user requirements are to find out the transportation layer which is at an abstract level, then at a granularity level this layer may further be categorized into sub-layers, i.e.,
railway line, roads, river, sea etc. As we know that we have successfully illustrated and validated the proposed methodology using three different case studies as it has been given in Chapter 5. In these case studies, we have observed that all types of changes mentioned in Chapter 3, i.e., data change, changes in functional requirement, structural change and combination of both data change and structural changes have been identified, detected and incorporated successfully by the proposed methodology. We have also provided the hash table as a storage structure to store the image layers and other related information. The functional requirements which cause the structural change are costly to handle since in this type of change a new layer is analyzed, designed and incorporated as a new artifact in an application. From the three case studies, we can conclude that less effort is needed if only a data change is handled as compared to handle a structural change.

6.5 Discussion

In this section we summarize results of the results and analysis of three (3) case studies, analysis of the working of our proposed methodology, and difficulties that we have faced and lesson learned from this research work.

6.5.1 About Achievement of Goals and Objectives

In the previous sections, we have analyzed the working of our proposed methodology using the three case studies by keeping in mind the goals and objectives that were set in Chapter 1 and Chapter 2 for the proposed methodology for remote sensing (RS) applications. One of the objective that was fixed, is that the proposed methodology should able to develop remote sensing applications from scratch and later its main focus should be on maintenance of the developed RS
applications. In other words, the proposed methodology should able to do maintenance after the
development of RS applications.

Through the successful illustrations of the proposed methodology using the three case studies we
are able to show the achievement of this objective. It should be noted that the proposed
methodology is capable to handle any number and types of images and layers.

The proposed methodology detects and incorporated (algorithms are given in Appendix II) the
four types of changes (see Chapter 3) in a uniform manner. The operators cover most of the users
information retrieval requirements, and they are formally proposed in Chapter 4. Algorithms are
also given for these retrieval operators. The case studies that are used in this dissertation are
carefully chosen to include all salient features and requirements of RS applications.

The proposed methodology is capable to handle any number of layers in an image of a RS
application. A separate phase of the methodology is responsible to identify number and types of
layer in a new image of the application. Since images of a RS application are stored and retrieve
temporally, therefore, the retrieved information can be used in making important and crucial
decisions about the locations. Those decisions can save cost and effort.

6.5.2 Problems and Difficulties Faced

The main problem that we faced during the research work is the selection and acquisition of
satellite data (spot images). The organizations and agencies dealing with satellite data are
reluctant to provide satellite images due to security and some other unknown reasons. In this type
of research work, case studies play a vital role. Therefore, availability of comprehensive case
studies should be ensured in advance. The case studies that we are able to get are two to three (2
– 3) images, which are just stuffiest but not enough to illustrate this research work. We could
hardly find at maximum three (3) images for each case study.
In our opinion if we were able to get more images for each case study, then we are in better position to illustrate the working of our proposed methodology in more elaborative way. As a consequence, a better analysis could be done.

6.5.3 Lesson Learned

During this research we have learned a lesson that before starting this kind of research (a specialized area like remote sensing area) case studies should be arranged well in time and advance. We have spent almost six (6) months to arrange satellite images (as a case study) of the district Lahore for Case Study III.
Chapter 7

Conclusion and Future Directions

In this chapter, we give our concluding remarks about this research work and possible future research directions in which this work can be extended.

7.1 Conclusion

In this dissertation, we have proposed a software development methodology for remote sensing application domain. This application domain has different and peculiar characteristics that make the development of this application domain different from other application domains, and they also make the existing development methodologies inappropriate to be used for the development of applications of the RS domain. Among these characteristics, different types of changes (such as data change, structural change and combination of these changes) to the applications that occur after their development make them difficult to handle and incorporate. The proposed development methodology works in two different steps, that is, the first case is when RS applications are being designed and developed from scratch (the first time when RS applications are developed from scratch). The second step deals when different types of changes occur to the applications after their development. In other words, the second situation/step is actually the maintenance of already developed RS applications. Since maintenance of already developed software/applications especially remote sensing applications is generally a time-consuming and costly task, therefore, the main contribution in this dissertation is handling and incorporation of all types of changes to already developed RS applications. The second step of the proposed methodology identifies different types of changes that occur to already developed remote sensing applications, their incorporation in the developed applications, and later on their tracing. Note that we store all types of changes with time dimension, and later on they can be retrieved, traced
and manipulated the stored changes temporally. For this purpose we have also defined a set retrieval operators and written their algorithms. These retrieval operators cover the functional requirements of RS applications including tracing and retrieving the history of the recoded changes.

The working of both steps of the proposed methodology is also written in the algorithms, which is a step towards automation of the proposed methodology. We have illustrated the working of the proposed methodology by using three (3) case studies; complete working and results of the case studies are given and analyzed, which shows successful working of the proposed methodology.

These three case studies are three different types of applications, and the proposed development methodology have successfully identified different types of layers in these three case studies and incorporated them in the already developed applications. This shows the maintenance capability and flexibility the proposed methodology.

7.2 Future Directions

We are actively working on the automation of the proposed development methodology. As we have already mentioned that a set of algorithms has been developed for both steps of the methodology, and working of the methodology has already been tested through three case studies. The proposed methodology is the first attempt to develop and use a formal methodology for the devolvement of remote sensing applications, in the future this methodology can further be improved from the experience that will be learned by using the methodology for more case studies.

The replication and provenance techniques have been used successfully to handle the problems of data dissemination in the web-based data intensive domains such as e-commerce and
bioinformatics. However, these techniques have not been adapted by the mainstream remote sensing community. With the increasing reliance of E-systems on remote sensing data and remote sensing applications, we feel that replication and provenance techniques must be adapted by remote sensing community if it is to handle the data dissemination issues and problems. Therefore, further work needs to be done to improve these techniques to make them more practical.


Beach, CA, USA, References Cited: 56, PP (558-567).


Pseudo code to capture the four types of Change

Data structure which is used in the following algorithm.

C_Type: array [1, 2, 3, 4] of change type;

Existing_object: array [1, 2, 3 ……n] of fresh object list

Procedure incorp_of_change (C_Type)
{
   Proc_0
   Similarity_Check(new object)
   IF Similarity= 100%
   TEHN
   c_type = type I
   Proc_1
   ELSE IF
   c_type = type II
   TEHN
   Proc_2
   ELSE I
   c_type = type III
   TEHN
   Proc_3
   ELSE
   c_type = type IV
Proc_4 (type IV)

END IF

}

Proc_1
{
C_type=Type1_Change
design_phase()
implementation_phase()
Incorporate_data_change()
}

Proc_2
{
C_type=typeII_change
If functional_requirements=c_type
AND
    C_type=TypeI_change
THEN
    Proc_1
ELSE IF
Functional_requirements=c_type
AND
    C_type=typeIII_change
THEN
    Proc_3
Proc_analysis_phase()
Incorporate_functional_requirement()
Existing_object[......]= new object
END IF

Proc_3
{
C_type=typeIII_change()
Proc_Pre_Analysis_phase()
Proc_Analysis_phase()
Proc_design_phase()
Proc_Implementation_phase
Incorporate_Structural_change
Existing_object[......]=new object
}

Proc_4
{
C_type=TypeIV_change
Proc_1
Proc_2
Proc_3
Existing_object[......]=new object
}
APPENDIX-II

Algorithmic Representation

The algorithm given in below, transforms an image which is represented by an $n \times n$ array matrix of integers (pencils) into an ADT open hash table data structure. Note that bucket size of the hash table is five (5). Each bucket of the hash table is linked to a link-list of nodes, and the fields of each node are given as follows:

Class Node

{
    Int SR ; /* Starting Row of an element (SR) of an $n \times n$ array matrix representing an image */
    Int SC ; /* Starting Column of an element(SC) */
    Int TNP ; /* Total Number of pixels(TNP) indentified for each layer */
    Node *link ; /* A Link field to that points to the next node */
}

Figure 1: Data Structure of 'Node'

FUNCTION Build_Layer_Table(IM): H_Table;

/* 'IM' is an image as input that is represented by an $n \times n$ matrix */
{
    Node *H_Table[5]; /* An array of node Type pointers for hash table with total number of five (5) buckets */
    FOR each IM(i,j) ;
WHILE \( i \leq n \) DO

Where \( j \leq n \) DO

IF IM(i,j) = Vegetation THEN

{ 
SR.Node= n1 ; /* starting row number */
SC.Node=n2 ; /* starting column number */
TNP.Node=Integer ; /* TNP(Total number of Pixel) will Get its value from the function Build_Hashtable_IM(i,j) using CASE statement for each layer */
Link to the Vegetation bucket of the hash table
}

ELSE IF 

IM(i,j) = Waterbody THEN

{ 
SR.Node= n1 ; /* starting row number */
SC.Node=n2 ; /* starting column number */
TNP.Node =Integer; /* Total number of Pixels(TNP) for each Layer */
Link to the node to the Waterbody Bucket of the hash table
}

ELSE IF 

IM(i,j) = Soil THEN

{ 
SR.Node= n1 ; /* starting row number */

SC.Node=n2 ; /* starting column number */

TNP.Node=Integer; /* Total number of Pixels (TNP) for Each Layer */

Link to the Soil Bucket

}

ELSE IF

    IM(i,j)= Railway Line THEN

{

SR.Node= n1 ; /* starting row number */

SC.Node=n2 ; /* starting column number */

TNP.Node=Integer; /* Total number of Pixels (TNP) for Each layer */

Link to the Railway Line Bucket

}

ELSE IF

    IM(i,j)= Urban Area THEN

{

SR.Node= n1 ; /* starting Row Number */

SC.Node=n2 ; /* starting Column Number */

TNP.Node=Integer; /* Total No. of Pixels (TNP) for each Layer */

Link to Urban Area in the Bucket

}

RETURN H_Table

    /* Build Hashtable for Vegetation */
Figure 2: Algorithm for Layers Construction

The time complexity of this algorithm is $n^2$

The following algorithm builds the hash table by inserting nodes to each of the five (5) layers. Each layer is built on the basis of classification algorithm. Note that pixels of an image are classified into five different layers (i.e., Vegetation, Soil, Water Body, Railway Line and Urban Area) through the classical knowledge of object recognition and classification (Guage, et.al, 2007; Rajeshwari, et al, 2008).

**FUNCTION** Classification (pixel(i,j)):String

{
    IF ColorOfPixel(i,j)== Green
    RETURN “Vegetation”
    IF ColorOfPixel(i,j)== Blue
    RETURN “Water Body”
    IF ColorOfPixel(i,j)== Brown
    RETURN “Soil”
    IF ColorOfPixel(i,j)== Black
    RETURN “Railway Line”
    IF ColorOfPixel(i,j)== Pink
    RETURN “Urban Area”
}

Figure 3: Algorithm for Classification

**FUNCTION** Build_Hashtable (IM(i,j), H_Table)
{ 
String Layer;

FOR each row ‘i’ of an image 
{
    FOR each column j of image represented by an n Xn array matrix
    {
        Layer = Classification(pixel(i,j))
        SWITCH (Layer)
        {
            CASE "Vegetation"
            TNP=0
            SR=i
            SC=j
            WHILE (Classification(IM(i,j))=="Vegetation"
            {
                TNP++
                j++
            }
            CALL InsertNodeatEnd(SR,SC, TNP, H_Table[0])
            CASE "Water body"
            TNP=0
            SR=i
            SC=j
        }
    }
}
\textbf{WHILE} (Classification(IM(i,j))=="Water body")
{
    TNP++
    j++
    \textbf{CALL} InsertNodeAtEnd(SR,SC, TNP, H_Table[1])
    \textbf{CASE} "Soil"
    TNP=0
    SR=i
    SC=j
    \textbf{WHILE} (Classification(IM(i,j))=="Soil")
    {
        TNP++
        j++
        \textbf{CALL} InsertNodeAtEnd(SR,SC, TNP, H_Table[2])
    }\textbf{CASE} "Railway Line"
    TNP=0
    SR=i
    SC=j
    \textbf{WHILE} (Classification(IM(I,j))=="Railway Line")
    {
        TNP++
        j++
        \textbf{CALL} InsertNodeAtEnd(SR,SC, TNP, H_Table[3])
    }\textbf{CASE} "Urban Area"
NOP=0
SR=i
SC=j

while (Classification(IM(i,j)) == "Urban Area")
{
  TNP++
  j++
  call InsertNodeatEnd(SR, SC, TNP, H_Table[4])
}
/* end of SWITCH */
} /* end of inner FOR loop */

j=0
i++
} /* end of outer FOR loop */

Figure 4: Algorithm for Building Hash Table

The time complexity of this algorithm (Figure 4), is $n^2$

function InsertNodeatEnd (SR, SC, TNP, H_Table[i])

Pointer=Create _Node(SR, SC, TNP)

Pointer2= H_Table[i]

if H_Table[i]==NULL

H_Table[i]= Pointer

else

while (H_Table[i]->link !=NULL)

Pointer2 = pointer2->link

208
Figure 5: Algorithm for Node Insertion

*The time complexity of this algorithm is $n^2$*

The following pseudo code detects the data change that may occur in the 2\textsuperscript{nd} image taken after some time.

\begin{verbatim}
FUNCTION  Change_Detection(H_Table1,H_Table2):Int
/* Detect whether total number of nodes in two hash tables are the same or Not */
# Node1=0 , # Node2=0
FOR each row of H_Table1 AND H_Table2
  Pointer1= H_Table1[i]
Poinetr2= H_Table2[i]
  WHILE  (Pointer1->link !=NULL)
    Pointer1=Pointer1->link
    #Node1++
  WHILE  (Pointer2-> link !=NULL)
    Pointer2=Pointer2->link
    #Node2++

RETURN  # Node1, # Node2
\end{verbatim}
The following pseudo code determines whether data stored in Nodes of two hash tables is same or not.

```plaintext
FOR each row of H_Table1 and H_Table2

Pointer3=H_Table1[i]
Pointer4=H_Table2[i]

WHILE (Poinetr3 ->Link!= NULL AND Poinetr4->Link!=NULL)

IF (Pointer3->SC!=Pointer4->SC OR Pointer3->SR=Pointer4->SR OR Pointer3->TNP!=Pointer4->TNP)

RETURN 1

Pointer3=Pointer3->Link
Pointer4=Pointer4->Link

}
```
The time complexity of this algorithm is \( n^2 \)

**Figure 6:** Algorithm for Data Change Detection

The following pseudo code detects the structural change that may occur in the 2\(^{nd}\) image taken after some time.

**FUNCTION** DetectStructuralChange(H_Table1, H_Table2): Int

\[
i = 0
\]

**FOR** each row of H_Table1 AND H_Table2

**IF** (H_Table1[i] -> link == NULL AND H_Table2[i] -> link1 != NULL)

\[
i++
\]

**RETURN** 1

The time complexity of this algorithm is linear

**Figure 7:** Algorithm for Structure Change Detection

The following pseudo code displays change (data change or structural change) that may be detected above

**FUNCTION** DetectChange() 

**IF** (DetectStructuralChange(H_Table1, H_Table2) OR DetectDataChange(H_Table1, H_Table2))

Display Change Detected
ELSE

Display  No Change

Figure 8: Algorithm for display of Change Detection

The time complexity of this algorithm is linear