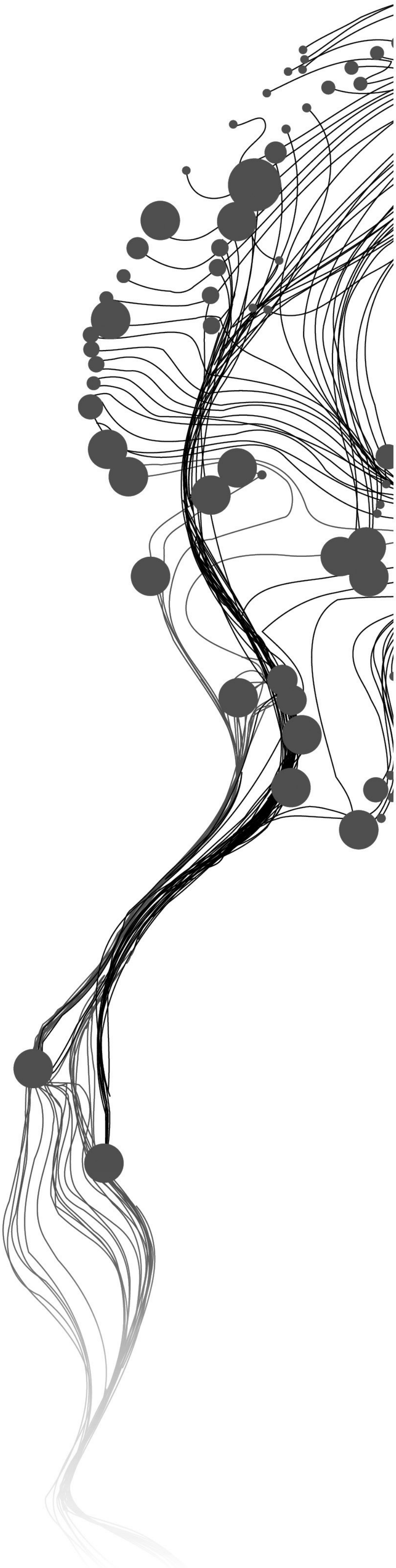
MATCHING RASTER AND TRAJECTORY DATA USING WEB SERVICES

SHASHISH MAHARJAN
February, 2011

SUPERVISORS:

First Supervisor: Dr. U. D. Turdukulov
Second Supervisor: Ms Dr. C. A. Blok



MATCHING RASTER AND TRAJECTORY DATA USING WEB SERVICES

SHASHISH MAHARJAN
Enschede, The Netherlands, February, 2011

Thesis submitted to the Faculty of Geo-information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:

First Supervisor: Dr. U. D. Turdukulov
Second Supervisor: Ms Dr. C. A. Blok

THESIS ASSESSMENT BOARD:

Dr. Ir. R. A. de By(chair)
Dipl. Geoinf. E. H. Jürrens

ABSTRACT

Moving objects are studied in different domains of science and technology. The movement of objects is influenced by various factors like environmental changes, influence of other agents, intrinsic properties, and spatial constraints. Researchers tend to explore relationship between moving objects and these influencing factors. They use conventional techniques to match trajectory data with external environmental data that affects objects' movement. These methods consume time and resources. The development in web technology, new standards and protocols have added new scope in this process. This study focused on investigating tools and methods to match trajectory data with raster data using web services. Two different cases: iceberg movement and animal movement were studied. Iceberg movement can be affected by external environmental factors like sea surface temperature and ocean wind. Similarly change in vegetation can affect movement behavior of grazing animals. These environmental data are distributed in multidimensional grid format on the internet by different organizations using different standards. GISciecne community uses Open Geospatial Consortium (OGC) Web Coverage Service (WCS) for data access and Catalogue Service for Web (CSW) for searching metadata. Similarly, Earth Science (ES) community have developed set of protocols like Open-source Project for a Network Data Access Protocol (OPeNDAP) for data access and Thematic Realtime Environmental Distributed Data Services (THREDDS) catalogue for finding datasets. These standards can be utilized to find appropriate data sources, extract required subset of data. Later the trajectory data can be matched with extracted subset of data to find associated values. The client is required to access WCS and OPeNDAP servers. On the basis of the predefined criteria like version support, spatial and temporal subsetting, inteprolation support, suitable clients, OWSLib and Pydap were selected.

The three tier architecture was chosen to develop a prototype web service. Users can interact with a client application to upload trajectory data, select suitable data sources and an appropriate interpolation method. The request sent by users was handled by python based middleware that was built on the top of the apache server. OWSLib and Pydap modules were used to connect and access data from the WCS server and OPeNDAP server respectively. After a subset of data was accessed from server, associated grid values were extracted and matched with trajectory data. For example iceberg trajectory data was matched with sea surface temperature and wind direction. In the case of missing values in the grid data, an interpolation method was applied. The final result was saved as CSV file format that can be later used for further analysis and visualization purpose. Also the performance of prototype was evaluated on the basis of time consumption for the processing. The web service was implemented as a proof of concept that will help researchers to explore possible relationship between change in environment and movement.

Keywords

data matching, trajectory, raster data, OGC WCS, OPeNDAP, web service

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank all individuals and organizations who contributed for completion of my MSc studies.

I would like to acknowledge the scholarship from government of the Netherlands under Netherlands Fellowship Program (NFP) and University of Twente, Faculty of Geo-Information Science and Earth Observation (ITC) for awarding me admission and academic skills.

In particular, I would like to express my gratitude to my first supervisor Dr. Ulanbek Turdukulov. I thank him for the great guidance, discussion and encouragement during the entire research period. I would also like to extend my gratitude for my second supervisor Dr. Connie Blok for her valuable suggestions and feedback that helped a lot to improve this research. I would like to express my sincere appreciation to Dr. Ir. Rolf de By for his comments and suggestion during my proposal and mid term defenses.

I want to express my thanks to Dr. Paola Petrelli from TPAC for answering my queries regarding servers and supporting me during difficulty accessing server. I also like to appreciate OWSLib and Pydap e-community for sharing experiences handling these modules.

I am grateful to all colleagues at the Department of Geoinformatics, in particular Mr. Uma Shankar Panday for fruitful discussions and support during my study. I also like to thank all fellow Nepalese friends for making the stay comfortable and memorable.

Finally, my deepest gratitude towards my parents and brother for their continuous support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1
Introduction

1.1 MOTIVATION AND PROBLEM STATEMENT

The world is dynamic. Many objects change their positions in the geographic space over time. This change in position has been studied in different domains of science and technology like biology, meteorology and transportation engineering (Dodge et al., 2008). The development in satellite network, in miniaturizing transmitters, and developments in cheap location based devices have made research on moving objects easier. Biologists implant mini transmitters in animals' body and trace their movement to study its impact on ecosystem. Whereas meteorologists study different natural phenomenon like movement of icebergs, storms using satellite technology to make people aware about their consequences. In this way large amounts of data is collected in different domains.

Moving objects create many trajectories during their life time. These trajectories are collected by different applications for different purposes (Spaccapietra et al., 2008). The major parameters of trajectory data are coordinates of the object, time, speed, direction of movement, accumulated distance (Andrienko et al., 2008). Based on these parameters data are analyzed to discover patterns like most frequent stops, most frequent moves in certain region and many other patterns (Alvares et al., 2007). Although these patterns help in studying behavior of moving objects, reasons of movement cannot be investigated solely from trajectory data since their movement is influenced by several factors (Dodge et al., 2008). The questions like what causes objects to move, why they move or stop can be explored only after studying those influencing factors.

Dodge et al. (2008) have categorized these influencing factors in four groups. They are: intrinsic properties, spatial constraints, environment and influence of other agents. These factors vary according to study area and type of moving object. The movement of icebergs is affected by environmental factors like ocean currents; animals might migrate due to change in vegetation; movement of vehicles is affected by heavy rainfall. This information regarding external factors have been collected using different satellites and data are made available in different formats. The matching of trajectory data with data about external factors help researcher to find relationship between moving objects and influencing factors. In different domains of science this kind of matching is being conducted. Beck et al. (2008) matched traces of giant panda with National Oceanic Atmospheric Administration (NOAA) NDVI data to explore altitudinal movement of these mammals during different seasons. Likewise traces of icebergs are matched with sea surface temperature, wind, current etc to study calving effects (Laine, 2008). Manually researchers download grid data, georeference them, match each trajectory record to a corresponding image cell, perform overlay, extract cell values for given location and time and compose values into resulting table (Turdukulov et al., 2010). This process consumes large amount of time and resources. Also, this approach becomes difficult for large dynamically changing data.

Development in web technology, protocols and standards has revealed new hopes to make this matching faster and easier. The use of web services can significantly decrease volume of data, computing steps and resources required (Di & McDonald, 1999). The satellite data providers have already adopted different standards to disseminate data using web services. Open-source Project for Network Data Access Protocol (OPeNDAP) is one of the standards being used by Earth Science community for sharing remote sensing data (OPeNDAP, 2004). Similarly, Open Geospatial Consortium (OGC) provides specifications like Web Coverage Service (WCS) for accessing geospatial coverages including satellite imageries, digital elevation data from remote servers (OGC, 2008). NOAA provides huge amount of NetCDF data that can be accessed using OPeNDAP and WCS protocol (NOAA, 2010). Similarly, National Snow and Ice Data Center (NSIDC) provide raster data in GeoTIFF format using WCS (NSIDC, 2010). The trend of using WCS is growing since it is interoperable, based on new standards and can support multidisciplinary users (Di et al., 2003; Nativi & Domenico, 2009). Most of the environmental data available on the internet are generated from observations or simulation results. Web based mechanisms can be built that would find appropriate data sources from the internet using both WCS and OPeNDAP standards, extract required data, match them with trajectory data and finally disseminate them through web portals to diverse users.

Finding required raster data on the web is not easy task due to lack of proper metadata and catalogues. Extracting small fraction of data from huge dataset that match spatio-temporal resolution of trajectory data is also challenging task. Moreover matching extracted raster data with the multi-temporal, multi-spatial trajectory data hiding technical complexity for the user is also a key issue to be dealt with.

In addition, matching of these data sets may lead to many computational and technical challenges (Coyne & Godley, 2005). The challenge starts form the data level, input trajectory data might be in different formats, grid cells might contain missing values. These kinds of inconsistency and uncertainty might affect the quality of matched datasets and can mislead interpretation of phenomenon. The problem may arise due to differences in temporal resolution between trajectory data and raster data. In many cases trajectory data are collected at a certain time interval like icebergs trajectory data are normally collected at an interval of one to five days, but data like temperature, ocean currents are collected daily. Hence interpolation techniques are essential during implementation of trajectory and raster data matching. Besides interpolating position of the moving objects, coverage data also need to be interpolated as it lacks values due to weather situations while collecting data.

1.2 RESEARCH IDENTIFICATION

1.2.1 Research objectives

There are two main objectives of this research, they are:

- To investigate methods and tools for creating web service that matches trajectory data with raster data available as WCS and OPeNDAP.
- To implement and evaluate the performance of the prototype.

The following specific objectives are devised from main objectives:

1. To set up different use cases for understanding user requirements.
2. To find and access data and metadata available through WCS and OPeNDAP.
3. To identify different methods to match trajectory and raster data while maintaining quality of service.
4. To identify a suitable dissemination method.

5. To identify an appropriate framework for matching data sets.
6. To implement web service for data matching.
7. To set up experiments for evaluating performance of designed prototype.

1.2.2 Research questions

On the basis of research objectives, the following research questions are formulated that are to be answered during the study period:

1. Who are the users and what are their requirements?
2. What are data and metadata options that are available through WCS and OPeNDAP?
3. What are WCS and OPeNDAP clients?
4. How to maintain quality of data matching services?
   - How to match metadata information?
   - How to match data with different spatio-temporal resolutions?
   - How to measure uncertainty in matched datasets?
5. How to disseminate the outcome of matching?
6. What framework should be adopted for matching services?
7. How to evaluate performance of designed prototype?

1.2.3 Innovation aimed at

The research aim is to provide a web service that will match trajectory data and raster data available through WCS and OPeNDAP.

1.2.4 Related work

Matching trajectory data with continuous data like environmental data temperature, wind, and vegetation is not new. This is being conducted in different field of study since long. However, conventional methods are used during the process. Data is downloaded from a FTP server or received in compact discs, transformed into a required projection, data are overlaid with raster images using certain parameters and result is disseminated in the form of image or tables. A similar approach was followed by (Ipopo, 2009) to match iceberg trajectory data with Sea Surface Temperature (SST). Coyne & Godley (2005) have highlighted a web based Satellite Analysis and Tracking Tool (STAT) which is used to access traces of animals like sea tortoise and merge the trajectory data with raster imageries like NOAA chlorophyll and SST to monitor behavior of sea tortoise. STAT uses FTP to obtain environmental data from data servers at NOAA, NASA, Collecte Localisation Satellites (CLS) and the Global Ocean Data Assimilation Experiment (GODAE).

Some work has been done however to access servers, download required grid data, match data using server technology and downloading data from a server for visualization by (Ha, 2010). The OPeNDAP protocol was used to connect to an external server, Integrated Data Viewer (IDV) and python script was used to match data on the server. The process was automatic but several issues like multiple client access and interoperability of web service were not dealt. Also performance was not optimized.

Geo-spatial data are shared by different communities using own standards. Geo-interface to Atmosphere, Land, Environment, Ocean netCDF (GALEON) was initiated to bridge gap between different standards (Domenico et al., 2007). The project implemented WCS gateway that enable THREDDS Data server (TDS) to respond WCS clients. That made possible to access large datasets available in ES community using WCS clients.

In most of the applications OPeNDAP protocol has been used to access the server, subsetting the required information from the large data source as most of the data providers follow OPeNDAP. However the scenario is changing since many organizations like NASA, NOAA and enterprises like ESRI are following WCS as a common standard to share raster data over the internet. Therefore an web application is required to access those servers following WCS standard as well as OPeNDAP, match datasets and also focus on user accessibility and computational time.

1.3 PROJECT SET-UP

1.3.1 Method adopted

The research project will be started by studying use cases of iceberg movement and animal movement to find user and user requirements for studying movements and factors affecting their movement. Then required data and metadata will be identified and also available web servers will be assessed. Different WCS clients will be evaluated based on several criteria like extensibility, data support, interpolation possibilities etc. Moreover methods for extracting data will be identified.

The measures to be taken for quality assurance of a web service, like provision of metadata for selecting the best web service, different interpolation techniques will be identified. In addition, effective methods will be identified in this phase to measure uncertainty.

The framework of matching will be identified and implemented. Basically there will be three major components in the web based framework. The first component will be client side application with a graphical user interface that will take trajectory data as input and allow users to select an appropriate web server providing raster data. The client side interface will also let user to choose interpolation methods for finding missing values in grid cells. The second component will be middleware that will be built on python and run on a web server. The role of middleware will be to fetch input from users and access data from servers. In addition it will match trajectory data with raster data, extract values from grid cells, perform interpolation. The matched data set including information about uncertainty will be returned to the client side user interface through the middle ware. Finally experiments will be set up to evaluate performance of prototype. The methods adopted is shown in figure 1.1.

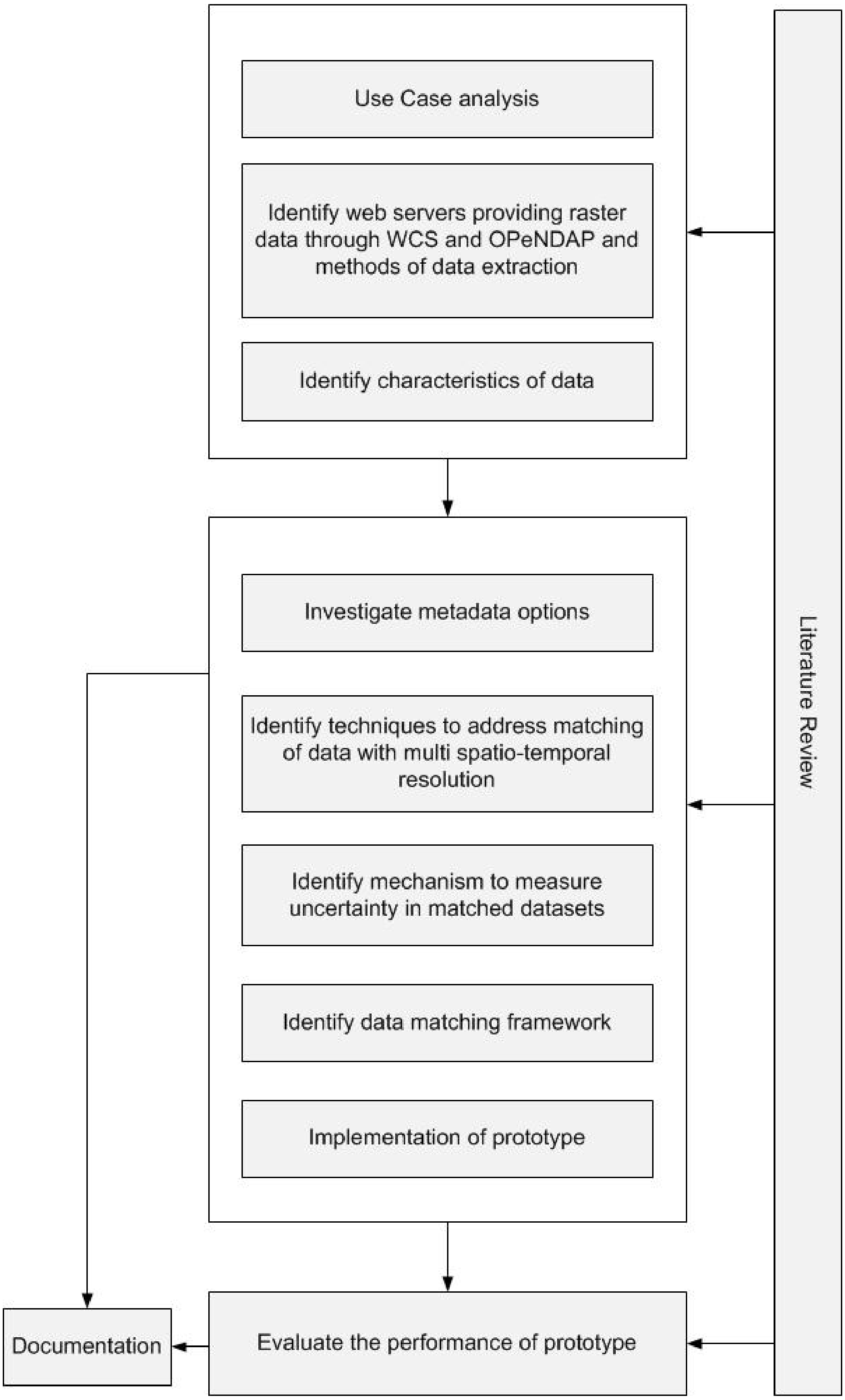


Figure 1.1: Flow chart of methods

1.4 ORGANIZATION OF THESIS

The remaining chapters of this thesis are organized as follows:

Chapter 2: In this chapter the concept of trajectory will be highlighted. Users of data matching and user's requirements will also be identified in this chapter. Moreover interpolation methods, and uncertainty will also be discussed.

Chapter 3: In this chapter web services and protocols from different communities used for accessing environmental data will be introduced. Different WCS clients will also be evaluated and selected in order to access data.

Chapter 4: The design and implementation of the prototype will be discussed in this chapter. The architectural design, work flow as well as user interface will be described. Also the performance evaluation of the prototype will be described in this chapter.

Chapter 5: The final chapter describes achievements of the study, conclusions, discussions and recommendations for future study.

Chapter 2

Need of trajectory data matching with raster data and user requirements

Moving objects around the world are monitored using cutting edge technologies. The collected data are analyzed and studied in various domains of science and technology.

This chapter gives insight on the concept of trajectory. As use cases, icebergs and animal movement will highlight the influence of environmental factors affecting the movement. In addition this chapter focuses on matching of trajectory and raster data and on the issues that need to be dealt with during the process: interpolation and uncertainty associated with it.

2.1 CONCEPT OF TRAJECTORY

Trajectories are dynamic: objects change their positions on geographical space during their lifetime. The path made by a moving object in space over a certain period of time is a trajectory (Andrienko et al., 2008). According to Spaccapietra et al. (2008), "a trajectory is the user defined record of evolution of position of an object that is moving in the space during a given time interval in order to achieve a given goal". Or:

trajectory $[t_{begin}, t_{end}] \rightarrow$ space

If t_{begin} is the time when an object starts moving and t_{end} is the time when an object stops; it occupies some positions in space during a certain interval of time t_{i}. So, a trajectory can be represented as a function that matches time moments with positions in space (Andrienko et al., 2008). During the lifetime of a moving object, its trajectory may be semantically segmented by defining a temporal sequence of time sub-intervals where the object position changes (moves) and stays fixed (stops) (Spaccapietra et al., 2008). Hence an object may travel a number of trajectories as a sequence of moves going from one stop to the next one (see figure 2.1).

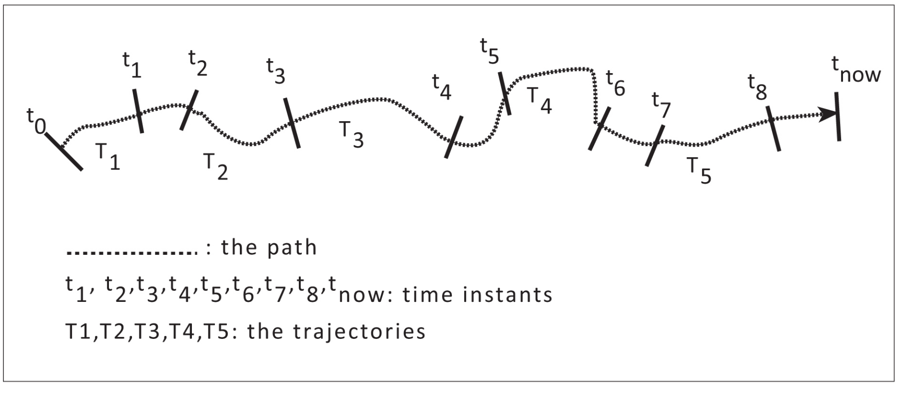


Figure 2.1: Trajectory definition as spatio-temporal path (Spaccapietra et al., 2008)

2.1.1 Characteristics of trajectories

A trajectory in its lifespan (t_{begin}, t_{i}) or its subset (t_{i}, $t_{\text{i+1}}$) may poses several characteristics. Those characteristics are defined and analyzed according to the need of the application. (Andrienko et al., 2008) has grouped characteristics into two categories. Movement related characteristics include:

- Time
- Position in space
- Direction of movement
- Speed of movement
- Change of the direction
- Change of the speed
- Accumulated travel time and distance

The overall trajectory characteristics are as follows:

- Geometric shape of the trajectory in space
- Traveled distance
- Duration of trajectory in time
- Movement vector
- Mean, median and maximal speed
- Dynamics of the speed
  - Periods of constant speed, acceleration, deceleration and stillness
  - Characteristics of these periods : start and end times, duration, initial and final positions, initial and final speeds, etc
  - Arrangement (order) of these periods in time
- Dynamics of the directions
  - Periods of straight, curvilinear, circular movement
  - Characteristics of these periods: start and end times, initial and final positions and directions, major direction, angles and radii of the curves, etc
  - Major turns ('turning points') with their characteristics: time, position, angle, initial and final directions, and speed of the movement in the moment of the turn
  - Arrangement (order) of the periods and turning points in time

The study of characteristics of trajectories can provide answers to different queries about moving objects. For example what is the position of an iceberg at a particular time? and how long will it take to reach a certain location? In addition patterns made by trajectories can also be studied using these characteristics like most frequently visited places, locations where moving objects spent more time. These patterns and characteristics collected from the trajectories can be analyzed to explore the behavior of the moving objects. However, questions like why a moving object suddenly changes its direction, why it stops or moves cannot be answered by only studying these characteristics. Thus while studying trajectories, factors that influence movement also should be considered and analyzed. Dodge et al. (2008) has categorized these influencing factors in four groups they are;

- Intrinsic property: It is an unique property of each moving object that influences the movement. For example : maximum speed, acceleration.
- Spatial constraint: It can stop or in some cases reduce the movement of the objects. It can be network barriers.
- Influence of other agents: It is another influencing factor that is more prominent in animal movement, where the movement of one animal can influence movement of other animals.
- Environment: It can affect movement of any objects. The sudden change in climate like heavy snowfall or forest fires can stop the movement of people. The change in vegetation or change in chlorophyll content in ocean motivate animals to migrate from one location to another.

It is clear that movement of objects depends upon various factors. Movement of animals may depend on their intrinsic factors, influence of other animals as well. These kinds of factors are difficult to grasp. The aim of this research is to demonstrate the matching between trajectory data and environmental data can be done using web services, and not to explain movement by matching trajectory data to all possible influencing factors.

The relationship between environmental factors and moving objects are made more clear in the description of use cases below.

2.2 USE CASES

2.2.1 Iceberg movement: users and users' requirements

Icebergs are created due to calving of glaciers. After formation they float into the sea and move around the world's ocean until they melt or are grounded for longer periods. The movement of icebergs and their behavior has been studied in different areas of science and engineering as it highly impacts the environment as well as navigation and engineering. Different models are used to predict iceberg movement behavior. For example, the International Ice Patrol (IIP) uses a drift model developed by Mountain (1980), with an iceberg deterioration model made by Anderson (1983).

These days icebergs are studied since they are one of the major indicators of climate change. The iceberg melting patterns are coupled with global oceanic and atmospheric cycles in order to study global warming (Stuart & Long, 2009). Oceanographers follow icebergs since they discharge cold freshwater to the sea that can influence currents and ocean circulation. Biologists study icebergs to explore how they influence ocean life. As icebergs melt, nutrients get mixed into the ocean around them that teems with plankton, fish, and other sea life (Arrigo et al., 2002). Navigators and route planners study icebergs data to prevent possible collision with sea vessels. An organization like IIP monitors movement of icebergs for safe navigation of ships. Petroleum industries also consider iceberg drift as risk to the petroleum installations (Fuglem et al., 1996). The users can hence be grouped into two categories:

- Navigators, Engineers
- Scientists like climatologist, oceanographers and biologists

Oceanographers and Climatologists may need answers to the following questions:

- What is the effect of environmental factors on iceberg movement?

Navigators and engineers may need answer to the following questions:

- What is the current position of a particular iceberg?
- How long does an iceberg stay at a particular location?
- In which direction will iceberg be moving and what will be the speed?

Users need up to date data about iceberg trajectories as well as weather/environmental data in order to address the above mentioned questions. The location, speed and directions of icebergs are important characteristics that are derived from the collected data. In addition to these data, factors influencing the movements are also collected to predict future direction, and location of icebergs. The direction of iceberg's movement is determined by ocean current, whereas wind moderates speed. The shape and size of iceberg also get changed as they move. According to Death et al. (2006), wind stress, ocean current and sea surface temperature are most dominant forces that influence iceberg movement. The matching techniques are required to analyze the relationship between iceberg movement and influencing factors.

Data collection

The National Ice Center (NIC) is the only authorized organization to track and name icebergs. NIC uses satellites enabled with scatterometer and radio meter equipment to keep track of icebergs larger than 10nm (nautical miles). NIC collects iceberg data in the interval of 15 to 20 days. The frequency of data collection is low due to the complexity of data collection techniques and also due to the low speed of iceberg movement. Brigham Young University (BYU) collects iceberg data at an interval of 1 to 5 days. The advantage of high temporal resolution is ability to see the change in motion due to environmental factors like ocean currents.

The data collected by both organizations are available on the internet and can be used by different organizations for their own purposes.

Data characteristics

Iceberg data from the year 1976 can be downloaded from the websites `http://www.scp.byu.edu/data/iceberg/database1.html` and `http://www.natice.noaa.gov/products/south_icebergs.html` in Microsoft Excel file format.

The attributes of available data are:

- ID of iceberg
- Date of data acquisition
- Position
- Size of iceberg
- Area
- Name of satellite used to collect data

On the basis of above attributes some other attributes can be derived like:

- Speed of an iceberg
- Acceleration i.e. change in speed
- Orientation
- Traveled distance by a particular iceberg
- Lifespan of an iceberg from its origin to its latest observation, etc.

2.2.2 Animal Movement: users and users requirements

Animals move from one to another habitat in their life span. Animals like leopards run several miles per hour to get their pray. Some species of fish like salmon travel thousands of miles from sea to fresh water. Many bird species migrate seasonally.

The movement of animals is basically for their survival i.e. for shelter, food, protection from predators and reproduction. The frequency of move can be annual or seasonal like some animals shift their habitat each year in dry and wet seasons while searching for food and water. In many cases it is nomadic, for example grazing animals like zebra and buffalo keep on moving to green pastures as they finish eating. This movement plays a crucial role in the ecosystem as every living creature being on earth is part of the food chain and dependent on each other.

Although the movement of animals is natural and part of ecology, their movement is affected by some external factors that can be natural like rainfall, drought, rapid change in vegetation or human induced, like deforestation, fire, pollution. In modeling of animal movement, some particular factors are considered. For example change in vegetation and its effect on behavior of animals are studied in many researches. NDVI is mostly used with species distribution. Boone et al. (2006) used NDVI based vegetation growth estimates to understand wildebeest migration. Similarly, Ryan et al. (2006) revealed that African buffalo reduced its short term ranging distance with increased values of NDVI. High temporal resolution of NDVI can be helpful in studying animal movement as it can be linked to location data of individuals at same time (Pettorelli et al., 2011).

Several questions regarding animal behavior can be asked by studying the relationship between the changing environment and animal movement. As mentioned in section2.1.1 there may be several factors affecting movement, and all factors cannot be modeled. So we limit the number of factors and related questions in this study. The users can be landscape ecologists and conservationists. They may need answers to following questions:

- What is current location of animal?
- What causes animals to move or stop?
- What is the effect of change in vegetation on movement of animals?

Data collection

Tracking of animal movement was a manual process in earlier years, animals were traced by inspection from vehicles or aircrafts. The tracking became easier by the use of radio transmitter attached to the animal's body but still radio collared animals need to be followed. The development of the ARGOS system was a mile stone in tracking animals using satellites. Still there are errors and accuracy issues.

The recent advancement in animal tracking technology have resulted in large volumes of data. The use of Global Positioning System (GPS) technology allows to record the sequential movement behavior of animals and compile datasets with many observations compared to older tracking techniques (Jonsen et al., 2005). The combination of GPS and GSM technology has made the tracking process faster and easier. This technology uses the cellular network to transmit positional data collected by GPS at regular intervals of time. The data is loaded to a server, and can be downloaded using web pages. When an animal is out of the GSM network, data is stored on the collar and later transmitted to the server, once the animals move to the network areas.

Data characteristics

There are two different kinds of animal trajectories available for buffalo and zebra. The data is collected twice a day at 8 AM and 4 PM for two months in the wet season and for one month in the dry season.

The attributes of the available data are:

- Date and time (fixed time, start time, end time) of data acquisition
- Position: site of the animal in each time stamp in geographical coordinates as latitude and longitude.
- Name of animal

2.3 DATA MATCHING

The case study descriptions showed that the movement of icebergs and animals are affected by external environmental factors. The study of trajectories is incomplete without the study of the influencing environmental factors. The matching of these two types of data provides new dimensions in exploring the behavior of moving objects. The matching of trajectory data with ocean currents, wind direction etc will ease the researchers to predict the location of the iceberg at a certain time. Likewise matching environmental data with animal movement data will help researchers to provide information about relationship between change in environment and movement.

Location and time (x, y, t) of given trajectory records can be matched with grid cells and associate values can be extracted. The focus of the study is on supporting users to explore possible relationships between external factors and trajectory data, to enable them to further analyze the data and perhaps make prediction based on found relationships. It certainly is not the intension to enable full integration and modeling of all factors that might influence object movement.

The matching of data from different sources with different spatio-temporal resolution and different degree of uncertainty is a complicated process. Interpolation techniques need to be applied to match spatio-temporal resolution of both datasets as well as to calculate values for missing data. The trajectory data collected may have positional uncertainty whereas raster data may have attribute uncertainty. The matching of these data with different uncertainties may give rise to another level of uncertainty that may mislead decision making processes. The aspects related to interpolation and uncertainty are discussed in next sections.

2.4 DATA INTERPOLATION

The movement of the objects can be traced using location based devices with GPS that can store many positions each second. Not every record can be transmitted and stored due to limited technical and economic resources. Similarly environmental variables that change most frequently cannot be recorded as it happens. So interpolation might be required to find values between recorded datasets. Not only to fill the gaps in each data sets, but also to match the spatio-temporal resolution of the different data sets, interpolation methods should be applied. However in this study we limit interpolation for estimating missing grid values only.

The matching coverage data is searched from the web server i.e. grid data of the same or higher spatial and temporal resolution than trajectory data. Afterwards associated values for positions (lat,lon,time) are extracted. Often grid data can have missing values due to the cloud cover for instance. The associated value for a particular position of trajectory data may not be available for a specific time. In these situation, missing values can be estimated using interpolation methods.

The interpolation technique required to estimate value of unobserved positions based on available points on space. There are several interpolation methods available and many of them depend on the application specific domain. It is not the scope of this thesis to overview of all possible methods, therefore we limit oursleves to one of the common interpolation technique - Inverse Distance Weighing methods.

2.4.1 Inverse Distance Weighing (IDW)

IDW interpolation is based on assumption that points near to each other have similar values. In IDW, the known values that are close to prediction location will have more influence on the predicted value than those farther away. So during calculation, points near to the neighborhood area are given high weights and points at far distance are given small weights.

According to Johnston et al. (2001), IDW can be performed using the following formula:

$$w(x,y) = \sum_{i=1}^{N} \lambda_i w_i, \lambda_i = \frac{\left(\frac{1}{d_1}\right)^p}{\sum_{k=1}^{N}\left(\frac{1}{d_k}\right)^p}$$

Where $w(x,y)$ is the predicted values at location (x,y), N is the number of nearest known points surrounding (x,y), λ_i are the weights assigned to each known point value w_i at location (x_i,y_i), d_i are the Euclidean distance between each (x_i,y_i) and (x,y) and p is exponent, which influences the weighting of w_i on w.

IDW is a simple method that is easy to program (Henley, 1981). In addition, this method is applicable to datasets of small size as well (Rhodes & Myers, 1993). This study does not aim to give in depth insight into interpolation methods. So only IDW was chosen to estimate missing values in grid cells.

2.5 UNCERTAINTY

Uncertainty is natural; it exists everywhere and is studied in several domains of science. Uncertainty is differently defined in various domains. Shi (2009) has discussed four factors that cause uncertainty in the GIScience domain. Those four factors are: inherent uncertainty in the real world, limited knowledge of human for perceiving the real world, technical limitations of measurement devices for acquisition of data, potential of generating and propagating uncertainty during processing and analysis.

Spatial data goes through different stages like collection, processing, transformation, integration with heterogeneous data and analysis before it is used for decision making process. During these various processes uncertainty is introduced, accumulated and propagated (Wang et al., 2005). The errors and uncertainties vary spatially and temporally. Traditional methods provide global accuracy assessment and uncertainty analysis that are not sufficient for making detailed management plans and decision making processes. "Spatially identifying the sources of uncertainties, modeling their accumulation and propagation, and finally quantifying them is critical to control the quality of spatial data" (Wang et al., 2005).

Spatial data preprocessing includes data cleaning, data transformation and data discretization in which many uncertainties are produced. The transformation process is one of the key sources of uncertainty as it changes the data from one form into another. Interpolation, done in order to estimate missing values also causes uncertainty. "Uncertainties can be eliminated by uncertainty handling techniques but never completely, even some new uncertainties will be produced in the handling process due to impropriety of the techniques" (He et al., 2005).

In this research project focus will be given to the error in phase of data interpolation.

Uncertainty due to interpolation

In order to fill gaps and estimate missing values interpolation method described in section 2.4 can be applied. The data are interpolated for the missing values but the interpolation methods are not perfect. The values generated from interpolation methods are only approximations not a real value. So each estimated value is linked with some degree of uncertainty. Users of system should be informed about degree of uncertainty associated with estimated data.

2.6 CONCLUSION

This chapter highlighted basic concepts of trajectory data and the need of matching trajectory data with environmental data. The consequences of matching of trajectory data and environmental factors needed to deal with, like interpolation and uncertainty, were also discussed. The case studies of iceberg and animal movement clarified the effects of external factors on movement of objects. The user requirements were described in the form of questions that will be answered in later chapters.

Chapter 3

Web services for accessing environmental data

3.1 INTRODUCTION

The phenomenon on the earth is monitored by two major scientific communities. They are GIScience community and Earth Science (ES) community. These communities have their own data models for representing the same geographic phenomenon. GIScience community collects datasets as static features with accurate earth location. Information related to location as well as time is stored as attributes. In ES community, geo-phenomenon are stored as set of parameters that vary as continuous function in 3- dimensional space and time (Domenico et al., 2007). The datasets are stored as multidimensional array structure and time is stored as dimension of the structure.

Both communities provide data to the wide range of users using web services. Two communities have their own family of protocols. GISciecne community has OGC WCS for data access and Catalogue Service for Web (CSW) for searching metadata. Likewise ES community has developed set of protocols like OPeNDAP for data access and Thematic Realtime Environmental Distributed Data Services (THREDDS) catalogue for finding datasets. In this chapter these protocols will be discussed with aim to choose an appropriate client.

3.2 WEB SERVICES

The internet is no longer only a medium to view and download contents. Internet is now being used as computing platform and it is possible due to development in web services technology. According to Gottschalk et al. (2010) "It is an interface that describes a collection of operations that are network accessible through standardized XML messaging". Web services allow users to access the functionalities via web using set of open standards. Due to these open standards system is neutral to operating system, programming languages and hardware too. The web service can be described in the form of Service Oriented Architecture (SOA). This architecture defines three roles and three operations. Three roles are service requester, service provider and service registry and three operations are: publish, find and bind.

- Service provider publishes or registers its service in some service registry based on the standard called Universal Description, Discovery, and Integration (UDDI) specification.
- Service requester makes queries to obtain a reference to required service using UDDI interface.
- Service requester gets information about accessing the service through Web Service Description Language (WSDL) and invokes the service and retrieve required functions

Web services that allow implementations to be updated and renewed without affecting clients are commonly known as "loosely coupled". In tightly coupled systems, if one of these interface changes, others communicating with it must be updated in order to accommodate the changes.

3.3 OGC WEB SERVICE

The Open Geospatial Consortium, Inc. (OGC) is leading organization that is responsible for the development of standards for geospatial and location based services. The geo service standards developed by OGC can be accessed without any cost. The standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications (OGC, 2010). The standards specified by OGC can be grouped as:

- portrayal services: that support visualization of geospatial information (for example : Web Map Service)

- data services: that allow access to collection of contents in databases as well as remote repositories through standard interface (for example Web Coverage Service, Web Feature Service)

- processing services: that are for web based processing (for example Web Processing Service)

On the basis of these standards SOA can be implemented in geospatial applications.

3.3.1 Web Coverage Service (WCS)

WCS supports networked interchange of geospatial data as "coverage" based on subsetting, scaling and reprojection. The term coverage is defined by ISO and OGC as "space varying phenomenon" i.e. geoographic object with some extent whose values depend on location and time. Clients can choose portions of information available on server based on spatial and other constraints similar to Web Map Service (WMS) and Web Feature Service (WFS). However there are some characteristics of WCS that make it different from WMS and WFS. "Unlike the WMS, which portrays spatial data to return static maps (rendered as pictures by the server), the WCS provides available data together with their detailed descriptions; defines a rich syntax for requests against these data; and returns data with its original semantics (instead of pictures) which may be interpreted, extrapolated, etc.-and not just portrayed". "Unlike WFS which returns discrete geospatial features, the WCS returns coverage representing space-varying phenomena that relate a spatio-temporal domain to a (possibly multidimensional) range of properties"(OGC, 2008).

There are three basic operations of WCS that are: GetCapabilities, DescribeCoverage and GetCoverage.

GetCapabilities returns XML document that describes services provided by WCS server and brief description of coverage. DescribeCoverage allows clients to request detail information of about available coverages. Server responds with XML document that describes services available for selected coverages. GetCoverage operation is performed after GetCapabilties and DescribeCoverage operations. This operation allow spatial, temporal and band subsetting, scaling, reprojection, and final result packaging, including data format encoding. One GetCoverage operation returns single coverage at a time that is encoded in a well-known coverage format like HDF-EOS, NITF, and GeoTIFF.

Detailed Functional Description:

The mandatory and optional parameters for each operation are discussed below.

GetCapabilities: GetCapabilities returns an XML document that describes service and data collections from which clients may request coverages.

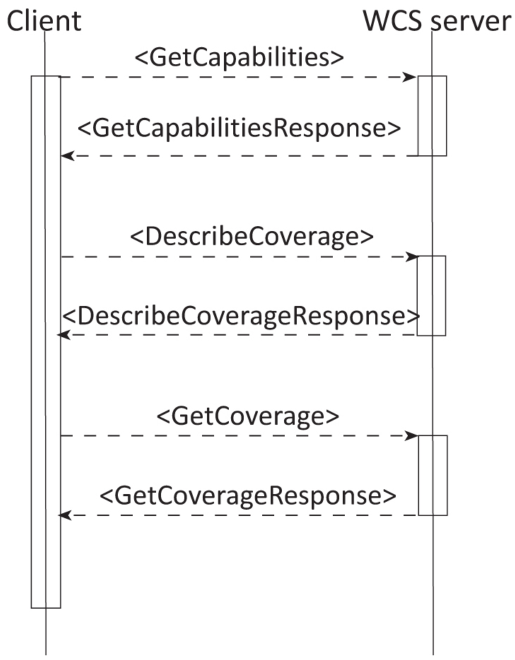


Figure 3.1: Basic operations of WCS

DescribeCoverage: DescribeCoverage operation returns an XML document that describes one or more coverages served by WCS. It provides the information for a client to assess the usability of data and to formulate GetCoverage requests.
Mandatory:

- SERVICE: "WCS".
- REQUEST: "DescribeCoverage".
- VERSION: the requested version.

Optional:

- COVERAGE: a comma-separated list of names of coverages to describe. If this is not mentioned , all coverages supported by server is returned.

DescribeCoverage Response: In response to a DescribeCoverage request, WCS returns XML document whose top-level element is a CoverageDescription containing CoverageOffering elements for all covereges requested. CoverageOffering extends CoverageOfferingBrief, to provide additional details on the domain and range of a coverage offering. The following are mandatory and optional CoverageOffering elements:
Mandatory:

- name (CoverageOfferingBrief): name of the coverage.
- label (CoverageOfferingBrief): a descriptive label of the coverage that can be used by the client for presentation in client forms or menus.

- lonLatEnvelop (CoverageOfferingBrief): a pair of GML position elements, in the WGS 84 geographic CRS with longitude preceding latitude and both using decimal degrees only.
- domainSet (addition to CoverageOfferingBrief) : domainSet must include a spatialDomain, or a temporalDomain, or both.
- rangeSet (addition to CoverageOfferingBrief): is a description of coverage values available from a coverage offering. It has the following elements and attributes:
  - name (mandatory)
  - label (mandatory)
  - metadataLink (optional)
  - descrition (optional)
  - axisDescription (optional)
  - nullValues (optional)
- supportedCRSs : a list of the CRSs in which the server can respond to GetCoverage requests.
- supportedFormats: a list of formats that GetCoverage supports. The service must support at least one of the following formats, to be compliant:
  - GeoTIFF
  - HDF-EOS
  - DTED
  - NITF
  - GML

Service providers can choose to support additional formats too.
Optional:

- metadatalink (CoverageOfferingBrief): a series of Xlink attributes to point to external sources of metadata. Metadata can be any of the following types: TC211, FGDC, other.
- description (CoverageOfferingBrief): description of the coverage
- keywords (CoverageOfferingBrief): keywords that describe the coverage
- SupportedInterpolations (addition to CoverageOfferingBrief): a coverage offering may support any of the six spatial interpolation methods:
  - nearest neighbor (default)
  - bilinear
  - bicubic
  - lost area
  - barycentric
  - none

GetCoverage: GetCoverage returns values or properties of regularly spaced locations, bundled in a requested format.

GetCoverage Request Mandatory:

- SERVICE: "WCS".
- REQUEST: "GetCoverage".
- VERSION: the requested version.
- COVERAGE: the name of the coverage requested. The current version supports a single coverage request only.
- CRS: name of the coordinate reference system in which requested domain constraints are expressed (BBOX).
- BBOX: minx, miny, maxx, maxy, minz, maxz (minz and maxz are optional).
- TIME: time1, time2, or min/max/res. This is used to request a subset corresponding to the specified time instants or intervals. The parameter "res" is optional.
- WIDTH, HEIGHT, DEPTH: all integers are used for requesting a coverage to be returned with a specific grid size, i.e. number of grid points or cells.
- RESX, RESY, RESZ : the parameters RESX and RESY define the grid-cell size along the first and second axes of the coordinate reference system given in RESPONSE_CRS. RESZ is optional for 3D grids. Either these or WIDTH, HEIGHT, and DEPTH are required, but if interpolation is not supported for the requested coverage, both sets of parameters will be ignored. In that case, BBOX alone will be used for sub-setting.
- FORMAT: the format to be used for returning the coverage.

Optional:

- RESPONSE_CRS: the name of the coordinate reference system in which the coverage response should be referenced. If omitted, the coverage will be returned in the same coordinate reference system as the request CRS (defined by CRS parameter).
- Parameter: val1, val2, or min/max/res. This is only used for range sets with compound values. The Parameter key is a variable string. It must match the name of a parameter listed in the range set description of the selected coverage. The parameter "res" is optional. The values are optional if the chosen range component has default values for the parameter.
- EXCEPTIONS: The format in which exceptions are to be reported by the Server. The default is "application /vnd.ogc.se_xml". Every WCS must support the default format.
- Vendor-specific parameters: experimental parameters.

GetCoverage Response: The response to a valid GetCoverage request is a coverage extracted from the coverage requested, with the specified spatial reference system, bounding box, size, value sets, and format.

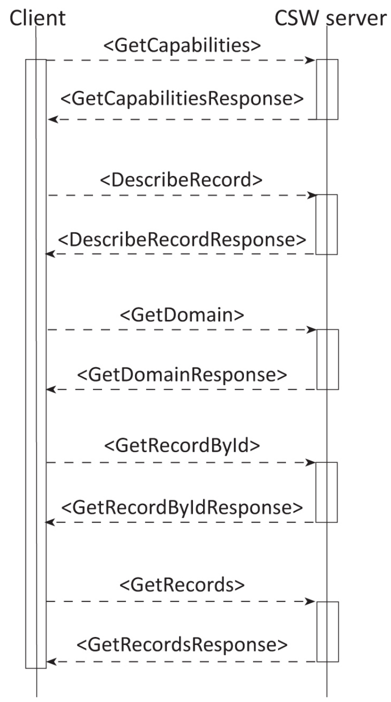


Figure 3.2: Basic operations of CSW

WCS versions

The recent version of WCS is 2.0 announced in November 2010. WCS version 2.0 is modified to support GML coverage model. WCS 2.0 supports all GML and ISO coverage types. The earlier versions 1.1.0 had functional changes like use of GridCRS in coverage description and requests, hierarchical coverage description. The WCS versions are modified during addition of extensions like processing extensions, transaction operation extension. The early versions of WCS were 0.5 and 0.7 that specifies operations of WCS.

3.3.2 Catalogue Service for Web

CSW is another OGC protocol that is used to find data and related services. It provides publication and search functionality and hence enabling a requester to discover and communicate with resource provider. In this process requester need not to have prior knowledge about the provider. CSW supports GetCapabilities, GetRecords, DescribeRecord and GetRecordById. The process is started when client sends GetCapabilities request to CSW server. Then server sends response in XML document that contains service metadata about the server. DescribeRecord operation is used to discover elements of information model supported by CSW server. GetDomain operation is optional. It is used to get runtime information about the range of values of a metadata record element. GetRecord operation performs two tasks that are search and present. The search portion is encoded using Query element. Similarly present portion of GetRecords operation is encoded using the outputSchema parameter and ElementName/ElementSetName parameters. The GetRecordByID request retrieves the default representation of catalogue records by using their identifiers. The basic operations of CSW is shown also in figure 3.2.

3.4 OPENDAP

OPeNDAP is a data access protocol that is mainly used in Earth Science domain. Its architecture is based on client server model. OPeNDAP provides client software that send request for data to server using the internet. The server responds to requests for dataset structure, data attributes and data itself (OPeNDAP, 2004).

OPeNDAP data access protocol (DAP) defines the communication method between client and server. DAP has four components they are:

- An intermediate data representation for data sets: that is used to transport data from the remote source to the client.
- A format for the ancillary data: is required to translate data set into intermediate representation and again translate intermediate representation into target data model. The ancillary data contains two parts that are Data Description Structure (DDS) and Data Attribute Structure (DAS).
- A procedure: that retrieves data and ancillary data from remote platform.
- An API: that consists OPeNDAP classes and data access calls which are designed for implementation of protocol.

The services requested from an OPeNDAP server are specified in a suffix appended to the URL. Depending on the suffix supplied, the server will provide one of following services:

- Data Attribute: This service is initiated when the server receives a URL with .das as suffix. It returns entire data attribute structure for given dataset in the form of text file.
- Data Descriptor: This service is initiated when the server receives a URL with .dds in its suffix. It returns a text file that describes the structure of the variables in the dataset
- OPeNDAP Data: This service is initiated when the server receives a URL with .dods in its suffix. It returns the actual data encoded as a Multipurpose Internet Mail Extensions (MIME) document.
- ASCII Data: This service is initiated when the server receives a URL ending with .asc or .ascii in its suffix. It returns requested data in ASCII format.
- WWW Interface: When the server receives a URL ending in .html, it produces an HTML form containing information from the dataset that can be used to create URL with which to request OPeNDAP data.
- Information: This service is initiated when the server receives a URL ending with .info in its suffix. It returns information of server and dataset, in human-readable HTML form. The returned document may include information about data server as well as dataset referenced in the URL.
- Version: This service is activates when server receives a URL ending with .ver. It returns the version information about OPeNDAP server software on the server.
- Help: This service is initiates when server receives URL ending in any suffix that is not recognized. It returns some help text in response.

3.5 THREDDS CATALOGUE

Thematic Realtime Environmental Distributed Data Services (THREDDS) is a project under Unidata. The users can use this middleware to publish, find and interact with data in convenient and efficient manner. THREDDS catalogue is one of the important components of TDS architecture. THREDDS catalogue can be defined as a way to describe an inventory of datasets. The available datasets are presented in simple hierarchical structure. These catalogues also provide methods to access each dataset, a human readable name for each dataset, and a structure on which further descriptive information can be placed. THREDDS catalogue serves local datasets as well as datasets on other servers like THREDDS or OPeNDAP servers. The OPeNDAP servers supports THREDDS catalogue interface.

Unidata has implemented WCS gateway. This gateway mediates between the THREDDS model and the OGC-WCS model. It supports three basic WCS operations that were discussed in section 3.3.1. The THREDDS WCS gateway can be used for spatial dimension query (using BBOX constraint), temporal dimension query (using TIME constraint) and multiple output formats (using FORMAT constraint). However some other features, subsetting and resampling using WIDTH/HEIGHT/DEPTH or RESX/RESY/RESZ, InterpolationMethod are not supported.

3.6 STATE OF THE ART: WCS AND OPENDAP SERVERS

The governmental organizations and universities archive observation and model data in the web servers using different standards as discussed in sections 3.3 and 3.4. The environmental datasets that are required for the case study mentioned in chapter 2 were explored in some of the servers. The servers following different standards were chosen for the study they are mentioned below.

- The Oak Ridge National Laboratory Distributed Active Archive Center (ORNL DAAC) maintains THREDDS data server. The server is located at http://daac.ornl.gov/thredds.shtml. It serves data through OPeNDAP, WCS version 1.0.0, NetCDF subset service and HTTP server. It provides regional and global data including vegetation data.

- TPAC has installed a THREDDS data server to serve public datasets. It uses OPeNDAP, Web Coverage Service (WCS) version 1.0.0, HTTP as well as publishing GridFTP URLs. The server is located at http://ngportal.sf.utas.edu.au/thredds/. It serves oceanographic and climate datasets.

- Jet Propulsion Laboratory (JPL) maintains OPeNDAP server. It provides AVHRR Sea Surface Temperature data , NSCAT Ocean Wind Vector, Wind Speed, and other oceanographic data. The server is located at http://dods.jpl.nasa.gov/opendap/.

- GMU maintains server that provides data using WCS version 1.0.0 and 1.1.0. The server is located at http://geobrain.laits.gmu.edu/cgi-bin/wcs110. It serves Landsat, MODIS, ASTER, GOES, Windsat data sets.

The table 3.1 summarizes some of the datasets.

Table 3.1 List of data providers

S.no	Source	URL	Variable	Coverage
1	JPL	`http://dods.jpl.nasa.gov/opendap/sea\_surface\_temperature/modis/data/aqua/L3\_mapped/sst/contents.html`	SST	Daily, Glboal
2	JPL	`http://dods.jpl.nasa.gov/opendap/ocean\_wind/quikscat/L2B/data/`	Ocean wind	Daily, Global
3	TPAC	`http://ngportal.sf.utas.edu.au/thredds/wcs/library/ncep2/pressure/uwnd/uwnd.2010.nc`	U wind	6 hourly, Global
4	TPAC	`http://ngportal.sf.utas.edu.au/thredds/wcs/library/ncep2/pressure/vwnd/vwnd.2010.nc`	V wind	6 hourly, Global
5	ORNL	`http://thredds.daac.ornl.gov/thredds/wcs/973/gimms_ndvi_qdeg_1981-2002/gimms_ndvi_qd_20021000.nc`	NDVI	Monthly, Global

3.7 WCS CLIENTS

WCS Clients are software packages or modules that are capable of performing basic WCS operations. An ideal WCS client is capable of communicating with OGC-compliant coverage servers for accessing multidimensional geospatial data and handling different coverage-encoding formats. Besides performing basic client-server communication, WCS clients are capable of accessing, visualizing coverage, and interacting with the user. The client also provides georectification, reprojection, and reformatting functions. The execution of those functions is performed according to the requirements of users and capabilities of the servers. "The interaction between WCS client and OGC compliant web coverage servers provides interoperable, personalized, on-demand data access and services of remote sensing data"(Di et al., 2002).

Some of the available WCS clients are explained below.

3.7.1 gvSIG

gvSIG is desktop GIS application that can handle most of the raster and vector data formats. Moreover it is capable of capturing, storing, handling, analyzing and deploying any kind of referenced geographic information. It can access to remote servers using OGC complaints specifications WMS, WFS and WCS (gvSIG, 2010).

gvSIG supports WCS version 1.0.0 and contains WCS client that allows user to access raster data and add to geographical view that can be overlapped with other information from local or remote data servers. The data can be discovered using embedded catalogue service. So user can select data services from the catalogue or can provide the URL of the service provider. After connecting to the server gvSIG receives metadata information. The detail information about coverage is given in the next stage. In that stage it provides the list of available coverage data, file format, CRS and interpolation methods. In addition, temporal subsetting and band selection is also possible. Users have to select appropriate choices from the dialogue boxes before gvSIG executes "GetCoverage" in order to display coverage data. It is developed in platform independent environment using Java and designed to be easily extendable. There is provision of advanced functionalities like scripting support. This allows operations to be performed using external scripts.

3.7.2 OWSLib

OWSLib is open source python OGC library that can access remote data sources using WMS, WFS and WCS specifications. It offers common API for accessing service metadata and wrappers for basic WCS operations: GetCapabilities, DescribeCoverage and GetCoverage (Domenico & Lowe, 2009). It supports 1.0.0 and 1.1.0 versions of WCS.

The first step is to instantiate a WebCoverageService object for a particular WCS service. This will call the "GetCapabilities" method of the server and populate appropriate python metadata attributes. The several available coverages are provided. Users can explore more information about a particular coverage (for example spatio-temporal extent, available output formats).

```
>>> airtemp.timelimits #get the temporal extents
      ['2024-01-15T00:00:00.0', '2054-12-15T00:00:00.0']
#find out which output formats are supported
>>> airtemp.supportedFormats
     ['cf-netcdf', 'GeoTiff']
```

This calls the DescribeCoverage method on the server to retrieve coverage specific metadata. DescribeCoverage requests can be expensive i.e., it can retrieve long list of coverage of different time. In order to maintain performance the detailed metadata is only retrieved from the server if it is specifically requested by the client. By using the information gained during "DescribeCoverage", a GetCoverage request is formulated and sent to the server.

```
output=wcs.getCoverage(identifier='AirTemperature',
time=['2024-01-15T00:00:00.0'],
bbox=(-80,30,50,60), format='cf-netcdf')
>>> f=open('test.nc', 'wb')
>>> f.write(output.read())
>>> f.close()
```

The output coverage file is then written to disk and viewed using suitable software. OWSLib can also supports Catalogue Service for Web (CSW) and can be used to discover metadata infromation of data providers.

Python has libraries to read scientific data formats, such as HDF (pyhdf) and netCDF (pynetcdf) and it can be combined with OWSLib. So OWSLib module can be incorporated easily into standard-alone desktop or web-based client as middleware between software components.

3.7.3 GDAL

GDAL is a translator library for raster geospatial data. It presents a single abstract data model to the calling application for all supported formats. It also comes with a variety of useful command line utilities for data translation and processing. It supports over 50 raster formats range of raster data formats including HDF4, NetCDF and also can access WMS and WCS servers using GDAL WCS driver. The current driver supports WCS 1.0.0 and WCS 1.1.0 servers. In addition it provides utilities for data translation, image warping, subsetting, and various other common tasks (GDAL, 2010). WCS server is accessed by creating a local service description XML file. The file contains the coverage server URL, and the name of coverage. There should be be no spaces or other content before the<WCS_GDAL>element. The example XML file is shown below.

```
<WCS_GDAL>
<ServiceURL>http://laits.gmu.edu/cgi-bin/NWGISS/NWGISS?
\</ServiceURL>  <CoverageName>AUTUMN.hdf</CoverageName>
</WCS_GDAL>
```

GDAL can be accessed from various programming platforms like python, c++, perl. It is used by different open source GIS applications like GRASS, MapServer, QGIS, and OpenEV as primary data access engine.

3.7.4 Multi-Protocol Geoinformation Client (MPGC)

Multi-Protocol Geoinformation Client (MPGC) is the OGC compliant multi-purpose client that can access geospatial data using WCS, WFS, WMS and Web Registry specifications. MPGC supports WCS 0.5 and 0.7 versions. MPGC can handle range of datasets like HDF, GeoTiff, GML, JPG, PNG, and GIF. Besides accessing subset of multi-dimensional data in different formats, it has functionalities like reprojection, resampling, reformatting, subsetting and visualization as well as analysis of multi-dimensional data (MPGC, 2005). MPGC is equipped with Catalog Service for Web (CSW) specification so that services can be discovered and registered. The server URL can also be taken as input from users. The list of coverage data is derived from "GetCapabilities" and other attributes for each data like bbox, range set, resolution, and spatial reference system, are derived from the "DescribeCoverage" response. On the basis of these information user can select particular coverage data. After choosing data, users can subset coverage data in spatial, temporal, resolution/size and range dimensions. Then "GetCoverage" request is sent on according to setting made by user and coverage is retrieved. The MPGC is a standard alone thick client based on Java platform. It uses the HDF Java Native Interfaces, which calls the HDF library.

3.7.5 Gaia

Gaia is an open-geospatial viewer that can access multiple geospatial sources such as OGC WMS, WCS and WFS, commercial services such as Microsoft Bing Maps, OpenStreetMap and Yahoo! Maps. This client supports various file formats including ESRI Shapefiles, Google Earth KML/KMZ, DXF, MIF, Geography Markup Language (GML) and GML for Simple Features (GMLsf)(Gaia, 2010).

Gaia supports accessing WCS server versions 1.0.0, 1.1.0 and 1.1.1. The list of WCS servers is provided that can be added and updated. Once a server is chosen, different available coverage are extracted. The required coverage can be selected and added as layer after inspecting provided preview and short information. There is provision of spatial subsetting using parameters but time subsetting is not possible in the current version. It supports GeoTIFF and NetCDF file formats.

Table 3.2 Comparison between different WCS client applications.

	OWSLib	gvSIG	GDAL	Gaia	MPGC
Interpolation	Yes	Yes	Yes	No	Yes
Spatial subset	Yes	Yes	Yes	Yes	Yes
Temporal subset	Yes	Yes	Yes	No	Yes
WCS version support	1.0.0, 1.1.0	1.0.0	1.0.0, 1.1.0	1.0.0, 1.1.0, 1.1.1	0.5, 0.7
Format of GetCoverages	GeoTIFF, HDF, NetCDF	GeoTIFF	Most of the raster formats	GeoTIFF, NetCDF	GeoTIFF, HDF, NetCDF
Extensibility	Yes	Yes	Yes	Yes	Yes
Integration into client server architecture	Yes	Yes	Yes	No	-
Metadata /Catalogue support	Yes	Yes	No	No	Yes
OPeNDAP/THREDDS	No	No	No	No	No

An open Extenders API was introduced in version 3.2. Users can extend and expand Gaia with new functionalities and capabilities.

The comparison between different WCS clients is shown in following table. The criteria for choosing best WCS clients are:

- Interpolation
- Spatial subset
- Temporal subset
- WCS version support
- Format of GetCoverages
- Extensibility
- Integration into client server architecture.
- Metadata/ Catalogue support
- OPENDAP/THREDDS support

Most of the WCS clients support the mandatory functions of the WCS and also provides support for most of the raster formats like NetCDF and GeoTIFF. The possibility of integration into system as middleware to access WCS server was also examined. The advanced functionality of using scripting language can enable the client program to act as middleware. gvSIG has scripting functionality with jython. Similarly OWSLib and GDAL can be used with programming languages.

Large volume of earth observation data are provided in the internet using OPeNDAP standard but only few data servers have adopted WCS standards. So it would be better if WCS client can also access data servers implemented using OPeNDAP standards. The WCS clients are unable to connect OPeNDAP servers. Nevertheless python module named Pydap can be used to access

OPeNDAP. OWSLib module supports the latest version of WCS,can connect with CSW servers, and has possibility to use as middleware. It can be used along with other python modules that can connect with OPeNDAP server. So OWSLib is appropriate WCS client. OPeNDAP protocol and Pydap library will be discussed in next section.

3.8 OPENDAP CLIENTS

3.8.1 IDV

The Integrated Data Viewer is an open source software developed by Unidata (IDV, 2010). It provides features for analyzing and visualizing wide range of data from satellite, radar to shapefiles and GeoTIFF. The metadata catalogue of THREDDS can be accessed to check available datasets in remote servers. It can be used to access multiple servers using different protocols like OPeNDAP, ADDE, HTTP. Users can select entire data or subset of data. IDV also allows to execute programs using jython scripts.

3.8.2 Pydap

Pydap is a Python library that implements OPeNDAP. It can be used as a client to access scientific datasets efficiently; or as a server to distribute data in varieties of formats (Pydap, 2010). It is possible to introspect and manipulate a dataset as if it were stored locally, with data being downloaded on-the-fly as necessary. Pydap client uses pydap.client.open_url function to connect with specified server. Once the connection is made, it can be interrogated using other functions. The names of variables can be checked using simple code.

```
>>print dataset.keys()
```

The available variables are displayed in the form of the dictionary. Further information about variables like dimensions, shape and attributes can also be checked. Users can subset the data using indexes. For example:
Using indexes

```
>>> print  sst[0,10:14,10:14]
```

Using spatial extent

```
>>>print sst[ 0 , (-10 < sst.COADSY) &
(sst.COADSY < 10) , (sst.COADSX > 320)
& (sst.COADSX <328)]
```

3.9 CONCLUSION

In this chapter different standards from ES and GIScience community were reviewed and available WCS clients were studied. On the basis of pre-defined criteria OWSLib module of python was chosen. The application can be built on python using modules OWSLib and Pydap. This application will be capable to access both OPeNDAP as well as WCS servers.

Chapter 4

Matching data: Design and implementation of the prototype

In chapter 2, two case studies were done to identify influencing external factors that affect movement of iceberg and animals. In chapter 3 standards and clients used to access environmental data were discussed. This chapter will describe processes that are required to implement the matching of trajectory data and environment data. In later sections three tier architecture for web based prototype will be elaborated in detail. In addition major issues of matching i.e metadata, fetching environmental data, matching data sets, interpolation methods will be addressed.

4.1 CONCEPTUAL FRAMEWORK FOR WEB BASED DATA MATCHING

Trajectory data are needed to be matched with external environment data to find relationship among them. The users, i.e. scientists and researchers need matching system that is capable of following functionalities :

- allows user to search for available data providers
- takes input from user for selecting the best data source, interpolation methods
- utilizes WCS and OPeNDAP for accessing grid data from the selected remote servers
- performs matching and if necessary interpolation for missing grid values

A conceptual architecture framework was chosen to address users' requirement. This architecture consists three basic components, i.e. client, middleware and server. The conceptual diagram of system is shown in figure 4.1

In client based interface user interacts with system by uploading trajectory data, selecting servers, choosing appropriate coverage data and suitable interpolation method for finding missing values and finally getting matched datasets. The role of middleware is to connect with client and send requests to server, to perform data matching, interpolation. The third tier is servers that are located elsewhere run by different organizations. Servers response with metadata and coverage after receiving requests from client. The work-flow of the system is shown in the sequence diagram in figure 4.2

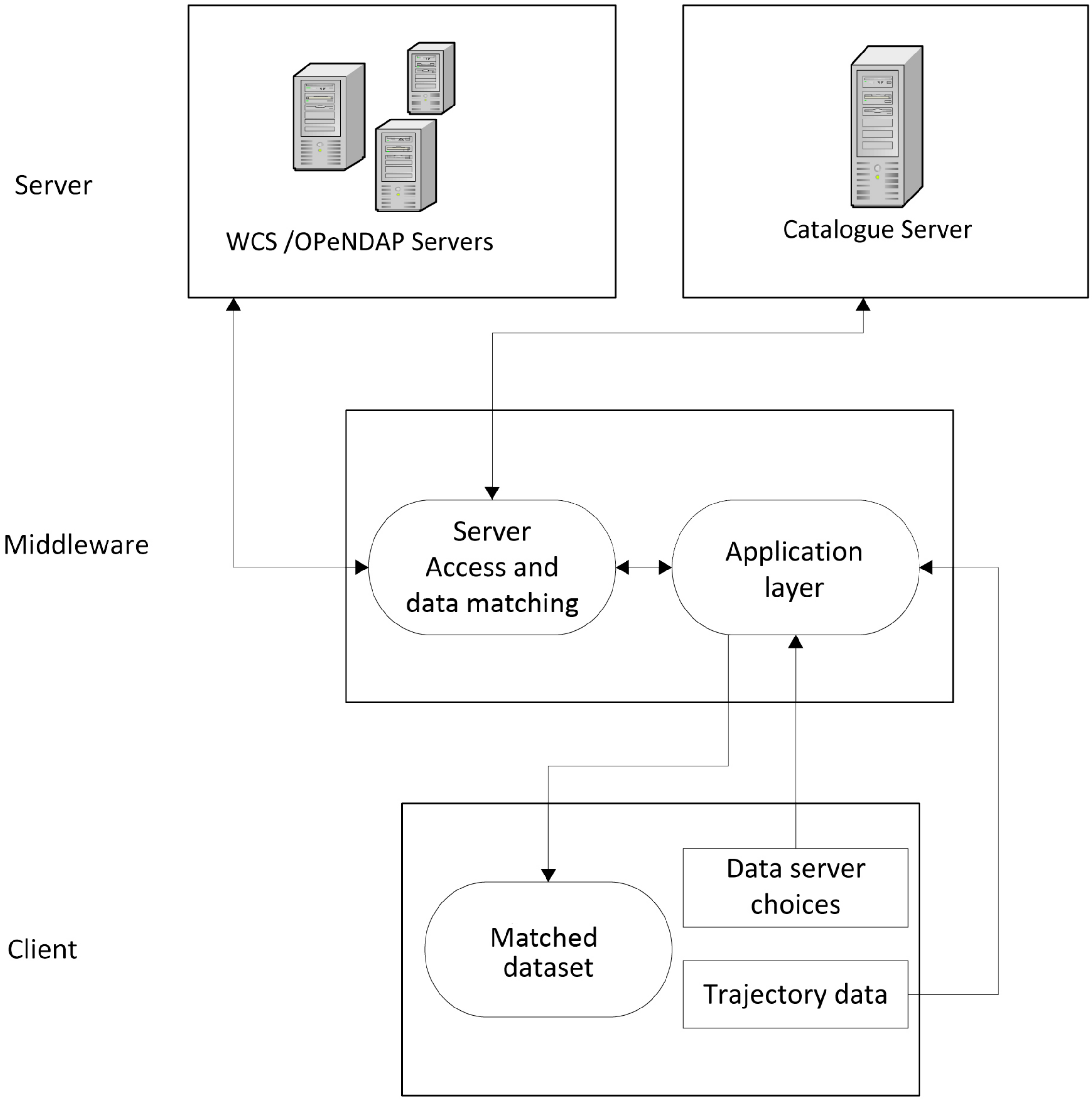


Figure 4.1: Conceptual architecture diagram

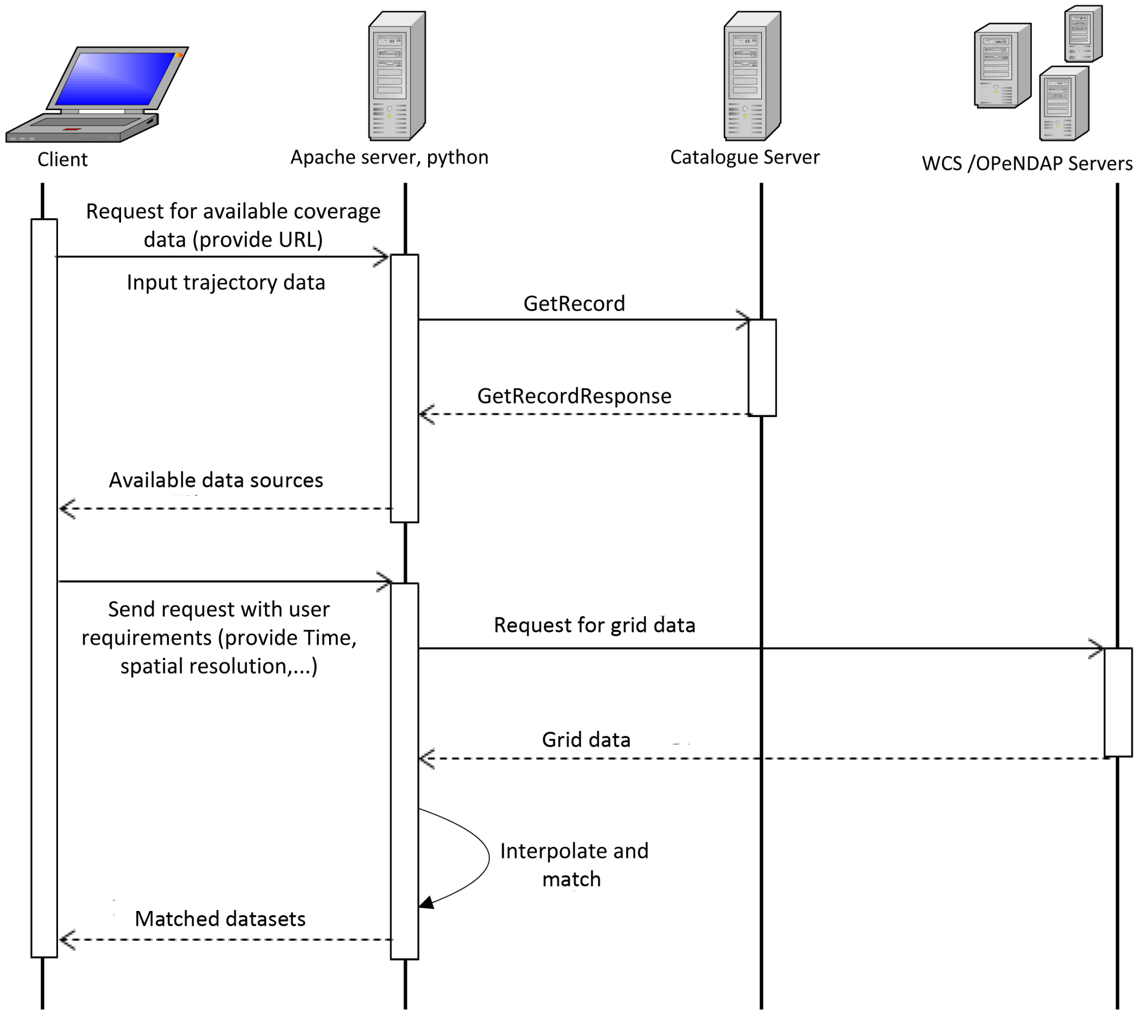


Figure 4.2: Sequence diagram for accessing coverage and matching with trajectory data

4.1.1 Client tier: user interface

User can interact with the system using client side interface that was built using HTML and javascript. User uploads trajectory data, chooses appropriate data sources, selects interpolation method and receive matched data set using this interface. The activities performed by client tier are discussed in following sections:

Uploading Trajectory data

At first user uploads trajectory data from local disk in CSV format using the interface. This data contains minimum information of moving objects like serial no, time, longitude, latitude etc. The iceberg trajectory data used during the test is shown in figure 4.4. The uploaded CSV file is stored in web server at middleware for further processing. The interface used to upload data is shown in figure 4.3.

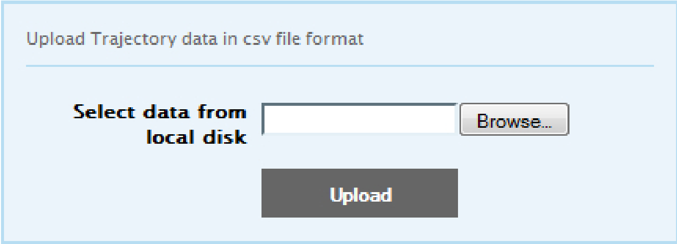


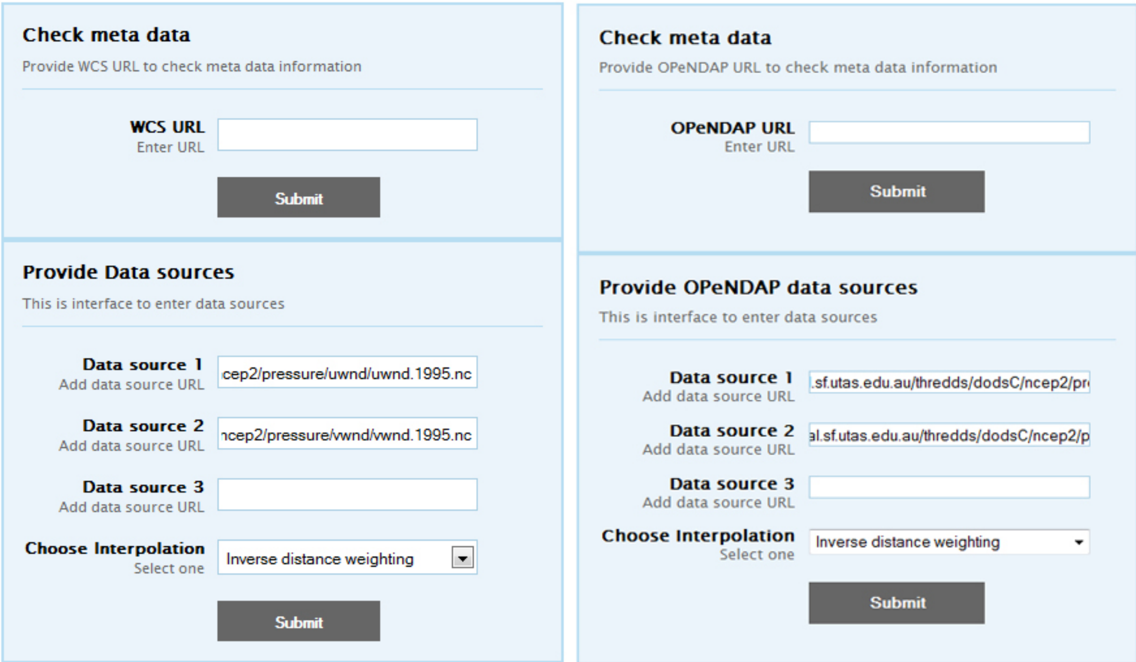
Figure 4.3: User interface to upload trajectory data

```
SrNo,iceberg,Gregorian,latitude,longitude,size,area_KM2
428,A22A,11.01.1995,-76.5,-43.9,36x27,3333.866688
429,A22A,17.01.1995,-76.5,-43.6,36x27,3333.866688
430,A22A,25.01.1995,-76.5,-44.2,36x27,3333.866688
431,A22A,02.02.1995,-76.5,-43.9,36x27,3333.866688
432,A22A,08.02.1995,-76.5,-43.9,36x27,3333.866688
433,A22A,14.02.1995,-76.5,-43.9,36x27,3333.866688
434,A22A,22.02.1995,-76.5,-43.9,36x27,3333.866688
435,A22A,01.03.1995,-76.5,-43.9,36x27,3333.866688
436,A22A,10.03.1995,-76.5,-43.9,36x27,3333.866688
437,A22A,04.04.1995,-76.5,-43.9,36x27,3333.866688
438,A22A,10.04.1995,-76.5,-43.9,36x27,3333.866688
439,A22A,17.04.1995,-76.5,-43.9,36x27,3333.866688
440,A22A,07.05.1995,-76.5,-43.7,36x27,3333.866688
441,A22A,30.05.1995,-76.5,-43.7,36x27,3333.866688
442,A22A,06.06.1995,-76.5,-43.7,36x27,3333.866688
443,A22A,13.06.1995,-76.5,-43.7,36x27,3333.866688
444,A22A,23.06.1995,-76.7,-43.8,36x27,3333.866688
445,A22A,28.06.1995,-76.7,-43.8,36x27,3333.866688
446,A22A,11.07.1995,-76.7,-43.8,36x27,3333.866688
447,A22A,18.07.1995,-76.5,-44,36x27,3333.866688
448,A22A,24.07.1995,-76.4,-44,36x27,3333.866688
449,A22A,01.08.1995,-76.5,-43.9,36x27,3333.866688
450,A22A,05.08.1995,-76.4,-43.9,36x27,3333.866688
451,A22A,15.08.1995,-76.4,-43.9,36x27,3333.866688
452,A22A,29.08.1995,-76.5,-44.3,36x27,3333.866688
453,A22A,12.09.1995,-76.3,-44.1,36x27,3333.866688
454,A22A,19.09.1995,-76.5,-44.4,36x27,3333.866688
455,A22A,30.10.1995,-76.4,-44.7,36x27,3333.866688
456,A22A,21.11.1995,-76.4,-44.7,36x27,3333.866688
```

Figure 4.4: Iceberg trajectory data in CSV format

Metadata for choosing an appropriate data source

After uploading data, the available data providers are searched using CSW client. Users have to provide URL to location of the CSW server. The bounding box of the area is chosen from trajectory data itself. CSW client sends request to the server to get information about the data providers for the particular area. The list of available services is displayed as a result of the query. Once user gets information about data providers and link to their servers, metadata can be checked. Spatio-temporal resolution of the data, coverage information can be obtained from metadata. On the basis of available information in metadata, appropriate data server is chosen. One of the server used during testing of prototype is THREDDS Data Server (TDS) installed by Tasmanian Partnership for Advanced Computing (TPAC). This server provides data through



(a) WCS (b) OPeNDAP

Figure 4.5: User interface to access data server

OPeNDAP, WCS, HTTP as well as publishing GridFTP URLs. The server is located at `http://portal.sf.utas.edu.au/thredds/`. Available dataset can be overviewed using WCS getcapabilties requests and OPeNDAP DDS and DAS requests. The request for metadata was sent for data source available in `http://portal.sf.utas.edu.au/thredds/catalog/ncep2/pressure/uwnd/catalog.html?dataset=ncep2/pressure/uwnd/uwnd.1995.nc` using interface shown in figure 4.5. The getcapabilities response is shown below:

OPeNDAP has a bit different metadata conventions. The response of metadata request is shown below:

```
uwnd {
        String long_name "6-hourly U-wind on Pressure Levels";
        Int16 valid_range -32765, -1266;
        Float32 unpacked_valid_range -140.0, 175.0;
        Float32 actual_range -100.5, 135.6;
        String units "m/s";
        Float32 add_offset 187.65;
        Float32 scale_factor 0.01;
        Int16 missing_value 32766;
        Int16 _FillValue -32767;
        Int16 precision 2;
        Int16 least_significant_digit 1;
        Int16 GRIB_id 33;
        String GRIB_name "UGRD";
        String var_desc "u-wind";
        String dataset "NCEP/DOE AMIP-II Reanalysis (Reanalysis-2)";
```

Data Source : http://portal.sf.utas.edu.au/thredds/wcs/ncep2/pressure/uwnd/uwnd.1995.nc

Meta data information

Title	None
Abstract	None
Keywords	[]
Access	NONE
Fees	NONE
Available Coverages	['uwnd']

Information about available coverages

Title	6-hourly U-wind on Pressure Levels
Keywords	[]
Time Limits	['1995-01-01T00:00:00Z', '1995-12-31T18:00:00Z']
SupportedCRS	['OGC:CRS84']
Formats	['GeoTIFF', 'GeoTIFF_Float', 'NetCDF3']
Boundingbox	(0.0, -90.0, 357.5, 90.0)

Figure 4.6: Output of getcapabilities request

```
        String level_desc "Pressure Levels";
        String statistic "Individual Obs";
        String parent_stat "Other";
        String standard_name "eastward_wind";
    }
```

Choosing appropriate data

User can provide different data sources for different environmental data like wind components, sea surface temperature, NDVI etc. While entering data source, users should use different interface for OPeNDAP and WCS as shown in figure 4.5.

During test of the prototype, external environmental data were chosen from two different WCS servers. The sea surface temperature data was derived from THREDDS server maintained by National Oceanographic Data Center (NODC). Similarly U and V components for wind direction were derived from server maintained by TPAC. The URL of data sources are:

- Sea surface temperature: `http://data.nodc.noaa.gov/thredds/wcs/ghrsst/cfg/aggregation/aggregate_GHRSST-V1_L4_GLOB_NCDC_AVHRR_OI_1995.ncml`
- U component of wind `http://portal.sf.utas.edu.au/thredds/wcs/ncep2/pressure/uwnd/uwnd.1995.nc`
- V component of wind `http://portal.sf.utas.edu.au/thredds/wcs/ncep2/pressure/uwnd/vwnd.1995.nc`

Users are provided with options for interpolation methods for finding missing grid values. In

Table 4.1 List of python modules used in middleware

S.no	Module name	Purpose
1	OWSLib	To connect and download data from WCS server.
2	Pydap	To connect and download data from OPeNDAP server..
3	Scientific.IO.NetCDF	To read and manipulate NetCDF files.
4	CSV	To read and save CSV files.

this prototype only inverse distance weighing was implemented. After submission of all information and uploading file, request is sent to middleware for further processing.

Finally user receives matched data sets in the form of CSV file format from middleware.

4.1.2 Middleware

Middleware runs on an apache web server and contains python codes that connect with user interface as well as data servers. It accepts requests from client side and process operations. The process starts by storing trajectory data uploaded by client. After user sends list of data servers, middleware accesses servers using different python modules. These separate python modules are used to send requests for metadata information and also coverage data. It sends metadata information received from data server to the client. Once client selects appropriate coverage name, it sends request for subset of coverage data to data server. After data gets downloaded in specified file format, it performs matching of trajectory data with grid data and extracts values from grid data. The extracted values are then merged as separate columns in trajectory data and saved as CSV file format. Afterwards matched datasets in the form of CSV file is sent to client. In later sections major tasks performed by middleware are described in detail.

Fetching environmental data

Environmental data such as sea surface temperature and ocean winds can be retrieved from the archives on the internet. Data are stored in the scientific data formats like NetCDF and HDF. These data can be accessed using WCS and OPeNDAP that were discussed in chapter 3.

The OWSLib python module was used to connect and fetch data from WCS server. Similarly Pydap python module was used to connect and fetch data from OPeNDAP server.

As available data would be large and difficult to manipulate, subset of data was downloaded using different techniques for different servers. This prior subsetting in time as well as in space ease manipulation of files during extraction phase.

- WCS subsetting

```
covID='uwnd'
bb=('311.3','-77.7','316.4','-75.3')
crstype='OGC:CRS84'
formatType='NetCDF3'
output=wcs.getCoverage(identifier=covID,bbox=bb,\
format=formatType,crs=crstype,resx='0.25',resy='0.25')
```

- OPeNDAP subsetting

```
sst = dataset['sst']
T=dataset['time']
X=dataset['lon']
Y=dataset['lat']
lon=X[(X > 313.5) & (X< 317.5)]
lat=Y[(Y > -78) & (Y < -74)]
print lat[:],lon[:]
time=T[(T>728303.0) & (T<728617.0)]
sub_sst=sst[(sst.time>728303.0)&(ss.time<728617.0),(sst.lat>-78)&
(sst.lat<-74),(sst.lon>313.5)&(sst.lon<317.5)]
return sub_sst,time,lat,lon
```

The subset of data is stored as NetCDF file format for later manipulation while using WCS. However while using OPeNDAP NetCDF file do not need to be stored for further processing.

Matching values for trajectory data

Once subset of data downloaded in NetCDF format it is read by using Scientific IO module of python. The following codes were used to read NetCDF file.

```
def readnetcdf():
    ncfilevwnd=Dataset('d:/tmp/vwnd1995new.nc')
    variableNames=ncfilevwnd.variables.keys()
    #print variableNames
    Y=ncfilevwnd.variables['lat']
    X=ncfilevwnd.variables['lon']
    vwnd=ncfilevwnd.variables['vwnd']
    T=ncfilevwnd.variables['time']
    return (vwnd,T,X,Y)
```

The following steps were performed in order to find matching data in stored NetCDF file. These steps were performed on the fly while using OPeNDAP.

1. Take trajectory data with (x,y,t) as input.
2. Find index of specific lat, lon and time.
3. Search for value in multidimensional array for same index.
4. Extract values for all selected points.

The time format of trajectory data was in ISO format while importing from local folder. Whereas time format in NetCDF file was stored in COARDS convention as mentioned in (Unidata, 2010). According to this convention time is stored as coordinate variable that is identifiable by its units. The specification of units are like : hours since 1995-01-01 00:00:00 or seconds since 1992-10-8 15:15:42.5 -6:00. So before matching data for specific time, time conversion was performed using following piece of codes.

```
for i in range(len(time)):
        T_array.append(datetimeToUdunits(time[i],\
        'hours since 1995-01-01 00:00:00'))
```

Wind direction is stored as two vector components in separate data sources. Two components were downloaded as separate files. These two vector components East-West component "U" and North-South component "V" were used to calculate wind speed and wind direction.

Interpolation and uncertainty

The matching grid cells can have missing values. In case of missing values interpolation method was applied as mentioned by user. In this prototype inverse distance weighing method was implemented. When missing values are detected function for interpolation is executed. The basic principle of inverse distance weighing was discussed in chapter 2. The function that was used to implement inverse distance weighing is given below:

```
def call_interpolate(lat,lon,time):
    latlist,lonlist,size,timestamp=readcsv()
    #print float(size[0])
    size_i=5*0.25


    if abs(search(lat,lon,time)-99)<10**-3 or search(lat,lon,time)==32767:
        a=(lat-size_i)/2
        b=(lon-size_i)/2
        value=0
        invdist=0
        earth_radius=7371.1*10**3
        count=0
        sumdistance=0
        for i in range(int(size_i/0.25)):
            for j in range(int(size_i/0.25)):
                data=search(a,b,time)
                if abs(search(lat,lon,time)-272.32)<10**-3 and data!=32767:
                    dLat=radians(a-lat)
                    dLon=radians(b-lon)
                    distance=2*earth_radius*arcsin(sqrt((sin((dLat)/2))**2
                    +cos(radians(a))*cos(radians(lat))*(sin((dLat)/2))**2))
                    count+=1
                    value+= data/distance**2
                    invdist+=1/distance**2
                    sumdistance+=distance

        value=value/invdist
        return value
        return count/sumdistance
```

Once data is interpolated the extracted values were joined with trajectory data and saved as CSV file format. The uncertainty quantification was not done. However users were notified about the interpolated values. The estimated values were indicated by "1" and other values were indicated by "0" in separate column of CSV file. The matched iceberg trajectory data with SST, wind direction in CSV format is shown figure 4.7. Similarly animal trajectory data matched with

NDVI data is shown in figure 4.8. The CSV file can be downloaded by user for further analysis or visualization.

SrNo	iceberg	Gregorian	latitude	longitude	size	area_KM2	SST	Interpolation	windspee	winddirectio	Interpolation
428	A22A	11.01.1995	-76.5	-43.9	36x27	3333.867	272.32	0	4.272	20.55606049	0
429	A22A	17.01.1995	-76.5	-43.6	36x27	3333.867	272.15	0	1.7	28.07251462	0
430	A22A	25.01.1995	-76.5	-44.2	36x27	3333.867	272.44	0	6.00333	29.98164812	0
431	A22A	02.02.1995	-76.5	-43.9	36x27	3333.867	272.28	0	1.50333	93.81399562	0
432	A22A	08.02.1995	-76.5	-43.9	36x27	3333.867	272.23	0	0.509904	11.31010086	0
433	A22A	14.02.1995	-76.5	-43.9	36x27	3333.867	272.18	0	4.11461	154.0576677	0
434	A22A	22.02.1995	-76.5	-43.9	36x27	3333.867	272.05	0	8.93588	174.8636081	0
435	A22A	01.03.1995	-76.5	-43.9	36x27	3333.867	271.92	0	5.25548	21.19406846	0
436	A22A	10.03.1995	-76.5	-43.9	36x27	3333.867	271.87	0	4.28018	232.5946491	0
437	A22A	04.04.1995	-76.5	-43.9	36x27	3333.867	271.51	0	1.3	337.380244	0
438	A22A	10.04.1995	-76.5	-43.9	36x27	3333.867	271.52	0	7.6922	195.8430729	0
439	A22A	17.04.1995	-76.5	-43.9	36x27	3333.867	271.54	0	3.20156	91.78987589	0
440	A22A	07.05.1995	-76.5	-43.7	36x27	3333.867	271.52	0	11.6499	214.5085199	0
441	A22A	30.05.1995	-76.5	-43.7	36x27	3333.867	271.44	0	1.13137	134.9998587	0
442	A22A	06.06.1995	-76.5	-43.7	36x27	3333.867	271.43	0	5.14782	330.9454271	0
443	A22A	13.06.1995	-76.5	-43.7	36x27	3333.867	271.5	0	12.088	141.0440775	0
444	A22A	23.06.1995	-76.7	-43.8	36x27	3333.867	271.52	0	8.70058	90.65853018	0
445	A22A	28.06.1995	-76.7	-43.8	36x27	3333.867	271.47	0	3.60694	43.87670151	0
446	A22A	11.07.1995	-76.7	-43.8	36x27	3333.867	271.44	0	2.14709	207.7585206	0
447	A22A	18.07.1995	-76.5	-44	36x27	3333.867	271.41	0	10.4019	88.89828616	0
448	A22A	24.07.1995	-76.4	-44	36x27	3333.867	271.4	0	5.44518	352.6139794	0
449	A22A	01.08.1995	-76.5	-43.9	36x27	3333.867	271.44	0	3.50571	86.72948253	0
450	A22A	05.08.1995	-76.4	-43.9	36x27	3333.867	271.43	0	3.60555	123.6900251	0
451	A22A	15.08.1995	-76.4	-43.9	36x27	3333.867	271.46	0	11.6619	210.9637542	0
452	A22A	29.08.1995	-76.5	-44.3	36x27	3333.867	271.43	0	9.47945	46.70980934	0
453	A22A	12.09.1995	-76.3	-44.1	36x27	3333.867	271.48	0	1.8868	327.9946966	0
454	A22A	19.09.1995	-76.5	-44.4	36x27	3333.867	271.43	0	6.96347	192.4395503	0
455	A22A	30.10.1995	-76.4	-44.7	36x27	3333.867	271.5	0	4.97393	59.82646705	0
456	A22A	21.11.1995	-76.4	-44.7	36x27	3333.867	271.61	0	3.20625	86.42363349	0

Figure 4.7: Matched iceberg data set in CSV format

SrNo	Animal	Gregorian	latitude	longitude	NDVI	Interpolation
428	Martin	03.10.2005	-3.86	34.33	0.211	0
429	Martin	04.10.2005	-3.31	34.51	0.255	0
430	Martin	05.10.2005	-2.57	34.41	0.323	0
431	Martin	06.10.2005	-2.65	34.36	0.229	0
432	Martin	07.10.2005	-1.42	34.49	0.459	0
433	Martin	08.10.2005	-1.23	34.72	0.505	0
434	Martin	09.10.2005	-0.73	35.04	0.55	0
435	Martin	10.10.2005	-0.31	35.09	0.342	1
436	Martin	11.10.2005	0.03	34.99	0.342	1
437	Martin	12.10.2005	0.16	34.43	0.603	0
438	Martin	13.10.2005	0.76	35.38	0.475	0
439	Martin	14.10.2005	1.29	35.3	0.612	0
440	Martin	15.10.2005	1.81	34.96	0.377	0
441	Martin	16.10.2005	2.53	34.62	0.318	0
442	Martin	17.10.2005	2.78	34.75	0.347	0
443	Martin	18.10.2005	3.03	34.36	0.318	0
444	Martin	19.10.2005	3.28	34.15	0.275	0
445	Martin	20.10.2005	3.53	34.02	0.373	0
446	Martin	21.10.2005	3.78	33.84	0.358	0
447	Martin	22.10.2005	4.03	33.66	0.373	0
448	Martin	23.10.2005	4.28	33.48	0.431	0
449	Martin	24.10.2005	4.53	33.3	0.409	0
450	Martin	24.10.2005	4.78	33.12	0.333	0

Figure 4.8: Matched animal data set in CSV format

4.1.3 Visualization of result

The matched data in the CSV form is converted into KML file to visualize in Google Earth software. In the map, icebergs are represented by arrow symbol. Wind direction is represented by orientation of symbols and sea surface temperature is shown by color of symbol (see figure 4.9).

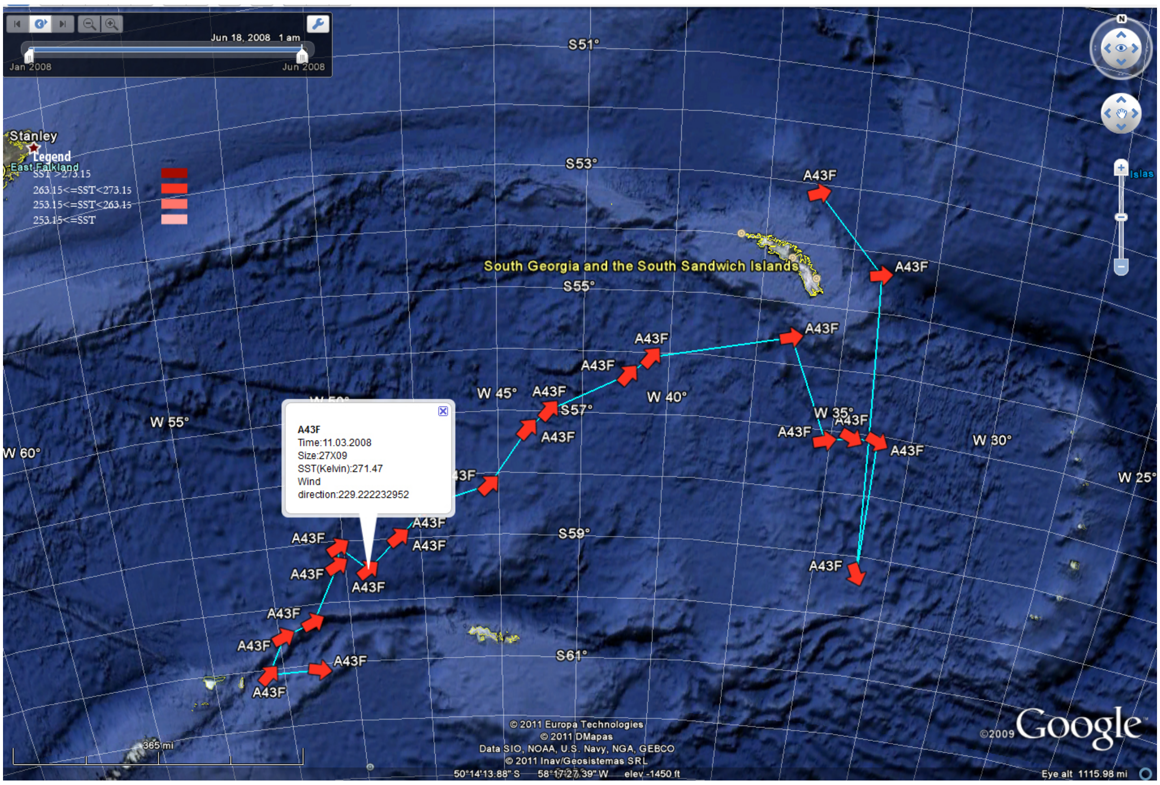


Figure 4.9: Visualizing matched datasets in Google Earth

4.1.4 Performance evaluation

The prototype was designed according to the users' need. It used both WCS and OPeNDAP standards to access environmental data. On the basis of time consumption for running the whole matching process the performance was measured. Python time module was used to calculate time consumed at each step of program execution. While using WCS for 342 trajectory records for icebergs in quadrant A during year 2008 total 1684 Kb of data was downloaded for each coverages. TPAC THREDDS server was used to access u and v component of wind. It took 00:00:32 seconds to perform whole process. Similar test was done for OPeNDAP. Same trajectory data for 342 records was used. It took 0:08:05 minutes to access server, subsetting, matching and extracting data with trajectory data. The test was done in personal computer with Intel Core 2 Duo CPU, 2.80 GHz processor with memory 4.00 GB in 32-bit windows operating system.

4.1.5 Conclusion

The conceptual framework was identified and implemented in this chapter. The architecture consists of three tiers they are: client tier, middleware and data servers. The work flow of client tier and middleware were described. The prototype developed during this phase is capable of getting inputs from users, accessing data and metadata from data providers, perform interpolation and matching. The matched data sets were saved in CSV file format. Later Google Earth software was used to visualize the result. The performance of the prototype was evaluated using the time benchmarking.

Chapter 5
Conclusion, Discussion and Recommendation

In this chapter findings of research are highlighted by answering the research questions. Also discussion on some issues identified during implementation are presented. In addition it provides recommendations for future works.

5.1 CONCLUSION

The research focused on identifying tools and methods for matching trajectory data with raster data using web servicess. In chapter 2, the need of data matching and interpolation methods were discussed. Likewise in chapter 3 technology for accessing remote data were highlighted. Two different standards (OGC WCS and OPeNDAP) were studied as protocols for accessing remote data. As a proof of concept, a prototype was designed and implemented. The prototype utilizes WCS and OPeNDAP protocols to access and subset large grid data. The result of matching can later be used for further analysis and visualization. The conclusions of this study can be summarized be providing answers to research questions as follows:

1. Who are the users and what are their requirements?
   The question has been answered in the sections 2.2.1 and 2.2.2. The users are different according to the use cases. Users of iceberg data with environmental data are navigators, planners, oceanographers, climatologist and biologist. Similarly, users of animal data with NDVI are landscape ecologists and conservationists. The requirements of users are described in the form of questions, they are:
   Oceanographers and Climatologists may need answers to the following questions:

   - What is the effect of environmental factors on iceberg calving?

   Navigators and engineers may need answer to following questions:

   - What is the current position of a particular iceberg?
   - How long does an iceberg stay at a particular location?
   - In which direction will be iceberg moving and what will be its speed?

   Landscape ecologists and conservationists need answers to following questions

   - What is current location of animal?
   - What causes animals to move or stop?
   - What is the effect of change in NDVI on movement of animals?

   All users need to see the relationship between changing position of the moving object with the influencing factors.

2. What are data and metadata options that are available through WCS and OPeNDAP?
   The envrionemtal data like sea surface temperature, ocean wind are generated from results of observation and simulations. These data are distributed by Earth Science and GI-sciecne communities using their own set of standards, OPeNDAP and WCS respectively. THREDDS WCS gateway has provides option to serve multidimensional grid data stored in different OPeNDAP servers though WCS standards. In this research data available through THREDDS server was used so that both WCS and OPeNDAP standards were used to access datasets.

3. What are WCS and OPeNDAP clients?
   In section 3.7 five WCS clients OWSLib, gvSIG, GDAL, Gaia, MPGC were analyzed on the basis of pre-defined criteria. They were interpolation, spatio-temporal subsetting, WCS version support, format of coverage data, extensibility, and integration into client server architecture, metadata/catalogue support and THREDDS/OPeNDAP support. The python OWSLib module was chosen as appropriate WCS client as it best meets predefined criteria. Similarly Pydap was chosen as OPeNDAP client as it can be used along with OWSLib in middleware built in python.

4. How to match metadata information?
   The issues of metadata were discussed in section 3.3.1 and 3.4. The metadata about data providers were checked using CSW client. Similary THREDDS catalogues were also checked for available data providers.

   Both WCS and OPeNDAP provides metadata information in their own formats. The metadata information was checked carefully before matching process starts. The prototype can be used to request metadata for particular data server.

5. How to match data with different spatio-temporal resolutions?
   Location and time (x,y,t) of given trajectory records was first read. Then index of same x,y,t was identified in the grid data. When indexes were discovered values associated with x,y,t were extracted and matched with trajectory data.The case was made simple by assuming trajectory data were collected in same time of the day. Due to complexity in interpolating trajectory position and priority of study, trajectory positions were not interpolated. Whereas grid values were interpolated by using inverse distance weighing method to find missing values.

6. How to measure uncertainty in the matched datasets?
   In section 2.5 the concept of uncertainty was highlighted. Uncertainty can be introduced to datasets at different levels during data matching. During the study uncertainty was not quantified due to limitation of time. However, users were notified if interpolations were performed.

7. How to disseminate the outcome of matching?
   The matched datasets need to be further analyzed in order to make decisions. The data was stored in CSV file format. This file format can be read by other analysis software or visualization tools for further processing and visualization purposes. In this study a CSV file was converted into KML format and visualized using Google Earth software.

8. What framework should be adopted for matching services?
   The conceptual framework for implementation of matching services was discussed in section 4.1. The three tier architecture was chosen to fulfill the users' requirements. The

prototype was implemented as a proof of concept. Different python modules were used to interact with the client interface, access data server, perform subsetting, extract values from grid data for a specific position, finding missing values in grid data, combining position data with associated values and finally providing matched datasets to users.

9. How to evaluate performance of prototype?
The performance of the prototype was evaluated in the section 4.1.4. Time consumption of process using OPeNDAP and WCS clients was tracked. While using WCS, NetCDF files need to be stored and read again for further manipulation. Whereas while using OPeNDAP, the manipulation was done on the fly. During the test 342 trajectory records were used.

5.2 DISCUSSION

Some of the issues that were identified during study period are discussed below:

- Using catalogues for finding data providers: CSW client was implemented to ease users finding data providers without prior knowledge about them. The appropriate external environmental data could not be discovered in WCS servers. Since most OPeNDAP has been operationally used for over 20 years, still many grid data are available through OPEnDAP/THREDDS servers. Alternatively data were searched in THREDDS servers maintained by ES communities. Most of the data used during testing were accessed from THREDDS servers that were discussed in section 3.5.

  THREDDS uses a hierarchical inventory structure in contrast the OGC, which uses a plain inventory structure. GALEON project was initiated by OGC and Unidata in order to make interoperable systems. The project implemented WCS gateway to serve data from THREDDS server using WCS standard. The data available as WCS on THREDDS server can be discovered using THREDDS catalogues but not using CSW catalogues. This gap can only be bridged by using mediator service. This component should be added in the catalogue as descibed by Nativi et al. (2009). This mediator component map the heterogenous service providers metadata models into a uniform data model. Hence metadata from both types of catalogues (THREDDS and CSW) can be discovered using single interface. Users can take advantage of this kind of system to explore large source of datasets required for data matching purpose.

- File formats: In the current implementation only NetCDF file format is used. This file format was selected as it is mostly used file format for outputs of climate and weather models. However there are several data formats like HDF, GRIB, GeoTIFF available through WCS and OPeNDAP. There is need for a separate reader for these different file formats to manipulate and extract values.

- Interpolation: The need of interpolation was highlighted in section 2.4. Both data sets i.e. trajectory and grid data are multi spatio-temporal in nature. So interpolation on both space and time was actually necessary. However during implementation grid data were interpolated to find missing values using inverse distance weighing method. The interpolation on time was not performed due to its complexity. So it was assumed that iceberg data were collected in the same time of the day.

- Value Extraction: In the section 4.1.2, the technique used to extract values from grid data for a specific point was explained. A moving object like iceberg can cover a large area and it may cover more than one grid cells. Whereas to extract values for moving object like animals, a single cell may also be enough. The iceberg's size was not considered during extraction of cell values.

5.3 RECOMMENDATION FOR FURTHER RESEARCH

The following recommendations are made for future implementation and research:

- The environmental data are maintained by both the ES and GIScience community. Interoperability has also been addressed by implementation of different standards. More study and implementation tasks should be performed to make a generic system that can discover data sources in servers using different standards. The gateway or mediator service is needed to make OGC clients capable of searching and accessing data that are provided by ES protocols i.e. OPeNDAP and THREDDS and vice-versa.

- In web based client server systems, many clients can access the server at once. In this study, the current system is not able to handle multiple clients. So, further study and implementation is required to address issues like user profile and session handling.

- The ultimate goal of scientists and researchers is to understand the effect of the external environment on moving objects. Only data matching cannot be used for decision making. Further study is required to couple data matching with other layers like analysis and visualization systems.

References

Alvares, L., Bogorny, V., Macedo, J. de, Moelans, B., & Spaccapietra, S. (2007). Dynamic modeling of trajectory patterns using data mining and reverse engineering. In *Tutorials, posters, panels and industrial contributions at the 26th international conference on conceptual modeling-volume 83* (pp. 149–154).

Anderson, I. (1983). Iceberg Deterioration Model. *Appendix C in Report of the International Ice Patrol in the North Atlantic, Bulletin*, *69*, 188–38.

Andrienko, N., Andrienko, G., Pelekis, N., & Spaccapietra, S. (2008). Basic concepts of movement data. *Mobility, Data Mining and Privacy*, 15–38.

Arrigo, K., Dijken, G. van, Ainley, D., Fahnestock, M., & Markus, T. (2002). Ecological impact of a large Antarctic iceberg. *Geophysical Research Letters*, *29*(7), 1104.

Beck, P., Wang, T., Skidmore, A., & Liu, X. (2008). Displaying remotely sensed vegetation dynamics along natural gradients for ecological studies. *International Journal of Remote Sensing*, *29*(14), 4277–4283.

Boone, R., Thirgood, S., & Hopcraft, J. (2006). Serengeti wildebeest migratory patterns modeled from rainfall and new vegetation growth. *Ecology*, *87*(8), 1987–1994.

Coyne, M., & Godley, B. (2005). Satellite Tracking and Analysis Tool(STAT): an integrated system for archiving, analyzing and mapping animal tracking data. *Marine Ecology Progress Series*, *301*, 1–7.

Death, R., Siegert, M., Bigg, G., & Wadley, M. (2006). Modelling iceberg trajectories, sedimentation rates and meltwater input to the ocean from the Eurasian Ice Sheet at the Last Glacial Maximum. *Palaeogeography, Palaeoclimatology, Palaeoecology*, *236*(1-2), 135–150.

Di, L., Chen, A., Yang, W., & Zhao, P. (2003). The integration of grid technology with ogc web services (ows) nwgiss for nasa eos data. In *NWGISS for NASA EOS data, in GGF8 & HPDC12 2003* (pp. 24–27). Science Press.

Di, L., & McDonald, K. (1999). Next generation data and information systems for Earth sciences research. In *Proceedings of the first international symposium on digital earth* (Vol. 1, pp. 92–101).

Di, L., Yang, W., Deng, M., Deng, D., & McDonald, K. (2002). Interoperable access of remote sensing data through NWGISS. In *Geoscience and remote sensing symposium, 2002. IGARSS'02. 2002 IEEE international* (Vol. 1, pp. 255–257).

Dodge, S., Weibel, R., & Lautenschutz, A. (2008). Towards a taxonomy of movement patterns. *Information Visualization, 7*, *3*(4), 240–252.

Domenico, B., Caron, J., Davis, E., Nativi, S., & Bigagli, L. (2007). GALEON: standards-based web services for interoperability among Earth sciences data systems. In *Geoscience and remote sensing symposium, 2006. igarss 2006. ieee international conference on* (pp. 313–316).

Domenico, B., & Lowe, D. (2009). Using python client libraries to access remote servers via Web Coverage Services. In *25th conference on international interactive information and processing systems.*

Fuglem, M., Jordaan, I., Crocker, G., Cammaert, G., & Berry, B. (1996). Environmental factors in iceberg collision risks for floating systems. *Cold Regions Science and Technology*, *24*(3), 251–261.

Gaia. (2010). *Gaia Geospatial Platform.* http://www.thecarbonproject.com/gaia.php/. (Accessed date: 2010.10.12)

GDAL. (2010). *Geospatial Data Abstraction Library.* http://www.gdal.org/. (Accessed date: 2010.10.27)

Gottschalk, K., Graham, S., Kreger, H., & Snell, J. (2010). Introduction to Web services architecture. *IBM Systems Journal*, *41*(2), 170–177.

gvSIG. (2010). *gvsig.* http://www.gvsig.org/. (Accessed date: 2010.10.25)

Ha, N. D. (2010). *Integrating grid and trajectory data via a web service: case study of iceberg movement*. Unpublished master's thesis, ITC.

He, B., Fang, T., & Guo, D. (2005). Uncertainty in spatial data mining. In *Machine learning and cybernetics, 2004. proceedings of 2004 international conference on* (Vol. 2, pp. 1152–1156).

Henley, S. (1981). *Nonparametric geostatistics*. Springer.

IDV. (2010). *Integrated Data Viewer.* http://www.unidata.ucar.edu/software/idv/. (Accessed date: 2010.11.27)

Ipopo, J. (2009). *Visualizing spatio-temporal data sets with different temporal resolution*. Unpublished master's thesis, ITC.

Johnston, K., Ver Hoef, J., Krivoruchko, K., & Lucas, N. (2001). *Using ArcGIS geostatistical analyst* (Vol. 300). Esri New York.

Jonsen, I., Flemming, J., & Myers, R. (2005). Robust state-space modeling of animal movement data. *Ecology*, *86*(11), 2874–2880.

Laine, V. (2008). Antarctic ice sheet and sea ice regional albedo and temperature change, 1981-2000, from AVHRR Polar Pathfinder data. *Remote Sensing of Environment*, *112*(3), 646–667.

Mountain, D. (1980). On predicting iceberg drift. *Cold Regions Science and Technology*, *1*(3-4), 273–282.

MPGC. (2005). *Multi-Protocol Geoinformation Client.* http://geobrain.laits.gmu.edu/mpgc/. (Accessed date: 2010.09.12)

Nativi, S., Bigagli, L., Mazzetti, P., Boldrini, E., & Papeschi, F. (2009). GI-cat: a mediation solution for building a clearinghouse catalog service. In *Advanced geographic information systems & web services, 2009. geows'09. international conference on* (pp. 68–74).

Nativi, S., & Domenico, B. (2009). Enabling interoperability for Digital Earth: Earth Science coverage access services. *International Journal of Digital Earth*, *2*, 79–104.

NOAA. (2010). *NOAA Earth System Research Laboratory.* `http://www.esrl.noaa.gov/`. (Accessed date: 2010.09.20)

NSIDC. (2010). *Atlas of the cryosphere.* `http://nsidc.org/data/atlas/ogc_services.html`. (Accessed date: 2010.09.10)

OGC. (2008). *Web Coverage Service(WCS) implementation standard, version 1.1.2, OGC* (No. 07-067r5).

OGC. (2010). *Open Geospatial Consortium.* `http://www.opengeospatial.org`. (Accessed date: 2010.10.27)

OPeNDAP. (2004). *Opendap user guide.* `http://www.opendap.org/user/guide-html/guide.html`. (Accessed date: 2010.09.27)

Pettorelli, N., Ryan, S., Mueller, T., Bunnefeld, N., Jedrzejewska, B., Lima, M., et al. (2011). The Normalized Difference Vegetation Index (NDVI): unforeseen successes in animal ecology. *Clim Res*, *46*, 15–27.

Pydap. (2010). *Pydap.* `http://www.pydap.org/`. (Accessed date: 2010.10.27)

Rhodes, H., & Myers, D. (1993). GRASS Used in the Geostatistical Analysis of Lakewater Data from the Eastern Lake Survey, Phase I. *Environmental modeling with GIS*, 438.

Ryan, S., Knechtel, C., & Getz, W. (2006). Range and habitat selection of African buffalo in South Africa. *Journal of Wildlife Management*, *70*(3), 764–776.

Shi, W. (2009). *Principles of Modeling Uncertainties in Spatial Data and Analysis*. Taylor & Francis Group.

Spaccapietra, S., Parent, C., Damiani, M., De Macedo, J., Porto, F., & Vangenot, C. (2008). A conceptual view on trajectories. *Data & Knowledge Engineering*, *65*(1), 126–146.

Stuart, K., & Long, D. (2009). Analysis of Antarctic Iceberg and Sea Ice Melting Patterns using QuikSCAT. In *Geoscience and remote sensing symposium, 2008. igarss 2008. ieee international* (Vol. 5).

Turdukulov, U., Blok, C., Köbben, B., & Morales, J. (2010). Challenges in data integration and interoperability in geovisual analytics. *Journal of Location Based Services*, *4*(3), 166–182.

Unidata. (2010). *Conventions for the standardization of netcdf files.* `http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html/`. (Accessed date: 2010.11.02)

Wang, G., Gertner, G., Fang, S., & Anderson, A. (2005). A methodology for spatial uncertainty analysis of remote sensing and GIS products. *Photogrammetric engineering and remote sensing*, *71*(12), 1423.