

1-1-2010

# Developing a web-based sea ice information system using GIS

Chenfeng Xiong  
*Ryerson University*

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Civil Engineering Commons](#)

---

## Recommended Citation

Xiong, Chenfeng, "Developing a web-based sea ice information system using GIS" (2010). *Theses and dissertations*. Paper 1010.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact [bcameron@ryerson.ca](mailto:bcameron@ryerson.ca).

# **DEVELOPING A WEB-BASED SEA ICE INFORMATION SYSTEM USING GIS**

by

Chenfeng Xiong

Bachelor of Computer Science,

Beijing Forestry University, P.R. China, 2006

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Civil Engineering

Toronto, Ontario, Canada, 2010

©Chenfeng Xiong 2010

## **AUTHOR'S DECLARATION**

I hereby declare that I am the sole author of this thesis or dissertation.

I authorize Ryerson University to lend this thesis or dissertation to other institutions or individuals for the purpose of scholarly research.

Signature \_\_\_\_\_

Chenfeng Xiong

I further authorize Ryerson University to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature \_\_\_\_\_

Chenfeng Xiong

# **DEVELOPING A WEB-BASED SEA ICE INFORMATION SYSTEM USING GIS**

Chenfeng Xiong

Master of Applied Science, 2010

Department of Civil Engineering

Ryerson University

## **ABSTRACT**

The Canadian Ice Service (CIS) has made available online access to some collections of its ice service archive. Currently, most of these historical ice information services are organized by a sort of static map images and web pages. Users can only view static result images of sea ice analyses or download raw sea ice data from the site. However, it is difficult for users to generate customized analyses without sophisticated information technology (IT). This thesis focuses on a sea ice information system (SIIS) designed specifically to improve the current CIS website with a possible method in developing web-based sea ice GIS system. The main achievements of the thesis are: (1) it accomplishes an SIIS design based on the popular web application principle known as Rich Internet Application (RIA); (2) it applies many current technologies around RIA and GIS to improve system quality; and (3) a prototype system is also developed to demonstrate how GIS may help in improving GIS-based online ice information services.

## ACKNOWLEDGEMENT

I would like to express my great appreciation and thanks to my supervisor, Dr. Songnian Li, for his thoughtful and valuable input, time, ideas and suggestions. I also greatly appreciate his efforts to provide me financial supports throughout my course and thesis studies. Without his patience, encouragement and understanding, this thesis would not have been possible.

I would like to thank Trudy Wohlleben, Steve McCourt, Richard Chagnon and Ziqiang Ou from the Canadian Ice Service, for providing me with the initial requirements and suggestions. Special thanks go to Dr. Ziqiang Ou for his constant consultation.

I also attribute my accomplishment to Dr. Jinyuan Liu, Dr. Ahmed Shaker and other faculty and staff members in the Department of Civil Engineering at Ryerson University, for their help and support. Many thanks for the help of Desmond Rogan, Mary Neelands, Kim Kritzer and Dianne Mendonca for their technical and administrative support.

Appreciation is extended to my fellow graduate students and friends at Ryerson University, Eric Chang, Xinbao Chen, Muhammad Butt and Zheng Niu for their help. The time with them was always happy and enjoyable.

Last, but not least, I wish to express my deepest gratitude to my parents for their unconditional love and support, and to my girlfriend Ting Liu, for her endless love, patience, and encouragement. Without their help and understanding, I would never have completed this thesis.

## TABLE OF CONTENTS

AUTHOR’S DECLARATION .....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT .....	iv
TABLE OF CONTENTS.....	v
LIST OF TABLES.....	ix
LIST OF FIGURES.....	xi
LIST OF ABBREVIATIONS .....	xiii
Chapter 1. INTRODUCTION.....	1
1.1 Problem .....	2
1.2 Objectives.....	5
1.3 Scope .....	6
1.4 Contributions.....	6
1.5 Thesis Organization .....	7
Chapter 2. BACKGROUND AND LITERATURE REVIEW .....	9
2.1 Sea Ice Data and Data Services .....	9
2.1.1 CIS Sea Ice Data.....	9
2.1.2 CIS Data Services .....	11
2.1.3 Other Sea Ice Data and Services .....	12

2.2	Current Development of Sea Ice Information Systems (SIIS) .....	16
2.3	Related Technologies .....	18
2.3.1	Rich Internet Applications.....	18
2.3.2	Human Computer Interaction (HCI).....	20
2.3.3	Web GIS.....	21
2.3.4	Other Related Technologies .....	23
2.4	Summary .....	28
Chapter 3.	METHODOLOGY AND DESIGN.....	31
3.1	Data Preparation .....	31
3.2	Methodology of Sea Ice Data Visualization .....	36
3.2.1	Viewing Ice Coverage.....	37
3.2.2	Animating Ice Coverage Changes.....	40
3.2.3	Graphing Trends.....	43
3.3	Methodology of Data Extraction.....	46
3.4	System Design .....	48
3.4.1	Grouping and Analysis of the Users of CIS.....	51
3.4.2	User Requirement Analysis.....	55
3.4.3	UI Organization .....	55
3.4.4	Considerations of System Structure .....	59

3.4.5	Considerations of System Implementation .....	60
3.5	Summary .....	61
Chapter 4.	SEA ICE INFORMATION SYSTEM PROTOTYPE DEVELOPMENT .....	63
4.1	Prototype Development.....	63
4.1.1	Prototype Components.....	64
4.1.2	System Configuration and Programming.....	75
4.2	Prototype Implementation .....	80
4.2.1	Overview .....	81
4.2.2	Mapping Sea Ice Coverage.....	82
4.2.3	Visualizing and Analyzing Ice Data.....	83
4.2.4	Accessing and Downloading Data.....	85
4.3	Simple Evaluation .....	86
4.4	Summary .....	88
Chapter 5.	CONCLUSIONS AND FUTURE WORK .....	89
5.1	Conclusions.....	89
5.2	Future Work .....	90
APPENDIX	.....	92
Appendix A:	Requirement From the CIS .....	92
Appendix B:	Weekly Regional Ice Chart Attributes.....	93



Appendix C:	Sea Ice Egging Code .....	95
Appendix D:	Table of Sea Ice Information Website Addresses.....	97
Appendix E:	Example Code Segments .....	98
REFERENCES .....		101

## LIST OF TABLES

Table 1.1 Current data provided by the CIS.....	3
Table 2.1 Types of CIS data .....	10
Table 2.2 Portion of weekly regional ice charts attributes.....	11
Table 2.3 Sea ice information websites (last accessed on March 24, 2010) .....	14
Table 2.4 Sea ice information system sites.....	17
Table 3.1 CIS data merge method comparison .....	33
Table 3.2 Techniques for sea ice data visualization.....	36
Table 3.3 Types of graphs plotted by Ice Graph tool (CIS, 2007) .....	44
Table 3.4 RIA based statistics advantages .....	46
Table 3.5 Possible users of CIS data.....	52
Table 4.1 Differences between proprietary and FOS software .....	70
Table 4.2 RIA platforms feature comparison matrix (re-written from (Roossien, 2009)).....	72
Table 4.3 Hardware specification of the prototype system .....	74
Table 4.4 Simple evaluation of the prototype system.....	87
Table A.1 Attributes of the CIS data table (CIS, 2009) .....	93
Table A.2 Egg coding for sea ice stages of development.....	96
Table A.3 Egg coding for floe sizes.....	96
Table A.4 Web addresses for currently running ice information system .....	97



## LIST OF FIGURES

Figure 1.1 3-step downloading of CIS current *.e00 data (CIS, 2003b).....	4
Figure 2.1 The nature of Human-Computer Interaction (ACM SIGCHI, 2009) .....	21
Figure 2.2 System architecture of ArcGIS Server (ESRI, 2009a).....	22
Figure 2.3 Relationship between web services, API, and mashup .....	24
Figure 2.4 Advanced SOAP-style workflow (zur Muehlen et al., 2005).....	25
Figure 2.5 Advanced REST-style workflow (zur Muehlen et al., 2005).....	26
Figure 3.1 Before and after manually removing redundancy from CIS data.....	34
Figure 3.2 Legend of sea ice concentration .....	35
Figure 3.3 RIA based web map service components .....	38
Figure 3.4 Operating procedure of thematic map tool .....	40
Figure 3.5 Sea Ice Animation Tool by NSIDC (NSIDC, 2010) .....	41
Figure 3.6 Sea ice extent trends from NSIDC (Climate4you.com, 2010) .....	43
Figure 3.7 The parameters of the Ice Graph Tool (CIS, 2009) .....	45
Figure 3.8 Modern SDLC (Valacich et al., 2009).....	49
Figure 3.9 Human-centered SDLC (Zhang et al., 2005).....	51
Figure 3.10 Possible actions by CIS information user .....	53
Figure 3.11 Possible actions by CIS map user .....	54
Figure 3.12 Possible actions by CIS map analyzer .....	55
Figure 3.13 User-centered UI design .....	56
Figure 3.14 Toolbox-style user interface structure .....	57
Figure 3.15 Menu-style user interface structure.....	58

Figure 3.16 Two possible web GIS system structures.....	59
Figure 3.17 Overall design of the system .....	62
Figure 4.1 Prototype WebSIS System components .....	64
Figure 4.2 GIS services in prototype system.....	69
Figure 4.3 Relationship between different kinds of software .....	71
Figure 4.4 Prototype system hardware components .....	74
Figure 4.5 Components of Sample Flex Viewer (Zhang, 2009).....	77
Figure 4.6 Sample Flex Viewer interface .....	78
Figure 4.7 The main interface of the prototype system.....	81
Figure 4.8 Prototype system mapping functions.....	82
Figure 4.9 Prototype system animation function .....	83
Figure 4.10 Prototype system Ice graph function .....	84
Figure 4.11 Prototype system thematic map function .....	85
Figure 4.12 Prototype system map management tools .....	85
Figure 4.13 Prototype system AOI cut and download tools.....	86
Figure A.1 Sea ice egg code definitions (CIS, 2009) .....	95

## LIST OF ABBREVIATIONS

ACM:	Association for Computing Machinery
AIR:	Adobe Integrated Runtime
AJAX:	Asynchronous JavaScript and XML
AOI:	Area of Interest
API:	Application Programming Interface
ARMAP:	Arctic Research Mapping Application
CARIS:	Computer Aided Resource Information System
CG:	Computer Graphic
CIS:	Canadian Ice Service
ESRI:	Environmental Systems Research Institute
FOS:	Free and Open Source Software
GeoRSS:	Geomatic Really Simple Syndication
GIS:	Geographic Information System
GP:	Geoprocessing
GUI:	Graphical User Interface
HCI:	Human-Computer Interaction
HCSDLC:	Human-centered System Development Life Cycle
HTML:	Hypertext Markup Language
HTTP:	Hypertext Transfer Protocol
IICWG:	International Ice Charting Working Group
IIS:	Internet Information Server
IT:	Information Technology
JCOMM:	Joint WMO/IOC Technical Commission for Oceanography and Marine Meteorology
JMS:	Java Message Service
JSON:	JavaScript Object Notation
KML:	Keyhole Markup Language
MB:	Mega Bytes
MSC:	Meteorological Service of Canada
MXML:	Multimedia Extensible Markup Language

NOAA:	United States Navy and National Oceanic and Atmospheric Administration
NSIDC:	National Snow and Ice Data Center
NZ:	New Zealand
OGC:	Open Geospatial Consortium
PC:	Personal Computer
POT:	Period of Time
REST:	Representational State Transfer
RIA:	Rich Internet Application
SAAS:	Software as a Service
SDK:	Software Development Kit
SDLC:	Systems Development Life Cycle
SIGCHI:	Special Interest Group of Computer-Human Interaction (in ACM)
SIGRID:	Sea Ice Grid
SIIS:	Sea Ice Information System
SMTP:	Simple Mail Transfer Protocol
SOA:	Service-Oriented Architecture
SOAP:	Simple Object Access Protocol
TF:	Time Frame
UI:	User Interface
W3C:	World Wide Web Consortium
WebSIIS:	Web-based Sea Ice Information System
WMO:	World Meteorological Organization
WMS:	Web Map Service
XML:	Extensible Markup Language

## Chapter 1. INTRODUCTION

Sea ice is formed on the ocean surface as a result of sea water freezing. During their respective winters, sea ice typically covers about 14 to 16 million square kilometers of the Arctic waters and 17 to 20 million square kilometers of the Antarctic by late in the season (IICWG, 2007). In Canada, sea ice coverage ranges from 1.92 to 4.20 million square kilometers (calculated from CIS weekly regional ice chart data of July 16, 2007 and Jan 1, 2007). Apart from sea ice, river/lake ice and ice of land origin (e.g., icebergs) are also presented in the Canadian waters. Icebergs are large masses of floating ice calved from glaciers. It is estimated that there are about 40,000 medium-to-large-sized icebergs annually in the Greenland (IICWG, 2007). Undoubtedly, ice represents a very important factor in the Canadian waters.

Ice conditions affect many marine activities. It is thus important to study climate change, environmental and ecosystem monitoring, and planning and construction of ocean structures. A report by the International Ice Charting Working Group (IICWG, 2007) states that ice information is vital and valuable for a variety of reasons. These range from marine transport to weather and climate change to natural resources to environment and habitation. There is a cogent need for ice information by many stakeholders such as mariners, scientists, the resource industry and northern residents. Ice services, initially created only to support local mariners, have since evolved into a wide range of sea ice information services developed to meet a plethora of users' requirements (WMO, 2006).

The distribution of ice information and services has been identified as one of the three focus areas that require special attention, with an emphasis on using the Internet to relay sea ice



information products and services to users (WMO, 2006). Many government agencies in countries such as USA, Canada, China, Russia, Japan, Denmark, Finland, Norway, etc., have provided ice services in various forms. These ice services contains operational ice information (real-time or near real-time) and some of them have historical ice data archives. Standards and data formats also have been developed to facilitate the sharing and exchange of sea ice data, e.g., SIGRID-3: A Vector Archive Format for Sea-ice Charts (IICWG, 2007) and Ice Chart Color Code Standard (WMO/TD-No. 1214). In Canada, the Canadian Ice Service (CIS) is responsible for providing ice services. Since 2006, the CIS began to produce digital Arctic regional sea ice charts in SIGRID-3 format.

This thesis focuses on developing alternative, web-based information services that allow easy access to and use of the CIS historical sea ice data that has been collected and analyzed over the past forty years. The sea ice archival data services are designed and developed using web GIS technology.

## **1.1 Problem**

The Canadian Ice Service, a branch of the Meteorological Service of Canada (MSC), is the leading authority for information about ice in the Canada's navigable waters (CIS, 2003a). CIS manages a vast collection of historical ice data. These archival data can date back to 1968 in digital vector format.

Current data of the CIS can be downloaded through a three-step procedure on the CIS's website (described in the next paragraph). The vector data of CIS are all stored in the format of GIS interchange file (\*.e00). Many studies, such as trend analyses of ice extent, have been done

based on this data. However, all of these studies can only be carried out on a local system or simply present analysis results on the Web. There are also some raster data available on CIS site. Table 1.1 illustrates the status of some existing CIS services which are based on their ice coverage data. The analysis images provided in these services are all produced and drawn by the technicians of the CIS. The reports are also written by the CIS technicians after analyzing data using local GIS software (for details about CIS data and services, see Section 2.1).

**Table 1.1 Current data provided by the CIS**

<b>Data Service Name</b>	<b>Description</b>	<b>Condition</b>
Daily ice hazard bulletins and charts	Describing ice conditions in active navigable waters of Canada	Raster image showing in static web pages
Daily iceberg bulletins and charts forecast	Forecast for Canadian waters south of 60N	Online access enabled simple text format
Weekly ice analyses of active ice areas	Analysis for strategic planning purposes	Charts in GIF image format with overview

Figure 1.1 combines three screenshots illustrating the current interface of downloading the ice chart data on the CIS website. This function can zip all the files requested for one-time downloading. However, if the amount or size of the files is too large, it may take a very long time to prepare the zip package and the page might look like frozen. Because the \*.e00 is an interchange file format and cannot be loaded directly by any current GIS software, after downloading the files, people who want to use these data have to convert the \*.e00 files to other formats.

**Step 1 - Collection selected:**  
 Select a **Collection** from the drop-down menu below, then click **Next Step**.

Weekly Regional Ice Charts - Black and White and E00 ▾

Next Step

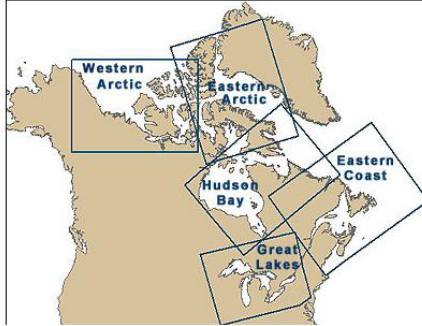
---

**Step 2 - Area(s) of Interest**  
 Select one or more **Area of Interest** from the drop-down menu below, then click **Next Step**.

☒ Eastern Arctic Regional  
☐ Eastern Coast Regional  
☐ Great Lakes Regional  
☐ Hudson Bay Regional  
☐ Western Arctic Regional

Next Step

Step 3 - Period




---

**Step 3 - Period**

1. Select whether you wish to specify a Continuous or Seasonal Period
2. Select the appropriate date/month/year information in the appropriate drop-down menus
3. Finally, click **Display Results**.

☒ Continuous:  
 From 2010 ▾ MAR ▾ 21 ▾ to 2010 ▾ MAR ▾ 21 ▾

☐ Seasonal:  
 From MAR ▾ 21 ▾ to MAR ▾ 21 ▾ in years 2010 ▾ to 2010 ▾

Display Results

**Figure 1.1 3-step downloading of CIS current \*.e00 data (CIS, 2003b)**

Data downloading is only one part of the problem. All of the analysis chart images can only be viewed after downloading (the GIF format data is available online but when user can see it, it has been downloaded to the browser cache). It is hard to explore ice coverage maps and impossible to do any customized analysis of the ice data online. In conclusion, the major problems of current CIS website in serving sea ice data are:

- Lack of online visualization tools which limits a user's ability to explore data;
- Lack of customized data downloading: users can only download data based on pre-defined regions in limited number of data formats; and
- Lack of analysis tools except for a slow and not-easy-to-use "Ice Graph Tool".

The need of a web-based system using GIS to support better access to the CIS Ice Service Archive is concluded from discussions through face-to-face meetings and email communications between CIS staff and the research group at Ryerson. For the initial exploration, the CIS would like to (see Appendix A for a complete list of requirements):

- Evaluate and select a data format for the database, develop a process for data conversion;
- Using two CIS Areas of Interest (AOI) design and develop a prototype database model; and
- Design and develop a comprehensive web-based querying tool for the Ice Service Archive that provides effective visualization techniques for web-based analysis and mapping of the ice data, simple graphing analysis, data extraction, and user-defined time-frames and AOIs (chart areas).

## **1.2 Objectives**

The overall objective of this thesis study is to design a web-based GIS information system to improve user experience and make more effective use of the over 40 years of historical ice data archived by the CIS. More specifically, the objectives of this research include:

- Categorizing user groups and analyzing user requirements generic to any sea ice data archives, including the CIS Ice Service Archive;
- Designing and structuring a web-based ice information system (WebSIIS) based on the above analysis;

- Developing a prototype WebSIIS to demonstrate how GIS may help improve online ice information services.

### **1.3 Scope**

This study mainly focuses on improving the service condition of online access to historical ice information through the CIS website. The design and development of the proposed web-based sea ice information is based on particular requirements collected from the CIS. However, because the sea ice data and the related ice information services share some common characteristics, the outcomes of this study should provide useful lessons for other similar developments. An initial discussion with the CIS staff indicates their preference of using proprietary software and platforms, if any, for such development. As such, no in-depth comparison study has been done to evaluate the proprietary and open software approaches for the proposed study. It is anticipated that using a properly selected set of open source tools the same objective may be realized.

### **1.4 Contributions**

The contributions of the thesis lie mainly in developing a multifunctional sea ice information system using RIA based web GIS techniques. The system provides some analysis functions within a web GIS interface which are commonly seen in web-based geographical information systems, but not existing in web-based sea ice information systems. The realized analysis function includes: trends animation, thematic mapping of ice coverage and statistics graphing. The system structure and building procedure should provide value to other similar system developments. To be more specifically, the contributions can be described briefly as follows:

- The research describing in this thesis builds up a sea ice information system integrating many useful functions. Since the system is building with both RIA and GIS technologies, the appearance and usability of the system is much better than other exiting systems in managing sea ice data.
- The development of a prototype system is another contribution. Some critical technologies such as human-centered principles and guidelines, RIA based web map service, spatial database, and GIS server are validated to be useful to build such a sea ice information system.

## **1.5 Thesis Organization**

In addition to this Introduction chapter, the main thesis body is divided into four chapters. The second chapter (Background and Literature Review) provides an overview of some current in-use SIISs and describes some major technologies used. Chapter 3 (Methodology and Design) provides an explanation of the detailed analysis and design information for the structure and method which the solution uses to solve the problem. The fourth chapter (Sea ice information system prototype development) describes the detailed development of the prototype system, including its components, coding and implementation. The final chapter of the thesis (Conclusion and Future Work) concludes the achievements addressed in this thesis and also indicates possible future directions for further study.



## **Chapter 2. BACKGROUND AND LITERATURE REVIEW**

This chapter reviews the current status of CIS data and services and some existing web-based sea ice information systems and sites. Attention is given to some of these reviewed sites or systems where web-based mapping or GIS plays a role in providing sea ice data and analysis services. To facilitate the understanding of the later chapters, some recent technologies that are related to the design and development of web-based SIIS are also introduced.

### **2.1 Sea Ice Data and Data Services**

Many countries around the world study sea ice. Information regarding sea ice is essential in many areas such as navigation, transportation, exploration of mineral deposits in the sea, the use of icebergs as a source of fresh water, and so on. Understanding the nature of sea ice and how it impacts the Canada's navigable waters is a major concern for the Canadian Ice Service (CIS, 2010). This section reviews the current CIS data and services as well as other similar sea ice data and services.

#### **2.1.1 CIS Sea Ice Data**

As mentioned in Section 1.1, CIS offers many kinds of data and services describing the information about sea ice in the Canadian waters. All of these services are based on a series of ice charts produced by a data collection team and a data production system.

Since last century, the CIS data has been collected by the CIS technicians through multiple sensors and other methods. The data available at CIS are summarized in Table 2.1. The data type "WMO" means that the data is available in color GIF image format following the standards from the World Meteorological Organization (WMO, 2006). "B&W" means that the data is



available in black and white GIF image and/or PDF format; "TXT" means that the data is available in text format; "XLS" means that the data is in excel work sheet format; "E00" means that the data is available in ESRI interchange (\*.E00) file format following the SIGRID-3 standard.

**Table 2.1 Types of CIS data**

Data Name	Time since	Update Frequency	Data Type
Daily Ice Analysis Charts	1999	Daily	B&W
	2003		WMO
Ice Thickness Data	1947	Uncertain	TXT
	1999		XLS
Daily Iceberg Charts	1968	Daily	B&W
Weekly Regional Ice Charts	1968	Weekly	B&W;E00
	2003		WMO; E00

The daily ice charts shows daily sea ice status prior to 6 p.m. every day. It is only available in GIF image format. The ice thickness data have two formats: data older than 1999 are in text format only, while the later data are available in spreadsheets. Daily iceberg charts only have data around the Newfoundland. The weekly ice charts are available in most formats, including a vector format of "E00". This is the only data that have a vector format from CIS. All these data can be downloaded through the current CIS website. The Weekly Regional Ice Charts are the data source used in this thesis project, so more information about them is given below.

### **Weekly Regional Ice Charts**

Table 2.2 lists some important attributes associated with the weekly regional charts (see Appendix B for a complete list of the attributes of weekly regional ice charts). It also describes the meaning of these attributes and provides three examples. Because the data is separated by region and stored in different files, each polygon in the data file has a "Region" attribute. In addition to the usual attributes such as area, creation date, ID, etc., there are some attributes

specially representing the ice type and conditions. The description of first level condition is done by attribute “A\_Legend”. It identifies the general type of ice, described by the polygons, indicating whether it is fast ice, remote egg represented ice, or normal land. The description of the second level condition is done by a formatted illustration symbol called an “Egg Code” (See Appendix C for more details), which is recorded by attributes starting with “EGG\_”. Any row in the weekly ice chart data table can be transferred into this form of “egg”, with the “egg” clearly showing the ice status of a specific area.

**Table 2.2 Portion of weekly regional ice charts attributes**

Field Name	Fast Ice Polygon	Ice Egg Polygon	Land Polygon	Description
AREA	802496.00	5.82 e^10	10.89 e^10	area of polygon
A_LEGEND	Fast ice	Remote egg	Land	legend name
REGION	CE	CE	CE	name of region*
PNT_TYPE	106	118	900	unique label id for polygon features
EGG_NAME	Q			name of egg
EGG_ATTR	9+_3_4_3_ @_7_.1_.7_ 5_@_@_3_3_ 3_@_@_@_			string value for egg attributes

\* BH = Baie d'HUDSON(Hudson bay) AE = Arctique de l'est(East Arctic) AO = Arctique de l'ouest(West Arctic) CE = Côte est(East Coast)

## 2.1.2 CIS Data Services

In addition to a variety of data described in the previous section, CIS provides many types of services. These services can be divided into four categories: ice charts, ice bulletins, ice forecasting and ice graph tool. The Ice Charts Service is a series of charts which can be downloaded, including:

- **Daily ice chart and image:** a chart analysis image generated from the daily chart data.
- **Regional ice charts:** an analysis chart generated from the weekly regional ice charts.
- **St. Lawrence River charts:** a special chart for St. Lawrence River.
- **Ice berg chart:** an analysis chart generated from daily ice berg chart.

All these services are ice status analysis images based on the different data provided by the CIS data production team. Most charts are available in both GIF and PDF formats. Ice bulletins service includes ice hazard bulletins and ice berg bulletins, both in English and French. These bulletins are all in TXT text format. An ice bulletin report usually describes some noteworthy points in everyday changing sea ice and ice berg. The ice graph tool is a web-based tool which can generate bar graph style statistics based on historical ice data. For every available area, information regarding the concentration of various stages of development of ice is extracted from all available Regional Ice Charts, and then summarized and stored in a database that is used by this tool (CIS, 2007). There are also some pre-generated ice graph images on some essential areas in daily ice services.

### 2.1.3 Other Sea Ice Data and Services

Besides CIS, there are many government agencies and institutions around the world serving ice information online through their website. Table 2.3 summarizes these sites with the types of data and areas covered. Some of these sites provide not only access to ice data but also to various levels of ice maps. In order to clearly describe this ice “mapping”, a classification of three kinds of web maps has been used: static raster image, raster image with animation, and vector map (Li, 2008). As for types of ice map services, four types of map services classified by Li (Li, 2008) are used. They are briefly described as follows:

- **Type A:** Data discovery and dissemination catalogue;
- **Type B:** Access, exploration and visualization;
- **Type C:** Data processing, query and analysis (a “(q)” sign indicating it has a query only);and
- **Type D:** Social mapping.

Table 2.3 also includes the host information of the site, types of the maps provided and the sea area covered. Information about the web address of each site can be found in Appendix D. Many sites include more than one kind of map type. In the table, map types are indicated by the following abbreviations:

- **SR:** Static Raster image;
- **AR:** Raster image with Animation;
- **VM:** Vector Map.

**Table 2.3 Sea ice information websites (last accessed on March 24, 2010)**

Name	Institution	Country	Map Type	Service Type	Sea Area
Canadian Ice Service	Government of Canada	Canada	SR,AR,VM	SR Type B AR Type A VM Type A	Canadian waters
Baltic Sea Ice Services	BSH Rostock	Germany	SR	SR Type A	North Sea and Baltic Sea
JCOMM Sea Ice Service	World Meteorological Organization	World Wide	SR	SR Type C(q)	Arctic Ocean
DMI Ice Charts	Danish Meteorological Institute	Denmark	SR	SR Type A	Greenlandic waters
SOK Ice Service	Admiral Danish Fleet Headquarters	Denmark	SR	SR Type B	Baltic Sea
Baltic Icebreaking Management	HELCOM	Baltic Sea States	SR	SR Type B	Baltic Sea
SMHI Polar View	Sweden's Meteorological and Hydrological Institute	Sweden	SR	SR Type B	Baltic Sea
FMI Ice Conditions and Polar View	Finnish Meteorological Institute	Finland	SR,AR	SR Type B AR Type A	Baltic Sea
Japan Ice Information Center	Japanese Maritime Safety Agency	Japan	SR	SR Type B	Sea of Okhotsk
Inland Water Information Centre	Inland Water Management and Waste Water Treatment	Netherlands	N/A	Text Report Only	North Sea
Arctic and Antarctic Research Institute	Russian Federal Service	Russia	SR	SR Type A	Eastern Arctic
National Snow and Ice Data Center	University of Colorado	USA	SR, AR,VM	SR Type C AR Type B VM Type A	World Wide
International Ice Patrol	United States Coast Guard	USA	SR	SR Type A	US waters
National Ice Center	United States Navy and National Oceanic and Atmospheric Administration(NOAA)	USA	SR, VM	SR Type C(q) VM Type C(q)	Alaska waters, Great Lakes and Antarctic Ocean
NZ Antarctic GIS	University of Canterbury	New Zealand	VM	VM Type C	Antarctic Ocean
University of the Arctic Atlas	University of the Arctic	World Wide	SR	SR Type B	Arctic Ocean
Arctic Research Mapping Application	ARMAP Organization	World Wide	VM	VM Type C	Arctic Ocean

With respect to the types of map services, these sites are divided into three classes. The sites in the first class only provide regular data downloading service without any advanced service such as data packing or AOI extracting. As well, some of them like the Netherlands Inland Water Information Centre only have a simple text report of daily ice condition. Others may have chart image or even vector-based ice data, but the vector data can only be viewed after downloading. These sites are marked as “Type A”.

The second class sites have some exploration and visualization functions for users to choose when downloading data or to get a quick preview. These sites provide some basic search or listing features and most of them also have some kind of overview or preview map when selecting data. Most of these sites provide on-line daily reports on real-time or near real-time ice conditions in the form of textual descriptions or raster image maps. These sites are marked as “Type B”, which include services covered by “Type A”.

The third class has web-based GIS services to provide more functions for the sea ice mapping. However, the service condition and scope are all different. Some sites provide raster data query service only, e.g., the National Ice Center and JCOMM sites. They both have an image-based search mechanism for users to quickly find their raster or vector maps. However, they do not provide any other GIS functions to allow users to further explore and analyze ice data. Other sites do implement a web-based GIS, providing vector or raster services. Examples of the later type of sites include: National Snow and Ice Data Center, NZ Antarctic GIS, and ARMAP. These sites are further reviewed in Section 2.2.

The sea ice web services and data sites (without web GIS service) mentioned above can all be called "static" websites and data services. These "static" sites have many inconveniences and shortages for users. The specific shortages are summarized in Section 2.3. The next section will continue to introduce some sites or systems using web GIS technology (Section 2.2).

## **2.2 Current Development of Sea Ice Information Systems (SIIS)**

Based on current web GIS technology, some sites and studies on sea ice are starting to migrate into web GIS or similar systems. This section introduces four cases of SIIS:

- Java based sea ice GIS tool
- National Snow and Ice Data Center
- NZ Antarctic GIS
- ARMAP

The "Java-based sea ice GIS tool" is a desktop program (Tang and Wong, 2006) and the other three are web-based systems available online. Table 2.4 shows similarities and differences between these sites or programs. Please note that the National Snow and Ice Data Center provides many data service but the only one relating to GIS is the "Sea Ice Index" service. The "Sea Ice Index" service only provides one GIS function, which is an animation of historical changes. It can create an animation of the extent of sea ice in a user-defined range. Nevertheless, the animation can only move between the overview extents of each historical map. That means if users want to see the historical difference for a small length of coastline, they have to download the data and view them using a GIS software tool.

**Table 2.4 Sea ice information system sites**

<b>Name</b>	<b>Language</b>	<b>Map Service Type</b>	<b>Data Process Support</b>	<b>Data Query Support</b>	<b>Data Analysis Support</b>
Java based sea ice GIS tool	Java	N/A	No	No	Map Overlay, Animation, Statistics
National Snow and Ice Data Center	Java Script	Image Server	No	Query by Historical Time Frame	Animation
NZ Antarctic GIS	Java Script	ArcIMS	Buffer	No	Measure, Identify, Set Units
ARMAP	Java Script, KML	ArcIMS	Buffer	Spatial Query	Measure, Identify, Set Units

The “Java-based sea ice GIS tool” is one kind of desktop program which supports data analysis operations but does not support any data processing or querying. This program is only a prototype program that studied the possibility and ways to analyze sea ice data easier. It is well known that a desktop program can be more powerful in data analysis comparing to the web systems. This program has various functions of analysis but lacks data management functions. Because it is a Java virtual machine-based program, the system performance is normally lower than a non-Java desktop program. The project had planned to convert its desktop solution to a web-based one, but there have been no publications indicating this has been done.

NZ Antarctic GIS and ARMAP are both based on ArcIMS, an Internet map server system provided by ESRI. ARMAP have three versions for web users (Java Script), Google Earth users (KML), and ArcGIS Explorer users (ArcGIS Explorer Configuration file). The Java Script version has almost the same functionality in comparing to NZ Antarctic GIS. The only difference between them is that ARMAP supports spatial query. These two systems are designed mainly for the scientists in studying sea ice. In terms of functions supplied by these sites, they both lack of analysis functions, like animation or statistics of trends, but they do have data processing



support, which is buffering. ARMAP also have a spatial query function in data management but this function can only be used in visualization not in data extracting. ArcIMS is a bypassing server program provided by ESRI. It is not updated anymore and has been replaced by ArcGIS Server products in the market. Its performance is slow compared to newer servers like ArcGIS Server.

In summary, the development of web-based SIIS is still in early stage, requiring more careful studies in its design, development and testing. All four identified systems provide just basic GIS functions, and are not compatible with those found in desktop programs. In addition, most SIIS reviewed here lack well-designed data management functions such as data extraction, data querying and data packing. This is surely an important part of the functions that a SIIS should have to improve data usage. The following section will introduce some most recent technologies to bridge the convenience of website and the power of desktop programs.

## **2.3 Related Technologies**

This section introduces the two most essential technologies in building a new style SIIS. One is Rich Internet Applications (RIAs) and the other is Web GIS. Related HCI (Human-Computer Interaction) principles and guidelines are also introduced because they have been applied, where appropriate, in the design and development of the proposed system.

### **2.3.1 Rich Internet Applications**

Generally speaking, Rich Internet Applications (RIAs) are applications that have many characteristics of desktop software applications. There are several definitions for RIAs, but in my opinion, Busch and Koch's (2009, p.3) description is most accurate:

“Rich Internet Applications (RIAs) are web applications, which use data that can be processed both by the server and the client. Furthermore, the data exchange takes place in an asynchronous way so that the client stays responsive while continuously recalculating or updating parts of the user interface. On the client, RIAs provide a similar look-and-feel as desktop applications and the word 'rich' means particularly the difference to the earlier generation of web applications. RIAs are basically characterized by a variety of interactive operating controls, the possibility of on-/offline use of the application, and the transparent usage of the client and server computing power and of the network connection.”

From the definition, the following characteristics of RIAs are significant compared to traditional web pages:

- RIAs are web-based but have desktop applications' appearance;
- RIAs are still working on client-server mode but many calculation and proceedings are moved to client side;
- RIAs are running synchronized with the server and only update the specific page sections when necessary;

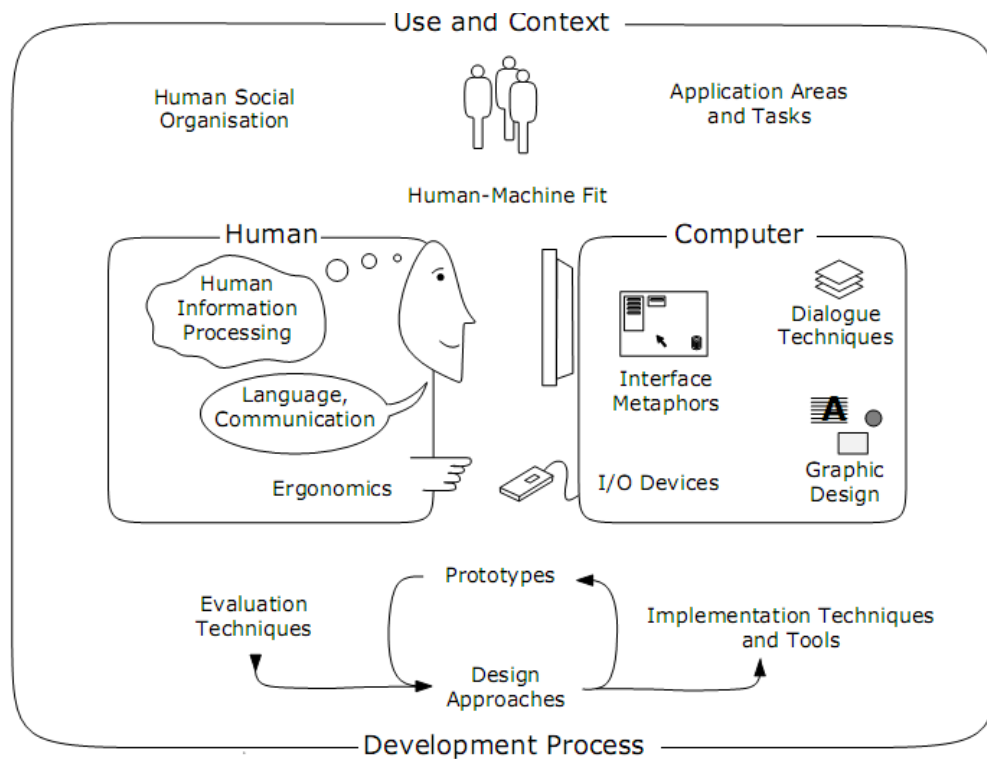
As it can be seen, “RIAs combine the Web’s lightweight distribution architecture with desktop applications’ interface interactivity and computation power, and the resulting combination improves all the elements of a Web application (data, business logic, communication, and presentation)” (Fraternali, et al. 2010 p.10).

There are two types of RIA frameworks – one is browser script-based and the other is plug-in-based. All browser script-based RIA frameworks are based on JavaScript AJAX technology. RIAs of this kind do not need a run-time framework support but their performance depends on different browsers. There are many different AJAX-based frameworks throughout the Web (e.g., Dojo, jQuery, Google web toolkit). The plug-in-based RIA framework is a “container framework”

(Busch and Koch, 2009), which means the application requires a run-time framework or web browser plug-in or virtual machine to run. RIAs of this kind have higher performance but users have to install plug-ins on their system or browser. Some popular RIA “container frameworks” are Adobe Flash/Flex/AIR, JavaFX and Microsoft Silverlight.

### **2.3.2 Human Computer Interaction (HCI)**

The study in Human–computer interaction (HCI) is actually the study of interaction between people and computer or we can say users and computers. The definition of HCI from ACM SIGCHI Curricula is described as: “Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them” (ACM SIGCHI, 2009). It often regards many fields of study such as computer science, behavioral sciences, art design etc. The study fields of HCI include user interface design and implement, human activity modeling, new interaction methods, interaction quality evaluation method etc.



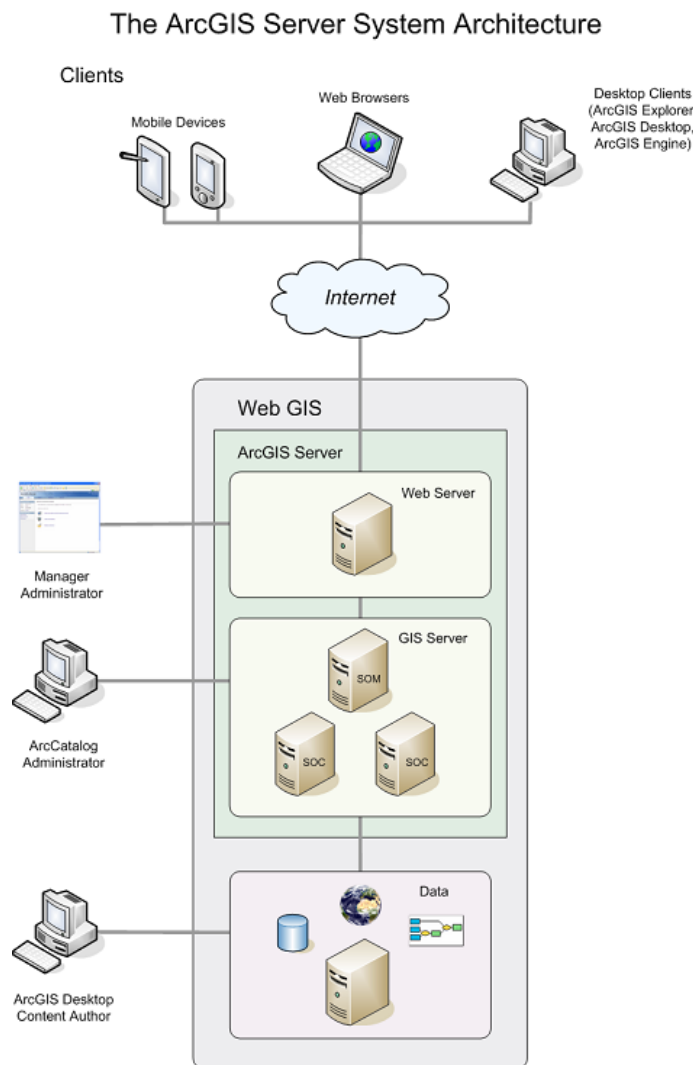
**Figure 2.1 The nature of Human-Computer Interaction (ACM SIGCHI, 2009)**

Figure 2.1 illustrates the typical study components of HCI. All the items included in the figure are fields that HCI studies. A basic goal of HCI is to make computers more usable and receptive to the user's needs to improve the interactions between users and computers. A long term goal of HCI is to design systems that minimize the gap between what users want to accomplish and the computer's understanding of the user's task.

### 2.3.3 Web GIS

A Web GIS is a GIS application which is made available through a common web browser. Typical GIS applications are normally huge and have CPU-intensive client side programs running on workstations or personal computers. Web GIS is usually developed to overcome system limits in delivering geographical information and free users from installing large GIS applications.

A popular web GIS system is the ArcGIS server system. The ArcGIS server is a proprietary product developed by ESRI. Figure 2.2 illustrates the system architecture of an ArcGIS server. An ArcGIS server is a combination of a web server, a map server and spatial data management. Developers or users can easily construct a customized web GIS system through the variety of tools provided by an ArcGIS server and related ESRI products such as ArcGIS Desktop.



**Figure 2.2 System architecture of ArcGIS Server (ESRI, 2009a)**

### **2.3.4 Other Related Technologies**

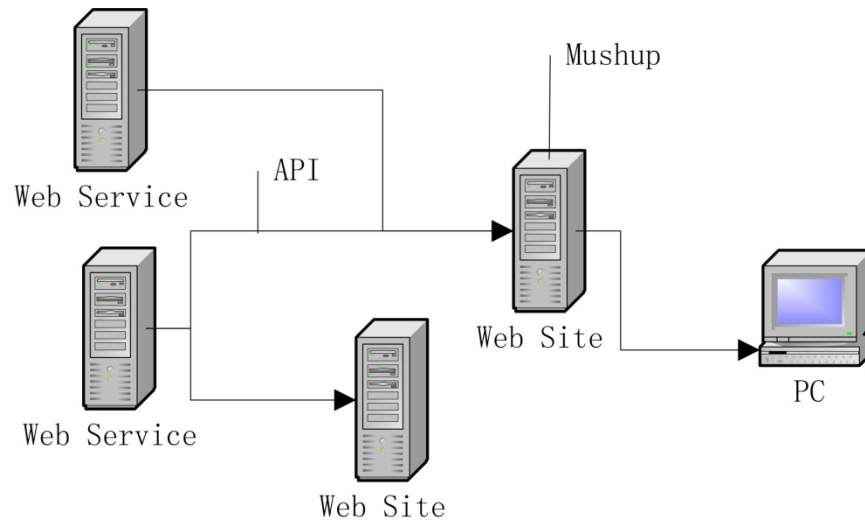
There are many recent technologies applied into the system such as Web 2.0, WMS, SOA, etc. With these new technologies, recent web GIS systems have far more power to handle geomatics data.

#### **1. Web 2.0**

Web 2.0 is a general concept that represents a website or service that has one or several of the following characteristics: User-centered Design, Crowd-sourcing, Web as Platform, Collaboration, Power Decentralization, Dynamic Content, SAAS, and Rich User Experience (Sharma, 2008). Web 2.0 is both a usage and a technology paradigm (Murugesan, 2007). From the usage point of view, web 2.0 offers a sort of service for users' communication and information updates. From the technology point of view, web 2.0 is a sort of websites with real-time interactive function and fast update rate; it also can be a platform for efficient service deployment and program development.

#### **2. Web services, API, and mashup**

A web service is a software system designed to support interoperable machine-to-machine interaction over a network (W3C, 2004). A web Application Programming Interface (API) is typically a defined set of Hypertext Transfer Protocol (HTTP) request messages along with a definition of the structure of response messages, usually expressed in an Extensible Markup Language (XML) or JavaScript Object Notation (JSON) format. A web mashup is a web page or website that combines information and services from multiple sources on the Web (Murugesan, 2007).



**Figure 2.3 Relationship between web services, API, and mashup**

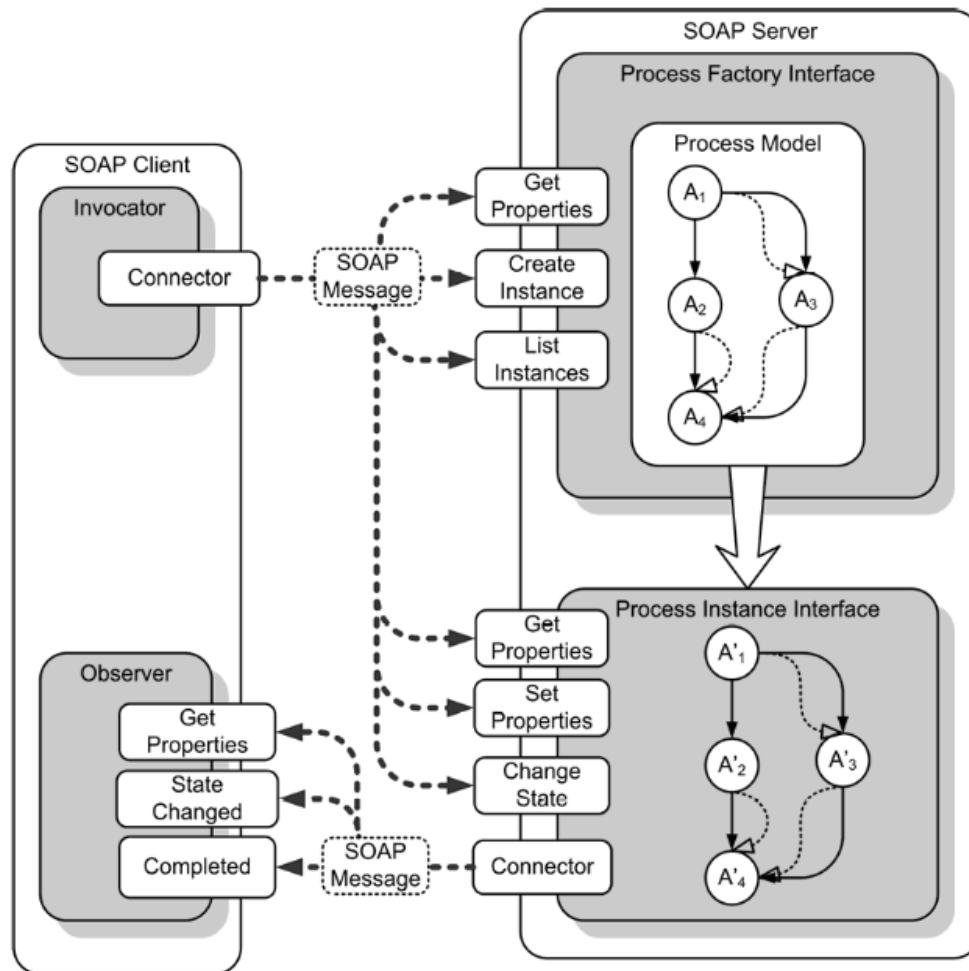
The relationship between web services, API, and mashup can be described by Figure 2.3. This figure shows two kinds of websites, one is normal website which consuming a web service, the other is a mashup site consuming two different web services.

### 3. SOA, SOAP, REST and Web mapping service

Service-oriented architecture (SOA) is defined as: an enabling framework for integrating business processes and supporting information technology infrastructure as loosely coupled and secure, standardized components - services - that can be reused and combined to address changing business priorities (Lawler, 2008). That means a project under this structure is always built upon a series of services and serves its users or client programs.

Simple Object Access Protocol (SOAP) version 1.2 is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols (W3C, 2007). Over the Internet, a SOAP message is transmitted through HTTP protocol, and also can be transmitted by other ways

(such as JMS or SMTP) (Li, 2008). This enables SOAP being widely used between servers and clients to extend the function of HTML. Figure 2.4 represents the general workflow of the SOAP structure. Each message contains all the orders or responses.

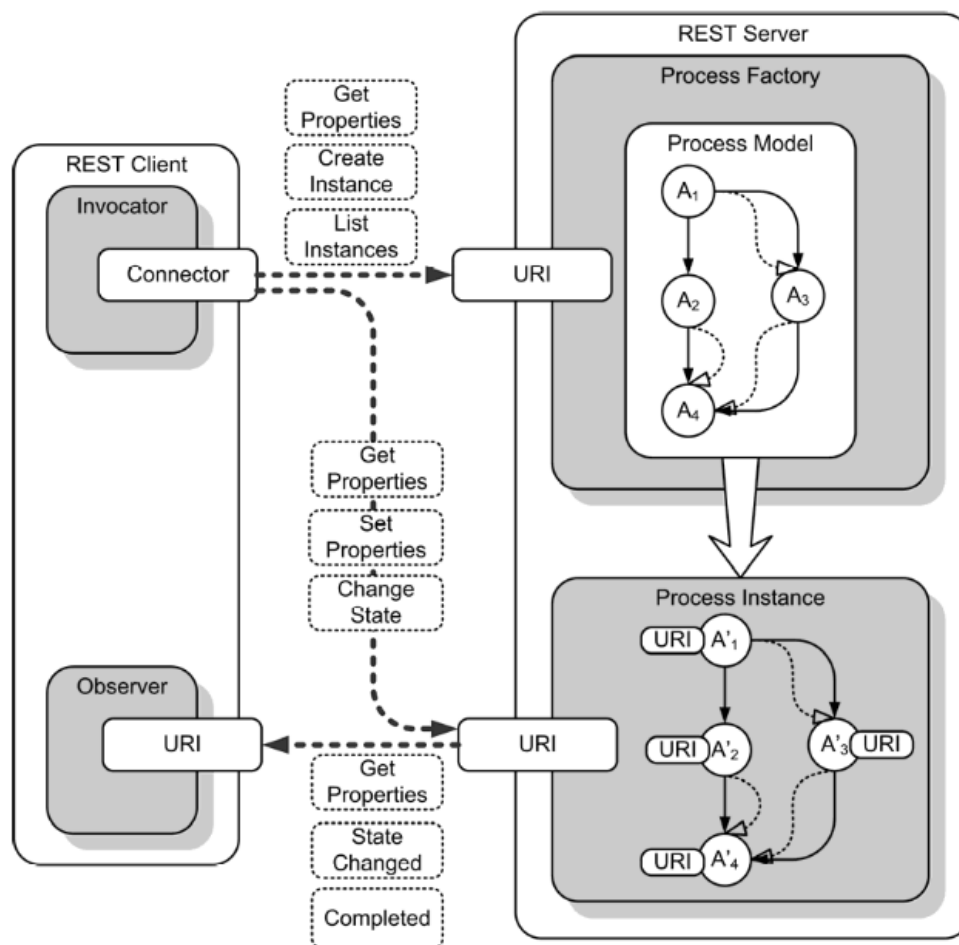


**Figure 2.4 Advanced SOAP-style workflow** (zur Muehlen et al., 2005)

Representational State Transfer (REST) is another kind of protocol like SOAP which enables the communication between client and server through HTTP. But as we can see in Figure 2.5, unlike SOAP, REST allows loose coupling of components to give more possibility and options for user to design the system. This is because of the different connection structure. In comparison with



the two figures (Figure 2.4 and Figure 2.5), we can see that REST can only send one kind of message to get one response while SOAP have to request all of them. Then we can easily come to a result that REST has lower connection bandwidth cost and more flexible components. Actually, the recent trend (so-called Web 2.0) has been away from SOAP-based services towards more direct style REST communications (Bell, 2008).



**Figure 2.5 Advanced REST-style workflow** (zur Muehlen et al., 2005)

#### 4. Web map service

Web map service (WMS) is a kind of electrical map service on the Web. It is also the name of a protocol defined by the Open Geospatial Consortium (OGC) for generic web map service

providing (OGC, 2004). There are many web map service providers on the Web, the most popular four are: Google, Yahoo!, Bing and MapQuest. All map service providers have their own on-line map exploring website and they all have some similarities and differences.

Google maps service is provided by Google.com. Google provides a fully functional web map service site. It supports many mapping functions such as traffic directions, location search and business area. Google also provides a service of “Street View” in most area of U.S. and some area worldwide. The “Street View” perspective joins many real-world photos of road scenery and takes visitor into a scene that looks like driving a car on the road. Besides web map, Google offers an application programming interface (API) called “Google Maps API”. With Google Maps API people can embed Google maps on third-party websites. Google web map uses the Mercator projection, so it cannot show areas around the poles. A related product is the Google Earth, a stand-alone program which offers more globe-viewing features, including showing of polar areas. Google Earth includes 3D overviews of major cities, mountains, and other terrains, as well as driving directions and maps. Additionally, it also includes Sky, which allows users to view stars and galaxies, and oceans, which allows users to explore the ocean floor and surface.

Yahoo! maps service is provided by Yahoo.com. Besides web map, Yahoo also offers a sort of API for embedding mapping functions into other web applications. Besides the similar AJAX API as Google, Yahoo! Maps API has more options for developers such as Action Script API, Simple REST and GeoRSS. With a variety of examples and help documents, developers can easily build up their own applications based on Yahoo! maps.

Bing maps service is provided by Microsoft Corporation. Besides functions such as search location, driving or walking directions, Bing map also provides traffic information in some downtown areas of North American city. The types of Bing Maps API are not as many as Yahoo offers, but also has an option among AJAX API, Silverlight API and SOAP API. The 3D version of Bing map is completely different from Google Earth or Google maps street view. Visitor can directly view the map completely from 3D perspective looking at the earth sphere.

MapQuest map service is provided by MapQuest Inc. Besides functions of web map such as locating, driving directions and nearby business searching, MapQuest also provides many cities' traffic, construction and even gas price information in U.S. MapQuest map also has a "360 View" perspective looks like Google's "Street View" but it does not support as many areas as Google does. A slight difference is that MapQuest can view the 2D vector map and the "360 View" at the same time while Google "Street View" can only display either the "Street View" or 2D vector map at one time. MapQuest also provides many options for developers. MapQuest map SDK supports almost all popular program languages or platforms like Action Script, JavaScript, Java, .NET and C++.

## **2.4 Summary**

Besides the CIS website, Chapter 2 reviewed some other sea ice data and service systems and also introduced some related technologies in building a SIIS. Through these reviews, a number of observations can be drawn.

First, in terms of data services, most online sea ice data sharing belong to service types A and B (see Section 2.1.3). That means most sea ice services are only based on simple web

technologies such as image viewing, file downloading and text description. For geographic information, such traditional ways are too abstract and hard to explore. Some of the better sites, like the National Ice Center website, used some simple map viewing systems so that users can interactively explore the data and make a quicker decision for data downloading. This kind of simple GIS tool does improve user experience in data service.

Second, in terms of data analysis services, most sites use only images to forecast, analyze ice conditions, or illustrate daily ice extent. However, if users want to view further information, like historical trends, they need advanced visualization and analysis tools such as GIS. One approach to solving this problem is what CIS has been doing: converting a specific range of maps into a GIF animation image so that users can view the changes of ice extent in some pre-defined ranges. This solution is not flexible when users want to see some user-defined ranges, but on the other hand, it has high performance for viewing. Another approach has been adopted by the National Snow and Ice Data Center. They use JavaScript to automatically change the map showing in the main window. This approach enables users to choose a customized time frame, but the performance still depends on the network speed. Other analysis tools, like buffering, are also provided by some sites using the Web GIS system. But there are also some limitations, including result formatting, map quality, etc. Generally speaking, the analysis functions in current sea ice GIS are far more simple than a desktop GIS system.

Third, the current CIS service system is a “static” site (see discussions in Section 2.1.3). This kind of system has the following limitations:

- ***Not straight forward***: data and services of the system are all listed as table or tree-view file lists; users have to download almost everything they need to view or analyze.
- ***Hard to explore***: with the amount of the data in the system growing on a daily basis, data management and visualization increasingly becomes a problem. The maintenance job could be very difficult for historical data.
- ***Hard to use***: data in a “static” system is stored as files. Thus, if users want to do some analyses on them, they have to collect and download all required data and then use some other desktop programs to analyze them. Even if a user only needs to find out the daily status of the sea ice, he or she has to open another map to confirm or find out the area name and then search in the data service.

In the end, by using current web GIS and other web technologies, a real-time online data exploring and analyzing system is possible to build and use in sea ice data management. This thesis intends to introduce an approach to building such a system and examine its benefits and limitations in providing online access of sea ice data.

## Chapter 3. METHODOLOGY AND DESIGN

This chapter describes the methodology and design in building a WebSIIS to improve online access to the CIS Ice Service Archive. At this stage, WebSIIS targets at providing three capacities: data visualization, statistical graphing, and data downloading/extracting. The scope of the design and development is also in line with the CIS preliminary requirements. Considerations are taken for later incremental expansion of the system.

### 3.1 Data Preparation

Ideally, a spatiotemporal database should be designed, developed and validated to hold all ice data in the archive. This is also expected by the CIS, according to the preliminary requirements provided (Appendix A). However, developing a spatiotemporal database is notoriously difficult and the work itself may require a separate thesis project. Since the ArcGIS server system was selected as the base platform for the development of the WebSIIS (see Section 4.1), the selected data was prepared directly into a geodatabase using current data files.

#### Existing CIS Ice Data Format

As mentioned in Section 2.1, most ice data from the ice service archive available on the CIS website are in static raster image format and cannot be used in a GIS system or tool. Only weekly regional charts are made available in vector format which can be used in a GIS system. In this thesis project, the weekly regional charts data were selected to design a data format and develop a process for data conversion because:

- The data has a set of characteristics shared by other types of chart data in the Archive;

- The current format allows easy conversion into a GIS format so that data visualization, management and analysis may be supported; and
- It allows the study to focus on system design and development, rather than preparing all data in the Archive.

The weekly regional ice data is stored in \*.e00 file format, which is the “ESRI interchange file” format. All ESRI GIS packages can read data stored in this format, but may not directly manipulate or process or analyze the data without any conversion. Other GIS packages on the market, such as CARIS GIS, can also import the data in this format. In other words, this is an old format frequently used in ESRI ArcGIS 8 or previous GIS systems but is not directly supported by current GIS systems. Because of this, it has to be converted to other formats before use.

The \*.e00 file can only be converted by a tool in ESRI ArcCatalog software and can only be converted into coverage file format. The coverage file can be converted to many other formats and fits most GIS systems on the market (e.g., shape file format). The coverage file format separately records vertexes, lines and polygons, and, as it is a format with large data redundancy, is not suitable for Internet use. Shape file format (\*.shp) is a file format developed and regulated by ESRI as a (mostly) open specification for data interoperability among ESRI and other software products (ESRI, 1998). It has good capability in ESRI software. In addition, as it was decided to use an ArcGIS server in the prototype system (see Section 4.1), the coverage file is then converted into shape file format to reduce data redundancy.

To use these weekly charts data in a GIS system, a complete map including all the Canadian waters is required to maximum user experience. It is apparently not convenient if the user has

to choose a region before viewing the ice maps. In addition, user cannot get an AOI covered two regions or more if the maps are not integrated. To present an integrated map to user, there are two ways. The first way is to directly import the original data into a spatial database and then develop a complex method to merge the maps automatically from different regions when using them. The second way is to merge the different area maps into one map first and then import it into a spatial database for direct use. Which one is more suitable is the problem to be discussed next.

Table 3.1 shows the pros and cons between the two methods. For the first method, the original data can be directly imported to the database. This makes it easy to update and add large amounts of data to the database. But because the results map is dynamically generated using some kind of translation program or middleware, it takes extra time to proceed every time the user needs to explore. This causes low system performance. In addition, dynamically-generated maps may have errors because it is almost impossible to test the maps one by one to find if the program is completely correct (to do so would cost much more than the other method).

**Table 3.1 CIS data merge method comparison**

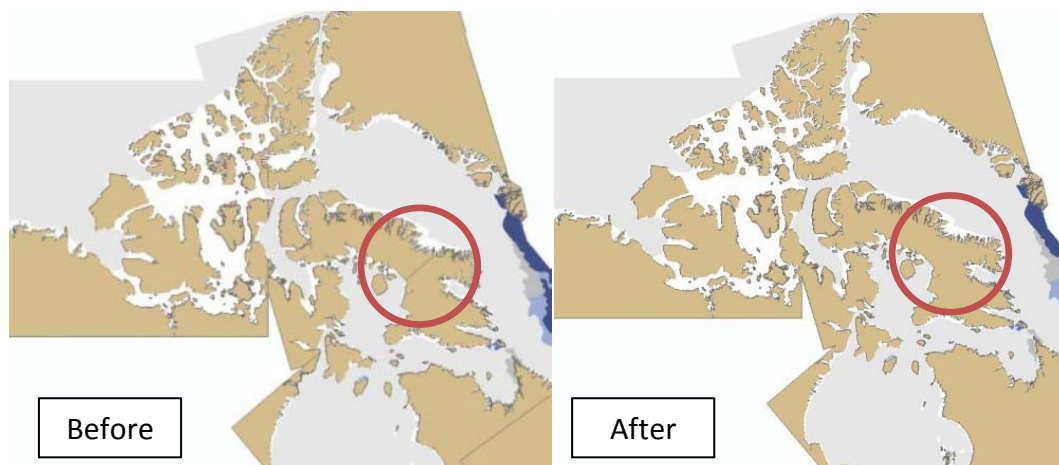
Method	Import to Database First	Map Merge First
<b>Advantage</b>	High speed data import and update	Fast data access No map error
<b>Disadvantage</b>	Slow data access May have some errors in the results map	Slow manual map merge and update

While the second method takes a long time to manually merge the maps before import, it prevents the occurrence of errors. After the map is merged and imported, a web GIS system can directly and quickly read the data from the database. Further, because the data need not to



be changed when use, special performance optimization is also possible. However, in this method, maps have to be merged before importing. This takes much longer time than the first method, so if the maps in the database need to be updated or expanded, it has to take longer to complete. Although each method has pros and cons, the second method was selected considering the combined benefits of loading speed and less errors, which are two important factors in using a GIS system.

The map is merged using the “merge” tool in the ESRI Arctoolbox. Because the data table should be merged together, some columns that are different in each area are of no use and are therefore deleted including the columns N\_980126# and N\_980126-ID (for column names, please refer to Appendix B). In addition, some overlying land polygons should be deleted to prevent unwanted cross lines in the map. Figure 3.1 shows the difference before and after some overlaying polygons are manually deleted. If there are some unwanted edge lines crossing the land, they should be removed manually.

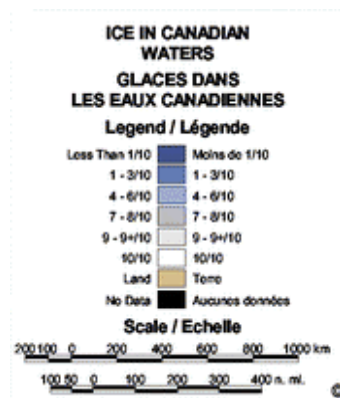


**Figure 3.1 Before and after manually removing redundancy from CIS data**

After map merging, the map data is ready to be imported into spatial database and used in a web GIS system.

The maps in Figure 3.1 have been colored with ice concentration attributes. What needs to be mentioned is that, before being used on the map server, the map needs a colored legend to make it comprehensible and carry more information. Figure 3.2 shows the map legend from the CIS. This legend has been applied to the maps before the map becomes a map service. The “N\_CT” column in the data table of the ice coverage maps is used together with this legend to determine the color of each polygon.

Other data-like ice berg charts, whether in images or texts, can also be managed by a relational database and integrated into the main SIIS to provide a full function of the current CIS site.



**Figure 3.2 Legend of sea ice concentration**

In conclusion, the data should be preprocessed before its use. This step is required for better system performance and management.

## 3.2 Methodology of Sea Ice Data Visualization

Data visualization facilitates visual data analysis, a method that has long been used by scientists in their research. Visual data analysis uses graphics, rather than texts, mathematical models and equations, to help people to explore data and develop ideas in their study (Lo and Yeung, 2007). There are many good techniques for visualization of geospatial data. Some basic ones were identified by Masuoka et al. (1996), including 2D/3D plots, 2D planimetric views, 3D perspective views, and animation.

The historical sea ice data in archives are spatial and temporal in nature and their visualization requires the use of different techniques. Table 3.2 lists some visualization tasks matched by possible techniques. GIS has long been recognized as a good tool for 2D/3D visualization of geospatial data including sea ice data. With increasingly more statistics functions being included in GIS, such as ArcGIS, the tasks listed in Table 3.2 can easily be realized.

**Table 3.2 Techniques for sea ice data visualization**

Visualization Tasks	Possible Visualization Techniques
Ice coverage characteristics according to attributes	2D planimetric views, e.g., generic GIS display functions, thematic mapping 3D perspective views, e.g., 3D surface modeling, 3D choropleth maps
Changes of ice coverage over years	Animation, e.g., 2D animation with time control and display, 3D animation of navigating through the map.
Ice coverage trend statistics	2D plots, e.g., line graphs, bar graphs, or linking graphs to map displays

Any visualization on a computer has a relationship with human-computer interaction (the techniques around it called HCI which is introduced in Section 2.3.2). Studying graphical user interface (GUI) is an important part of HCI in recent years. It is also important in GIS. The

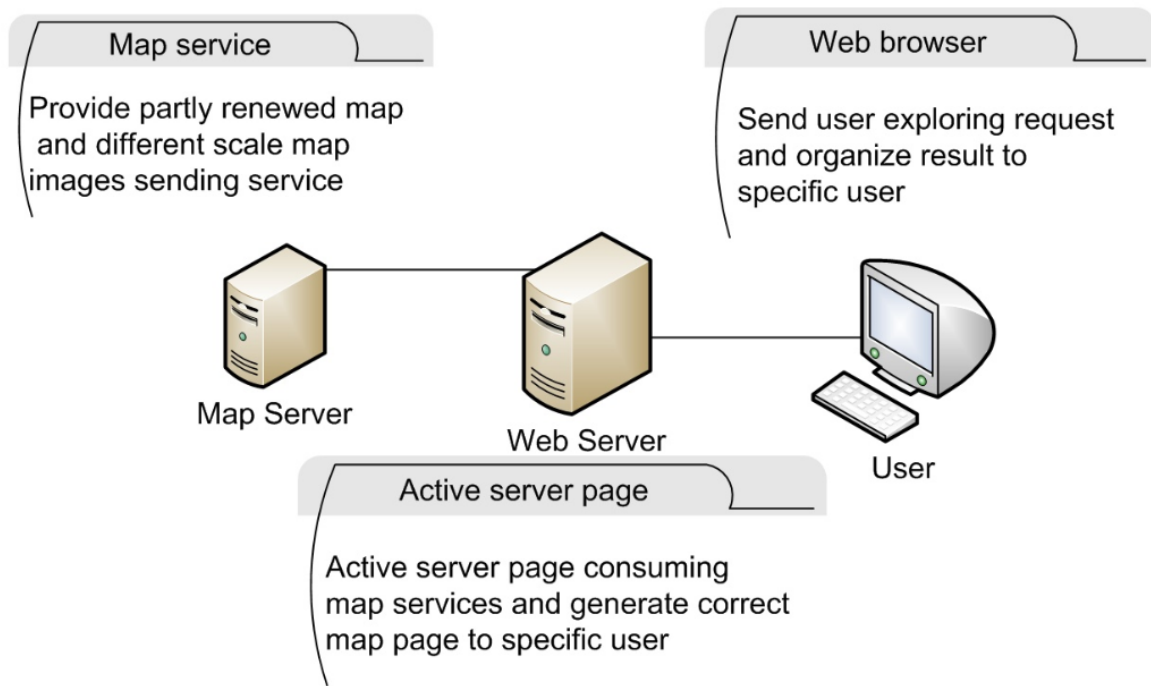
techniques in Table 3.2 are all based on GUI. GIS GUI development is highly dependent on the development of GUI (Lo and Yeung, 2007). The designs mentioned in this section use more recent techniques or developing tools to gain better GUI performance and a better HCI experience for the user.

Many GIS software can easily present all visualization tasks listed in Table 3.2 onto a desktop computer. However, no matter how powerful the current GIS software is, when it is on the Web, everything becomes different. As reviewed in Section 2.2, most web systems related to sea ice still use the basic level of 2D planimetric display. Few systems have thematic maps that can be freely changed by the user; animation functions that can be freely customized; and statistic tools that are dynamic and real-time. Thus, the visualization design of the WebSIIS in this thesis is divided into three parts: viewing ice coverage, animating ice changes, and graphing trends. At the end of each part, there are also discussions regarding the improvements or benefits of each specific design compared to the current CIS website are presented.

### **3.2.1 Viewing Ice Coverage**

After preprocessing, the map data is still a vector map that can only be viewed in desktop GIS software. It needs a map server to convert the data into a map service for further use. With a map server, a vector map can be viewed on the web page just like popular map services such as Google Maps and MapQuest. The overall functions of map servers, web servers and user browsers are shown in Figure 3.3. When a user starts to explore a map, the browser first sends its request to the web server. Then the web server sends a request for the required map image parts to the map server. The map server dynamically adjusts the correct scale and/or projection,

extracts necessary data parts and sends it back to the web server. Finally, the web server generates a new page for the client browser (Active Server Page) or directly sends necessary image segments to the client browser (partly refreshes the page).



**Figure 3.3 RIA based web map service components**

The result presented to the user is a free exploring map just like Google maps or Yahoo! maps. This style of map visualization has three main benefits compared to a site with only static images:

- ***Straight forward:*** As everyone knows, a colored map is the most easy-to-understand style in complex data expressing. In addition, a free exploring colored map can maximum the expressing power of a colored map and present a straight-forward ice concentration distributing status to the user.

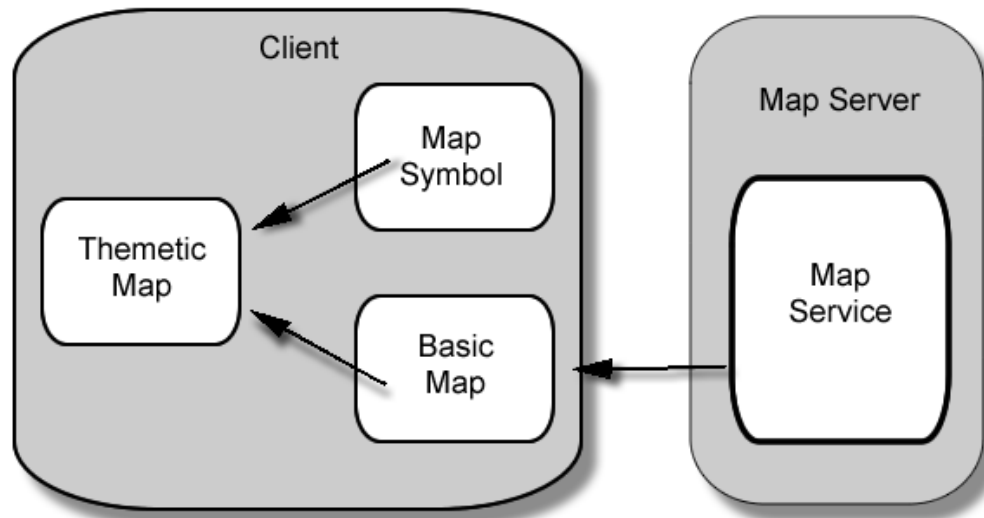
- **Easy to use:** Most people experience in using some kind of online map service like Google Maps. The same exploring mode can let people become familiar with the system quickly and feel comfortable using it.
- **Suitable for most users:** As a public service site, the GIS website may have thousands of different visitors every day. Each user has totally different levels of knowledge and GIS skills. Using the most popular exploring mode is the best fit for most people, as anyone with even rudimentary skills can easily and quickly find what they want.

As a result to the analyses above, visualization using web map service style becomes a good choice for sea ice data. This also provides a good basis for the toolbox style control panel which is further discussed in Section 3.4. With dynamic map service, some advanced visualization functions become possible. Since 3D visualization on the Web is not good at supporting complex surfaces with large amounts of data, the most commonly used 2D plot, especially customizable thematic maps, is selected to show an example of advanced map visualization of ice conditions.

Thematic mapping is a very commonly used method in geographic visualization. It is used to portray spatial variations and interrelationships of geographical distributions of a particular theme using different colors or map symbols.

To produce a thematic map in a GIS system, a definition of the thematic symbol is first required. Next, the symbol is applied to one column of data in the data table of a GIS map. As Figure 3.4 shows, the process of thematic mapping can be mostly done on client-side. The client can apply

various symbols for one map and change it instantly when needed. The direction of the arrow in Figure 3.4 represents the data streaming direction when a thematic map is being generated.



**Figure 3.4 Operating procedure of thematic map tool**

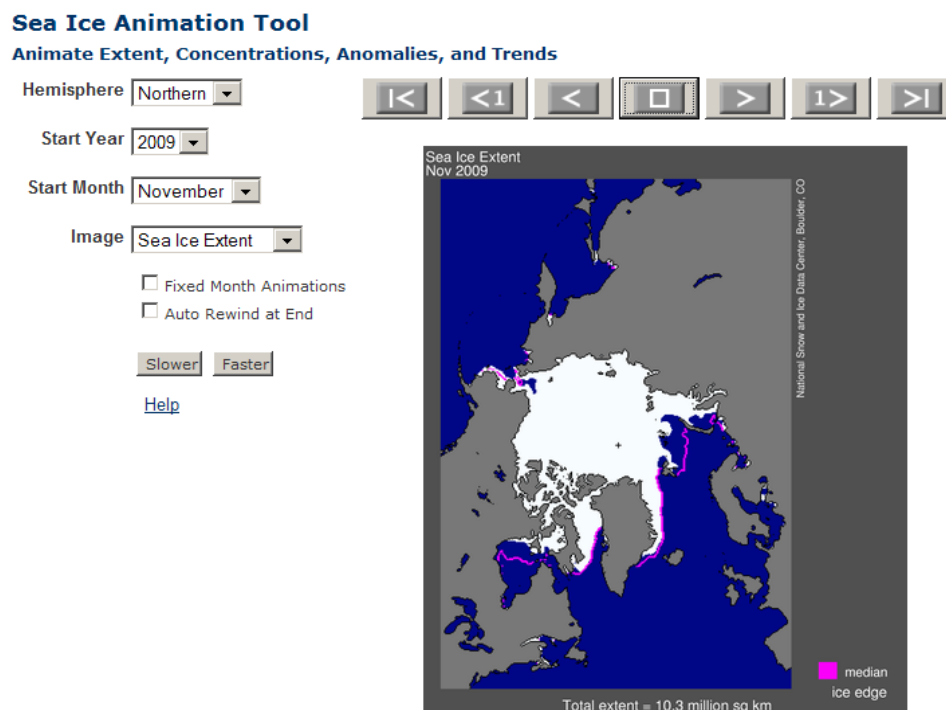
It is worth mentioning here the benefit of RIA applied on the thematic mapping part of web SIIS. With RIAs' powerful graphical function, the thematic map can even be animated if the client-side computer is powerful enough to change the symbology of a map in less than 0.1 second. This function can certainly allow for better visual analyses of ice trends.

### **3.2.2 Animating Ice Coverage Changes**

Animation is a kind of computer graphics (CG) technique for time-related spatial data visualization (Lo and Yeung, 2007). Animation between different coverage maps at the same place at different times can highlight all the changes of the specific area and show trends. Generally, animation in GIS can be divided into 2D and 3D. 2D animation usually focuses on the animation through time. The most commonly used 2D animating technique is to animate the

transparency or visibility of layers. 3D animation is hard to realize on the Web because of the high performance requirements not only on the client's PC but also at the network connection. In most cases, huge data transforming is needed to execute a 3D animation. Therefore, both the good preload technique and the vector and model based real-time 3D rendering technique may partly solve the huge data set problem. However, this is another research topic and cannot be easily implemented into WebSIIS.

There is only one sea ice animation tool generated by NSIDC available online. The interface of that tool is shown in Figure 3.5. This tool is actually a half-automated image viewer. When user presses the play button, it shows the images in the predefined sequence, one by one. However, user can neither change the order of the sequence nor animate the images when zooming in.



**Figure 3.5 Sea Ice Animation Tool by NSIDC (NSIDC, 2010)**

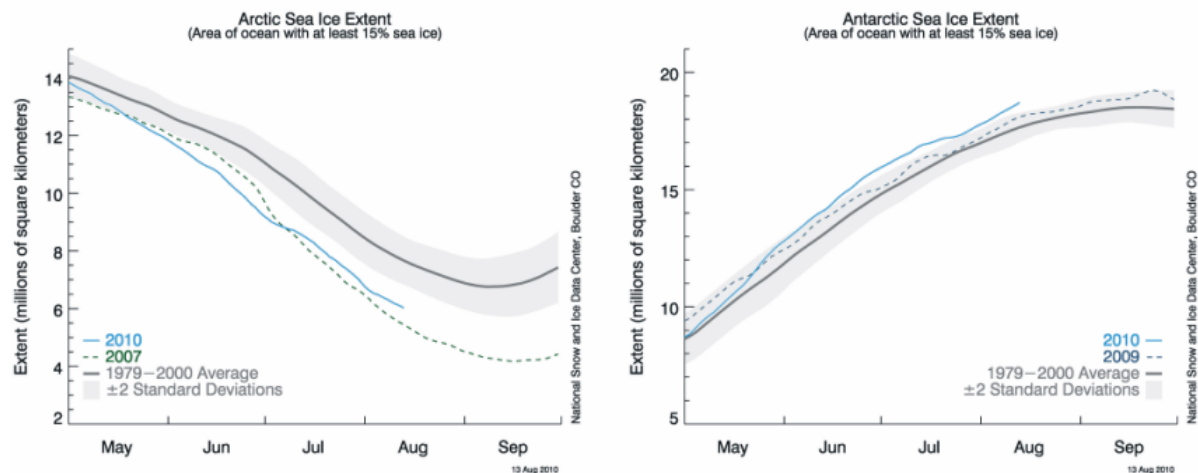


2D animation is generated by continuously changing similar images within a certain time. This is what is done in coverage animation, similar to videos in VCD, DVD or online sites like YouTube. No matter what format of video the player is working on, a drag-enabled status bar and a play and pause button exist. It is a common assumption that people always feel comfortable in using something similar to what they are already familiar with. Thus, it is better to design the animation function just like a video player. A freely drag-enabled player can easily compare two or more maps in the animation sequence. Therefore, for a better user experience and a higher analysis value, a time line bar with pause and drag function is required. To be straight-forward, the animation tool is designed only to have a status bar and a play/pause/stop button with preloaded maps. It directly controls the displayed maps in the main interface. User can either drag the bar to see the difference between specific maps or let it play in a certain time delay to see the changes. The preloaded maps can make the playing easier.

Compared to the animation tool of the NSIDC, the WebSIIS animation tool focuses more on the user's intention through the status bar. With the dragging function of this bar, user can easily compare any two of the images in a sequence using a simple mouse click without any delay. Another advantage is that the tool looks more like a normal video player and is easy for users at any level to understand and use. The final advantage is the flexible sequence. The WebSIIS animation tool image sequence is linked to the main map view. So it can benefit from the design of the "view list" mentioned in Section 3.4. That means the play sequence of the tool can be customized easily and freely to any available data in the database. This is very flexible for visually analyzing changes in the archived ice data over the past 40 years.

### 3.2.3 Graphing Trends

Statistical graphs are mainly plotted in the form of a line graph, bar graph, pie graph, scatter plot, or parallel coordinate plot. Some of these graphs have been widely used for visualizing the trend of ice coverage over a period of time. Figure 3.6 exemplifies such use of line graphs.



**Figure 3.6 Sea ice extent trends from NSIDC (Climate4you.com, 2010)**

The examples use statistical graphs in visualizing ice coverage changes. They can clearly show ice extent trends and comparison of some historical data. However, all these trend statistics are static images generated by professional researchers or technicians. They cannot be changed anymore when published. The Ice Graph tool provided by CIS is a different one.

The Ice Graph tool is an online program used to generate statistics about ice conditions in the Canadian waters. Statistics are computed from a number of pre-defined regions. The data is based upon the Canadian Ice Service Digital Archive – the Regional Charts collection. For every available region, information about the sea ice is extracted from all available Regional Ice Charts, summarized, and stored in a database that is used by this tool.

Table 3.3 shows the specific types of statistics graphs that the Ice Graph tool creates. Each kind of statistics chart has two types: "total concentration" and "stage development". Total concentration describes the overall ice coverage percentage within a given time frame, and the stage development shows the developing trends during the same time frame.

**Table 3.3 Types of graphs plotted by Ice Graph tool (CIS, 2007)**

Bar graph type	Description
Weekly Ice Coverage for Season – Total Concentration	Each bar represents the total amount of ice present expressed as a percent of the total sea area for the given area and date (week). The graph spans only one (1) season.
Weekly Ice Coverage for Season – Stage of Development	Same as the Total Concentration graph above, except that the bar is subdivided in colors representing the different stages of development. The graph spans only one (1) season
Historical Ice Coverage for a Single Date – Total Concentration	Each bar represents the total amount of ice present expressed as a percent of the total sea area for the given area and date (week). Useful for comparing the same week across multiple seasons.
Historical Ice Coverage for a Single Date – Stage of Development	Same as the Total Concentration graph above, except that the bar is subdivided in colors representing the different stages of development. Useful for comparing the same week across multiple seasons.
Historical Weekly Ice Coverage – Total Concentration	Each bar represents the total amount of ice present expressed as a percent of the average sea area for the given area and date (week). The graph may span multiple seasons.
Historical Weekly Ice Coverage – Stage of Development	Same as the Total Concentration graph above, except that the bar is subdivided in colors representing the different stages of development.
Historical Total Accumulated Ice Coverage – Total Concentration	For "Total Concentration" outputs, each bar represents the average amount of ice present expressed as a percent of the total sea area for the given area and season.
Historical Total Accumulated Ice Coverage – Stage of Development	Based on the above model, the next thing is to decide the way to express the statistics.

As illustrated in Figure 3.7, in order to generate an ice graph, user only needs to fill the options (left image). All types of graphs are generated after the options are set. Users can view or download any one of each in the resulting page (right image).

Step 1 - Region Selection

CIS EA Baffin Bay

CIS EA Baffin Bay: E Baffin Bay

CIS EA Baffin Bay: NW Baffin Bay

CIS EA Baffin Bay: W Baffin Bay

CIS EA Eastern High Arctic

CIS EA Eastern High Arctic: Eureka - Norwegian Bay

CIS EA Eastern High Arctic: Jones Sound

☐ Display short region names

Step 2 - Ice Seasons and Dates Selection

Historical Period:

Start Season: 1968

End Season: 2008

Ice Season (mddd):

Start Date: 1022

End Date: 1225

Ice Season for "Weekly Ice Coverage for Season" Graph:

1982

Date (mddd) for "Historical Ice Coverage for Single Date" Graph:

0709

Step 3 - Optional Parameter Selection

Period for normals calculation:

Start Season: 1971

End Season: 2000

☐ Show warning for interpolated data

☐ Shade interpolated data

☒ Plot average

☐ Plot median

☐ Plot trendline

☐ Scale graph to data values

Select the graph size: 1230x780

Reset Defaults

Generate Graphs

Links To Products

CIS EA Baffin Bay

Weekly Ice Coverage for Season 1982, 1022 - 1225

Tabular Data:

[TXT](#)

[CSV](#)

Total Concentration

Stage of Development

Historical Ice Coverage for Single Date 0709, 1968 - 2008

Tabular Data:

[TXT](#)

[CSV](#)

Total Concentration

Stage of Development

Historical Weekly Ice Coverage 1968 - 2008, 1022 - 1225

Tabular Data:

[TXT](#)

[CSV](#)

Total Concentration

Stage of Development

Historical Total Accumulated Ice Coverage 1968 - 2008, 1022 - 1225

Tabular Data:

[TXT](#)

[CSV](#)

Total Concentration

Stage of Development

New Request

**Figure 3.7 The parameters of the Ice Graph Tool (CIS, 2009)**

The current Ice Graph system is based on a relational database with data extracted from the weekly charts' data table. To generate any graph, users have to set the parameters every time and regenerate the graph every time by setting the parameters again. They have to click the mouse at least three times and wait for two times of page refreshing (or three if user wants to change the target region) to get a single graph. This is step-by-step process is easy to follow, but may not be efficient.

To make this complex statistical graphing system look better, the choice is still RIAs. With RIA's powerful graphic function, a programmer can easily develop programs that allows user to generate various kinds of charts, tables or histograms. Thus, applying the above rules (i.e., the Ice Graph rules) into the program can let it dynamically generate statistical results and even create animations.

Compared to non-RIA static graphics, another RIA's advantage is that everything can be dynamic. Because RIAs can do whatever desktop programs do, it is not hard to realize a dynamic graphical statistics tool which can instantly respond to any change of the parameters

made by a user. Some specific advantages of RIA-based statistics graphing compared to current static graphing are listed in Table 3.4.

**Table 3.4 RIA based statistics advantages**

Item	Static graphing	Dynamic graphing
Appearance	Simple and crude	Rich and beautiful
Freedom level	Average	High
Responding time	Slow	Fast
Other	Not straightforward if people do not understand some parameters	People can play with the parameters and see instant results

As shown in Table 3.4, RIA has a good appearance and a fast response time. The freedom level means how much and how freely the parameters in the statistics tool can be set. Although, theoretically speaking, they both can include many parameters, dynamic graphing can instantly show the results, which makes it possible to test many different settings quickly. Thus, the freedom level is higher than static graphing. For the same reason, getting instant results without having to wait is more straightforward in dynamic graphing, and users may find more interesting groups of statistics or parameters to study.

### 3.3 Methodology of Data Extraction

Data processing and access is an important part of GIS data management. There are many different data sets on the Internet and at various sea ice centers in countries around the world. Without a GIS-based solution, the analysis of sea ice data requires users to gather and download data to local computers and then use professional GIS software to process and analyze them. The difficulty level of data access is determined by the difficulty level of the data searching, downloading and extracting, or data format conversion. Currently, there are three main ways to access data from a sea ice data centre:

- ***Directly view and download:*** the data usually is in raster image format. Vector data may be viewed by map services but cannot be downloaded directly through map services.
- ***Query the data online and download:*** the query mostly is TF-based. In many cases, users have to download the data one by one if the original data is not stored in one file. Some data centers, like the CIS, have a data packing service to pack the data before downloading.
- ***Order the data through mail, e-mail or an order system online:*** not all data are accessible online. In many cases, experts have to put in their order with the data centre to obtain the data they want. For example, most data on the NSIDC has to be ordered.

As mentioned in Section 1.1, the current CIS data access is mainly based the online query and downloading method. At the same time, ordering through a subscriber system is also provided for special needs (CIS, 2003c). To deal with the problem mentioned in Section 1.1, a fast and convenient data-access method becomes imperative. This thesis introduces a solution package in data access which includes three complementary parts:

- AOI (area of interest)-enabled dynamic data extraction;
- Fully customized zip package for one-time fast downloading; and
- Server-side real-time format conversion based on the user's preference.

First of all, the AOI means the area smaller than the complete ice coverage area that the user is interested in. The user may only want to download a part of the ice data. This process involves data extraction, which may be too computing extensive and require too much network data transmitting to be performed on the client-side computer. As such, assigning data extraction

tasks to the server-side based on user defined AOI is more efficient. For example, letting a user specify his/her AOI as a rectangle and sending the rectangle's parameters to the server invokes a server-side process of data extraction.

The current CIS site has a function of creating zip files before downloading. But this function cannot temporarily save the zip results for downloading if the downloading process fails or some other interruptions occur. If the data amount is large, the zip procedure may take a long time; likewise, if the users refresh the web page, they have to do the zip process again. Two design considerations were taken in the design of the proposed Web SIIS: visually displaying the progress of the zipping process to the users and temporarily storing user-defined zip parameters as well as a copy of zipped files.

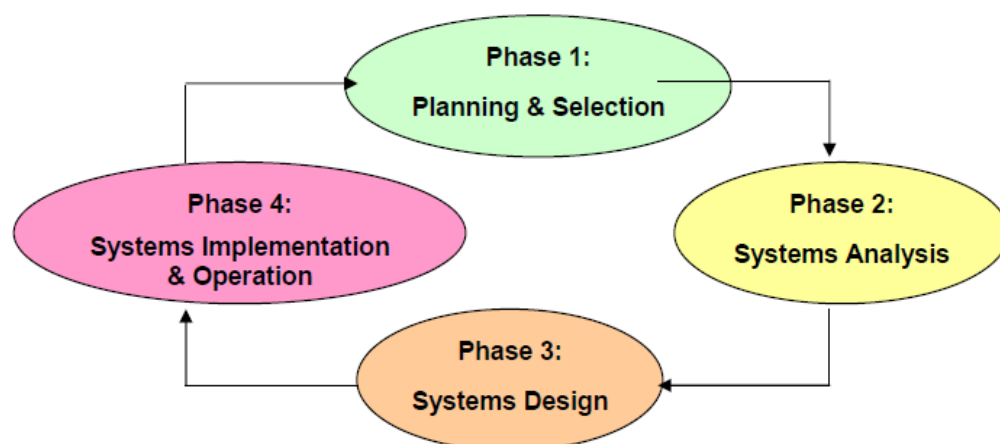
In addition to data downloading, it is often the case that users may need the data in a format different from the default format the data center produces. On-the-fly data converting or reformatting certainly helps more advanced users. With current GIS server technologies, this becomes a fairly easy task, requiring less coding but more system configurations. For example, SAIF and ArcGIS Server may be integrated to utilize their rich data transformation and conversion functions to support ice data extracting into different formats.

### **3.4 System Design**

There are many different development methods for an information system. Of these various approaches, the systems development life cycle (SDLC) model is a commonly accepted one for describing the complex issues involved in information systems development, as it captures the spirit of the systems development process (Hoffer et al., 2005). In this consideration, the

developing process of WebSIIS mainly follows the modern SDLC model but refers to some useful considerations of the Human-centered SDLC (HCSDLC) by Zhang et al., (2005) to improve user experience.

Most modern information systems are produced following SDLC or similar models. Figure 3.8 is a representation of the modern SDLC model structure. The four phases in this life cycle describe the development procedure of an information system. A typical system development procedure should first plan and select the necessary goals that the information system should achieve. Then, analyze these goals and design an executable structure and sectors to fulfill the requirements gained in phase 1. After finishing the analysis and design, the system proceeds to the implementation and operation phase and comes out as a prototype system. This prototype system is then examined by the users or goal providers and they decide whether it is acceptable or needs to be planned again. After some iterations of this cycle, an acceptable information system is born and becomes ready to use.

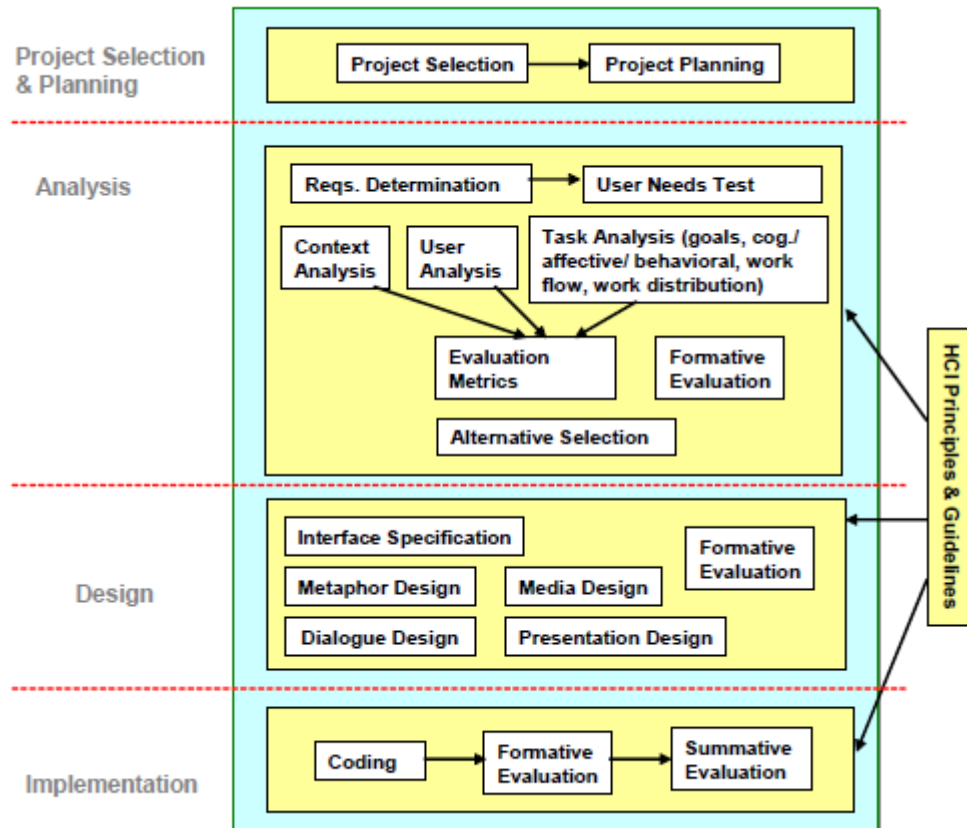


**Figure 3.8 Modern SDLC** (Valacich et al., 2009)



Usually, the user interface (UI) design is a small part of the system design. In many systems development approaches, including the recent ones, the HCI issues and concerns – if even covered at all – are not considered systematically (Zhang et al., 2005). Very often, screen layout, menu design, buttons, colors and other interface features are considered in UI design, but users of an information system may run into problems that are beyond the interface level. Illogical organization of data/information in the system, lack of control over the system, or misfit between the nature of the task and the support provided are all problems a user may encounter even when the UI is well-designed. This calls for a human-centered system analysis and design.

HCSDLC is an improved SDLC by fully integrating HCI principles and guidelines into modern SDLC. Figure 3.9 shows the development methodology of HCSDLC. Different from SDLC, HCSDLC considers user needs by applying HCI principles at the beginning of the system analysis. It applies HCI principles and guidelines to every aspect of system analysis, design and implementation. An information system produced under this SDLC would be human-centered and very user-friendly. However, since there is no chance to connect with the real user of the CIS data, the detailed proceeding of the SIIS design only applied HCI principles and guidelines into user analysis, each section of detailed design and final implementation.



**Figure 3.9 Human-centered SDLC (Zhang et al., 2005)**

To apply the user-centered HCI principles to the WebSIIS, the user's needs should firstly be considered. In the following part of this thesis, grouping and analysis of the user needs of the CIS data are addressed to help better apply the HCI principles. This is only a simple analysis without a formal user survey, but still has some reference value in system analysis and design. Then, two issues when considering HCI are discussed. One is the UI organization design and the other involves considerations in system logical structure and design.

### 3.4.1 Grouping and Analysis of the Users of CIS

Possible users of CIS data from the CIS website are listed in Table 3.5. To model and group the users into a more general structure for easier program design, three groups are divided

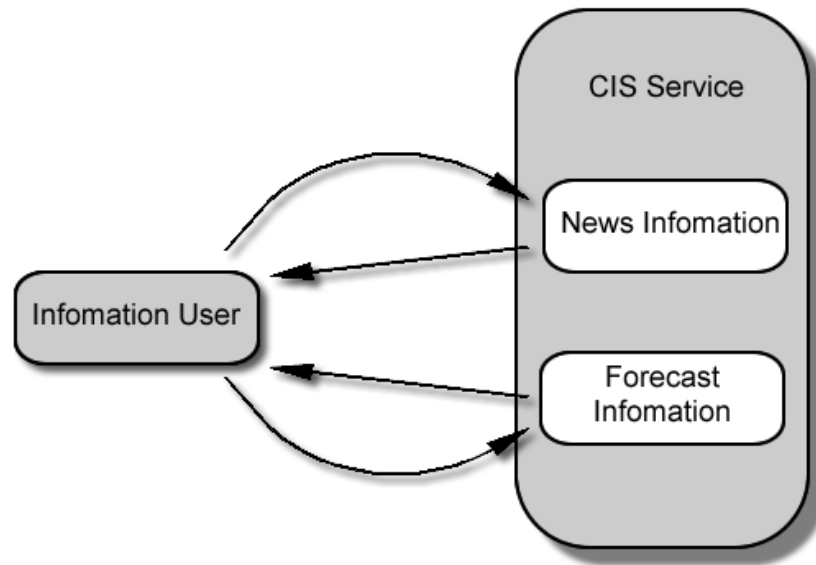
according to the users' IT skills and how often/how much they need to use GIS analysis functions. The groups are: information user, map user and map analyzer. The following explains how to define these group and analyzes the needs of users in each group.

**Table 3.5 Possible users of CIS data**

User	Description
the Canadian Coast Guard	weather and ice information for marine safety, icebreaking operations and efficient marine transportation
port authorities	site-specific information on current and long-term ice conditions in ports and shipping routes
commercial shipping companies	information for strategic and tactical vessel passage planning
fishermen and fishing fleets	en route and on-site ice conditions for ice-encumbered areas
offshore oil and gas companies	iceberg and sea ice information for exploration and production, both on-site and in transit
mining industry	ice conditions along coastal areas and supply routes
construction industry	site-specific current and historical data for offshore and onshore projects, such as bridges and port facilities
tourism and recreation industry	technical and general information for the operation of cruise ships and the enjoyment of passengers
marine insurance companies	risk assessment for offshore operations affected by ice
environmental consultants	data, analysis and expert advice for impact assessments
research scientists	Ice information for research on transportation, construction, climate change, meteorology, oceanography, biology and socio-economic impacts.

### 1. Information user

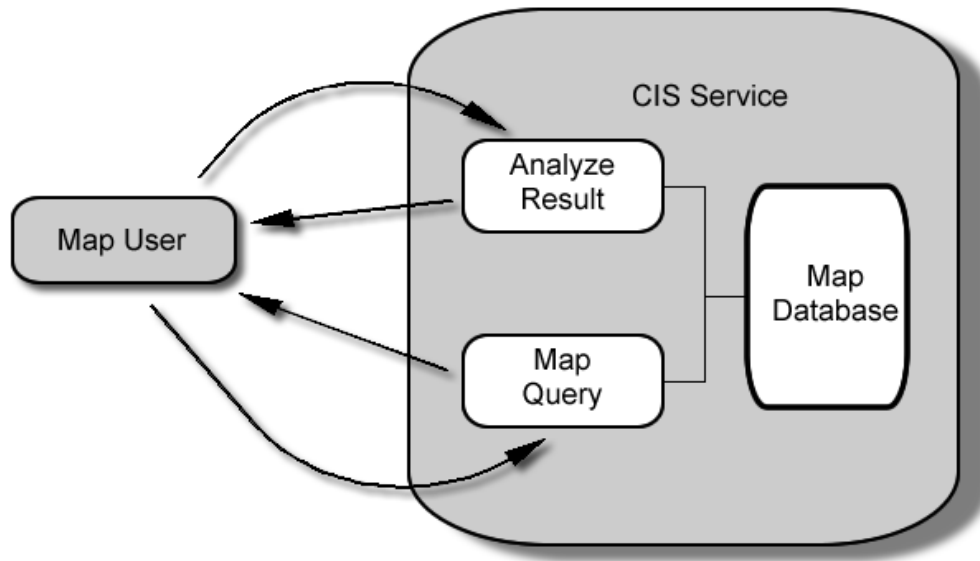
An information user is a general viewer of the system. Information users need only to see the result of some predefined analysis or news reports. Figure 3.10 shows the information users' action. Information users are simple information explorers and they may have low or medium IT skills. All they want is some information. Coast guards, port authorities and fishermen can be divided into this group. They may only be interested in the condition of the ice on a specific day or in short-term (a few days ahead) ice forecasting. So the need of them may only include simple data and information exploring.



**Figure 3.10 Possible actions by CIS information user**

## **2. Map user**

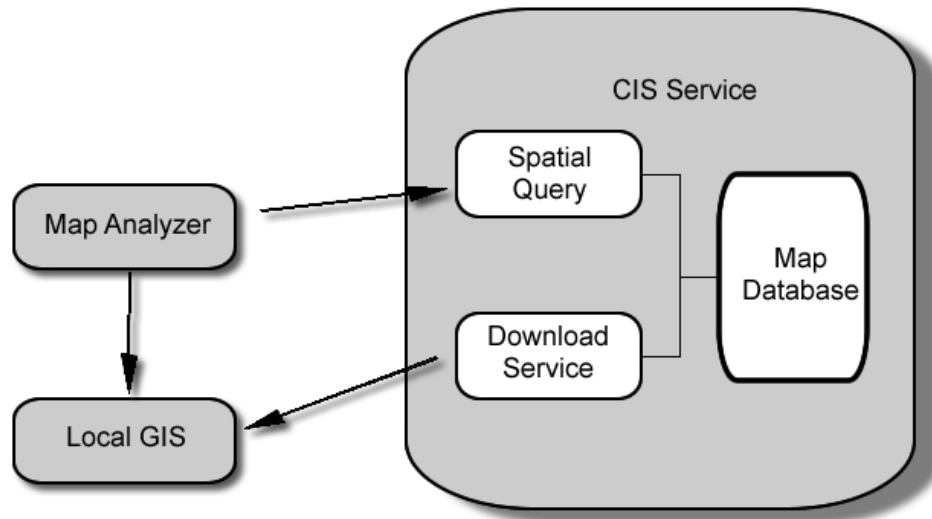
A map user is a user who may need to explore some historic ice coverage or use some simple analysis function to obtain customized results. Map users may query maps to view or get information from them. They may also want to query some past analysis results and compare them. However, as map user may not necessarily have advanced IT skills, predefined functions of analysis (simple analysis functions) are sufficient for them. Figure 3.11 shows the possible activities of such users. Different with general information users, map users need some query support and analysis services for their tasks. Commercial shipping companies, off-shore oil and gas companies, fishing fleets, the mining industry, and tourism and recreation industries can be categorized into this group. They all need some historic data to support their plans, but common analysis suffices. Thus, in addition to exploring data, map user also needs some simple analysis functions.



**Figure 3.11 Possible actions by CIS map user**

### **3. Map analyzer**

Map analyzers are experts who may need to download data and analyze locally or search for data in which they are interested. As Figure 3.12 shows, a map analyzer usually has high level IT skills. He or she prefers to download and customize the analysis and then run them locally. Thus, map analyzers always require convenient query and downloading systems. Typically, construction industry, marine insurance companies, environmental consultants and, of course, research scientists are part of this group. They usually analyze historic ice data then come to a conclusion or build a forecast model. So besides the exploring function, data extracting function is more essential for this group of user.



**Figure 3.12 Possible actions by CIS map analyzer**

### 3.4.2 User Requirement Analysis

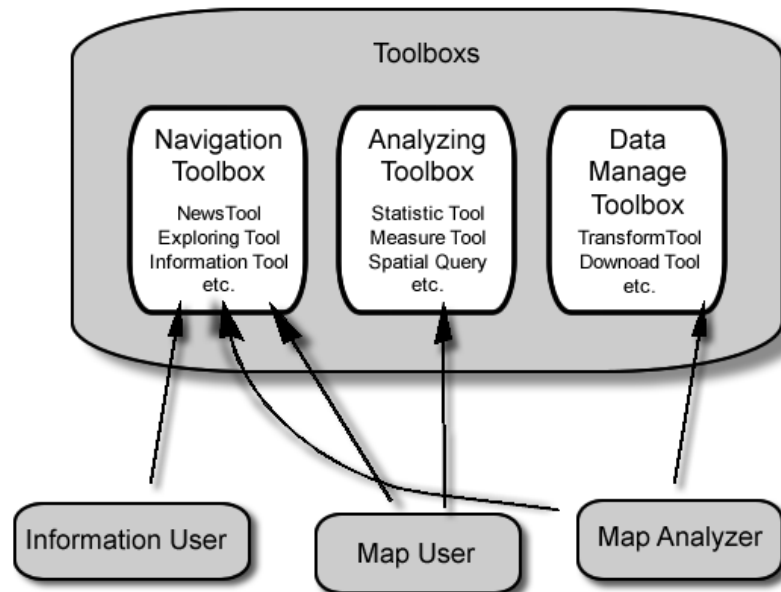
In summary, the three group of user mainly needs: data exploring, simple data analysis and data extracting functions in a sea ice information system. In addition, based on the review in Section 2.1.3, the following requirements should be essential for a SIIS user:

- **Coverage maps exploring:** All types of users need to explore the ice coverage.
- **Data analysis:** Thematic mapping, data animation, data statistics, etc. are convenience functions that many users want.
- **Data querying and extracting:** most expert users need this kind of function.

### 3.4.3 UI Organization

For better HCI effect, the concepts of tool bar and tool box and their structures are introduced to include more GIS functions for ice mapping. Although it is a common operation mode in desktop software, toolbox is hard to realize using traditional HTML language. So here we need a kind of RIA platform to contain and support a tool box system. As mentioned in Section 2.3,

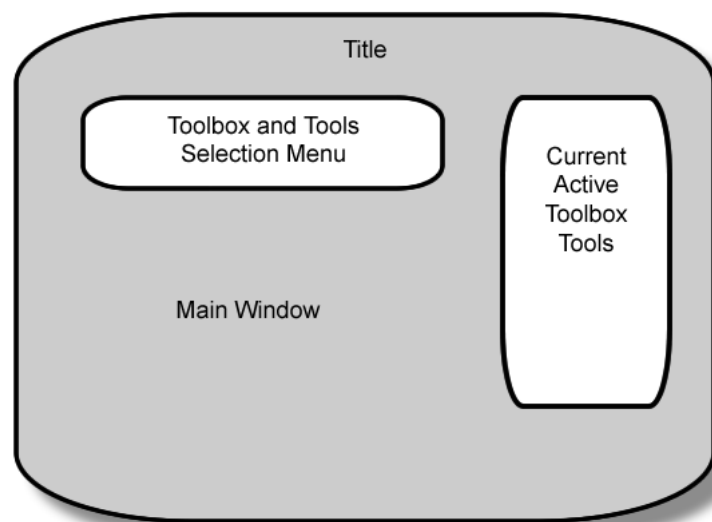
based on an RIA platform we can construct a website with interfaces similar to those of desktop software embedded into the web browser. As such, building a real online GIS with the similar interfaces of a desktop GIS is possible. Certainly, for a public service system, professional GIS software is not acceptable because the system has to fit most users' needs. A flexible toolbox is a good choice to classify different tools into different sections to fit different users' needs. Figure 3.13 shows the general classification of the toolboxes and the relationship between different users and different toolboxes.



**Figure 3.13 User-centered UI design**

As shown in Figure 3.13, tools are classified into three different toolboxes. The navigation toolbox includes all the map exploring tools and some shortcut tools, like news tool, frequent use information tool, etc. These tools are useful for most users. The news tool provides news updates and may include some shortcut links to the latest ice condition maps. Exploring tools provide basic map exploring functions like zoom, full extent, change background, etc. The

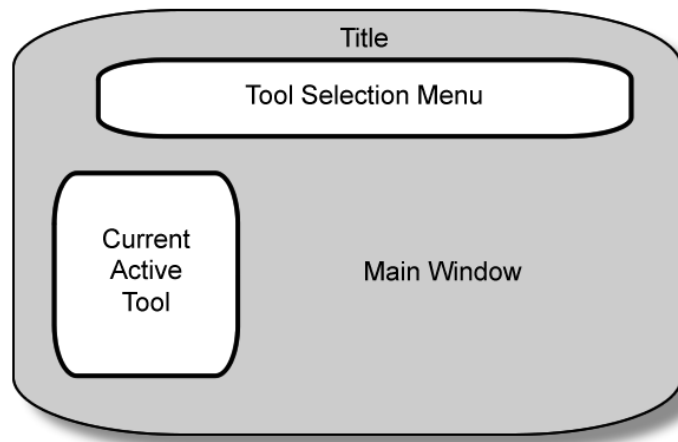
information tool may include some frequently used map links, statistical results or other data information. The analyzing toolbox mainly includes statistical tools and some GIS analysis tools for map users and map analyzers. Map users may frequently use these analysis tools. The data manage toolbox mainly includes the tools for managing data, such as data packing, downloading and transforming. The downloading tool may usually be used by map analyzers. All these toolboxes and classifications may slightly change with different system environments and customer requirements. Nevertheless, the important point is that the toolbox style control panel is the best fit for current systems and requirements.



**Figure 3.14 Toolbox-style user interface structure**

To illustrate, Figure 3.14 shows a possible interface structure of a toolbox-style control panel. When a user selects a group of tools, they will show up on the right side. Or a user can also add specific tools to the toolbox on the right side.





**Figure 3.15 Menu-style user interface structure**

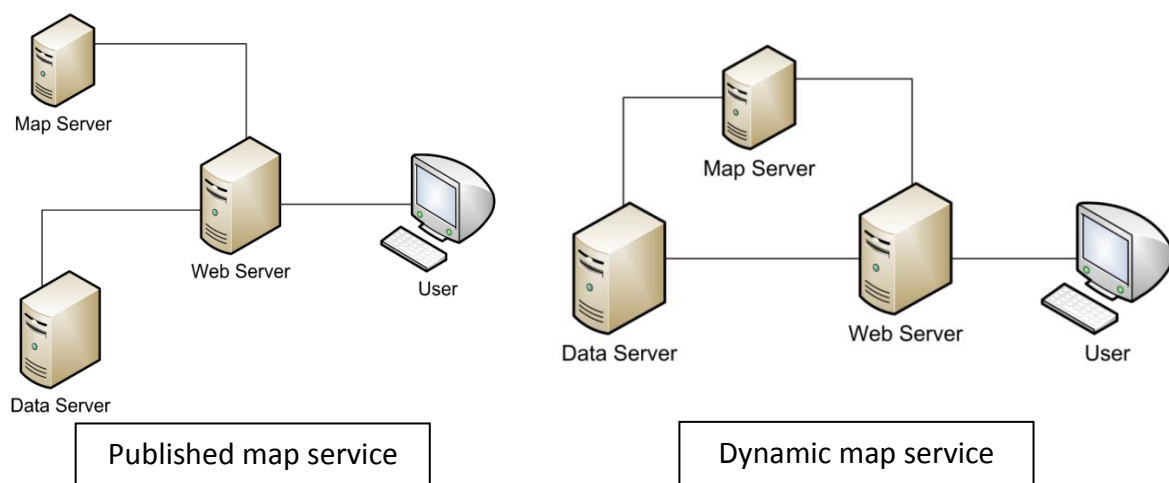
In comparison, Figure 3.15 shows the traditional menu style interface widely used in current desktop software. The biggest difference between these two interfaces is whether the user can see many useable tools at the same time. A traditional menu-style control panel can only allow the tools to be used one by one. However, a toolbox-style interface lets a user have a group of tools handy to use. From this information, the toolbox-style interface has the following benefits:

- ***Every user can find the tool they want:*** Grouped tools cover all user needs. Any user can find a suitable tool to use. The system can be designed in a more user-friendly style and give more guiding information to an information user while at the same time not losing any advanced functions for more advanced users.
- ***Easy to be familiar with:*** Desktop software-style operations are familiar to people. Information users can easily find what they want when it is similar to their previous experience in using, for instance, Google Maps. At the same time, map users or map analyzers can also find the tools they are familiar with when using ArcGIS.

### 3.4.4 Considerations of System Structure

As introduced in Section 2.3, a Web GIS system is a kind of client-server-based and integrated system that combines map service and other services to realize GIS functions. This is a good option for publishing the CIS vector data online and making it more useful. Besides the functions described above, the selection of a system structure should be considered.

A typical web GIS system includes two kinds of services: map service and data service. In theory, with map and data services, a web GIS system can do anything that a desktop GIS can. Therefore, to build a web GIS system with sea ice data, both map services and data services should be considered. These two types of services can be built up either separately or integrated. As shown on the left in Figure 3.16, a separate system has a map server which supports map exploring and a data server which supports map data management and even some dynamic data proceedings. The map server and data server are separated and work individually. The other possible structure is shown on the right of Figure 3.16. In this structure, the map server dynamically obtains map data and uses it as map service when requested.



**Figure 3.16 Two possible web GIS system structures**

The two structures both have pros and cons. The first structure has high loading speed but requires double the storage space to save map data. This is because the map data should be published as a map service onto the map server besides being stored in the data server. Further, the data in this structure is hard to change or update because it needs manual synchronization. In contrast, the integrated structure using dynamic map service only needs half of the storage space compared to the first one and is easy to update data. However, it has long loading time when user explores maps. Considering that the CIS is a public service site, the loading time influence user experience a lot, and so the first structure was selected to build the system.

### **3.4.5 Considerations of System Implementation**

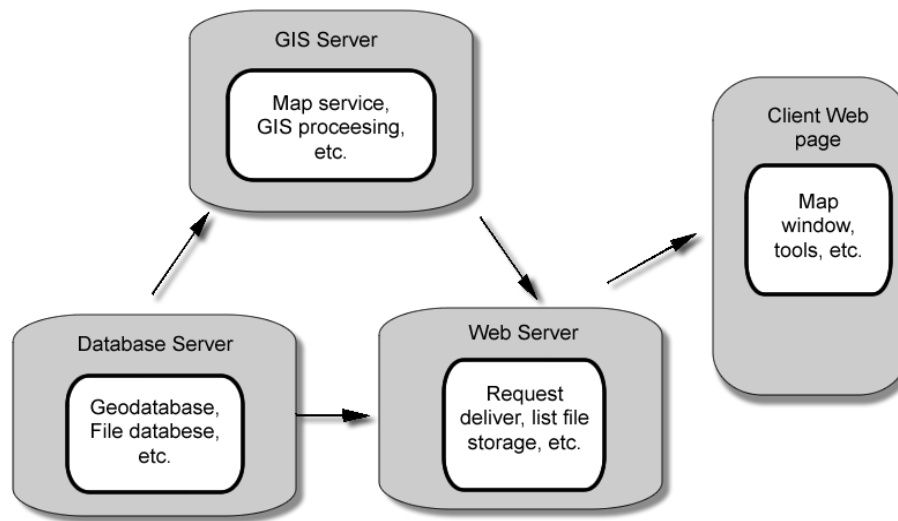
Usually, users need to view map displays of ice coverage or conditions at different times, and sometimes want to switch between different displays to compare their differences. Thus, an ice coverage viewing list which can be dynamically managed is required for fast viewing. The map views of ice coverage or conditions may be partly downloaded into the client buffer so as to enhance the switch and viewing speed. At the same time, users may also want to create a data downloading list by adding large amounts of data into it. If these data are all buffered to the client, system performance would definitely decline. On the other hand, there may be some coverage maps that users only want to take a look and do not want to download. It is impossible to fulfill all these requirements using a single ice coverage list. For this reason, double-list design comes into play.

This design includes two lists, called the “view list” and the “download list”. The amount of ice coverage maps in the view list is limited to a small amount to ensure system viewing

performance. While there can only be a few maps in the view list, it offers high viewing performance. The maps in the download list are only a name in a table which is not buffered in the client to ensure optimal system performance should a large amount of data be added to the list. This list may be long and may also include some maps in the view list, but the maps in it cannot be viewed directly. Therefore, the data in the download list could be packed into a ZIP file and downloaded just by a single mouse click. This double-list design solves the problem of conflicts between fast viewing and large data downloading. It is a successful implementation design integrating all the requisite functions into one system.

### **3.5 Summary**

This chapter described the design and some design considerations of the proposed WebSIS for the CIS Ice Service Archive. The design and related considerations focus on system requirements, functional designs, and technology considerations for system implementation. Figure 3.17 provides an overview of the system. At a higher level, the system is mainly formed by four parts: the database, the GIS server, the web server and the client-side web interface. When a user uses a tool in the web interface to form/send a request, the request is first analyzed by the web server which then decides where to obtain the data or find the service to complete the request.



**Figure 3.17 Overall design of the system**

The arrows in Figure 3.17 are showing the data stream direction when users have different requests. When a user makes a request, the first decision made by the web server is whether it can directly respond to the request. If it does not have the requested data, the next decision should be where to forward this request. If the request is to display a map or do some analysis on maps, the web server forwards the request to the GIS server. If the request data is some text report document or image, the web server forwards the request to the database server to fetch the data. Under other conditions, when the request reaches the GIS server, it should again decide whether it can respond to the request. If there is a map displaying request, it directly responds to the web server, while if there is a GP request, it executes the requested GP script or GP model and sees if it needs the data from the geodatabase or not. Although this structure looks similar to the one on the right in Figure 3.16, this actually is the overall system, and the map service and the GP service are both running on the GIS server. The data stream going into the GIS server from the database is for GP service only, not for map service.

## Chapter 4. SEA ICE INFORMATION SYSTEM PROTOTYPE DEVELOPMENT

A prototype of web-based sea ice information system, named *WebSIIS*, has been developed according to the CIS requirements and designs discussed in Chapter 3. A prototyping approach has been adopted to build the prototype based on well-understood requirements, and gradually improving it as further requirements are identified. The purpose of prototyping is to:

- Facilitate further discussions with the CIS staff;
- Confirm/refine requirements from the CIS's perspective;
- Demonstrate an implementation of web-based access to the CIS Ice Service Archive using a proprietary GIS server platform.

The chapter is organized into three parts. The next section describes the prototype development, followed by a presentation of the prototype implementation. The chapter ends with a summary on some issues and problems encountered.

### 4.1 Prototype Development

A prototype was developed based on the requirements identified through discussions with the CIS. The functions demonstrated in the developed prototype relate to the following capabilities of a WebSIIS:

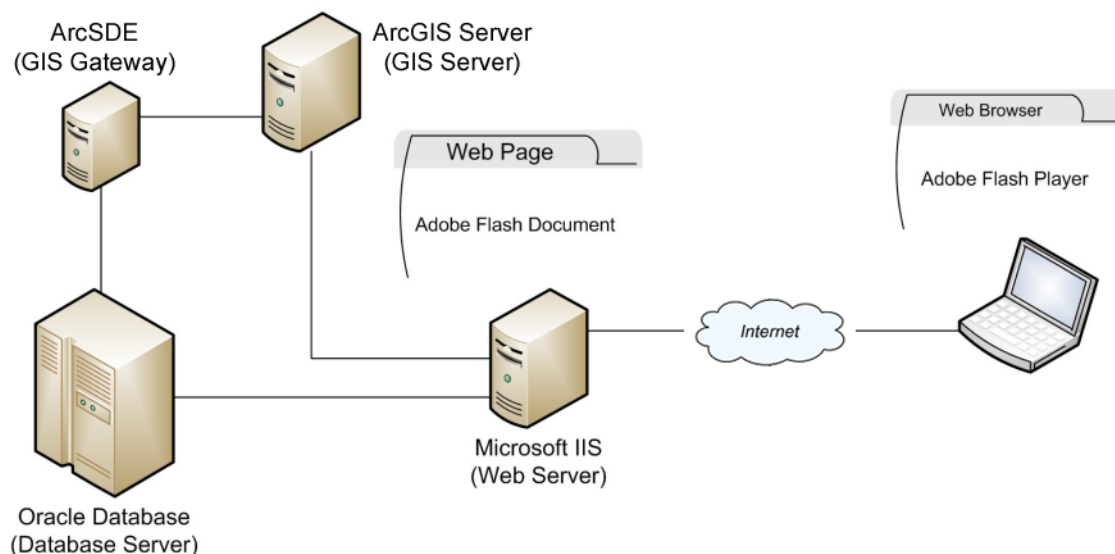
- **Visualization capacities:** simple visualization of historical sea ice data in the form of map-based displays, animation and statistics graphs, with user-defined area of interest (AOI) and periods of time (POT); and

- **Data access, processing and management capabilities:** centralized ice data archive, accessed by users online through user-defined AOI or POT and data packing for downloading.

The description of a prototype system begins with an introduction of the software and hardware components, followed by some configuration and programming issues before introducing the implementation of the prototype functions.

#### 4.1.1 Prototype Components

As discussed in Section 3.5, the client-server-based system architecture calls for a web server, a GIS server, a database server, some server-side service components, a web browser, and the required hardware and networking components. Figure 4.1 illustrates the overall system configuration of the WebSIIS.



**Figure 4.1 Prototype WebSIIS System components**

The system is centered with an ArcGIS server. It uses Adobe Flex to build a web RIA page in Adobe flash document format (\*.swf) storing on the web server. The flash document is programmed with the assistance of an ArcGIS Server Flex API to consume GIS services provided by the ArcGIS Server. When a user requests some kind of common data, such as ice news or forecasting reports, a flash client will directly connect to the database to retrieve the data, just like a common website. But if a user queries for spatial data or performs some other GIS operations, the system will connect to the ArcGIS server and act like GIS software. The following provides detailed descriptions about the components forming the system.

#### ***4.1.1.1 Software components***

Adobe Flash Player is a web browser plug-in that supports most 32-bit browsers in the market (for a complete list of supported web browsers, please refer to (last visited on August 9<sup>th</sup>, 2010) <http://www.adobe.com/uk/products/flashplayer/systemreqs/>). Version 10 of the flash player does not support Microsoft Windows 98, Microsoft Windows ME, Mac OS X v10.1–10.3, and Red Hat Enterprise Linux 3 and 4 operating systems. However, Flash player 9 still supports these systems and can perfectly play back the flash documents created by Flex SDK 3.5 or earlier (Adobe, 2010a). This player is required in client browsers to play back the flash document from the web server. The biggest benefit of using Adobe Flash to build the website is that the programmer does not need to worry about the compatibility between browsers and systems. In addition, as an RIA platform, Flash documents can easily carry complex programs with multimedia, vector and even 3D graphics (Adobe, 2010b).



ESRI ArcGIS Server is an all-in-one GIS server, as introduced in Section 2.3. Version 9.31 is selected to build the system. It provides not only basic functions of a typical Internet map server but also many complex GIS operations using its geoprocessing services. Geoprocessing services can be either customized from the existing services or developed from scratch.

There are many proprietary Internet map server software systems currently on the market, such as Bentley Systems' GIS products, ESRI's ArcGIS products, Cadcorp SIS product suite, MapInfo Professional, GeoMedia WebMap, GlobalMapper, and Manifold System. But not all of these map servers can provide a powerful development environment to build the system. These systems can be divided into three types. One is the CAD-based system that has an extension to supply web map services like Bentley Systems' GIS products and Cadcorp SIS product suite. These map servers do not support many GIS functions such as spatial query, so they are not considered here. The second type is simply a web map server system extended from a desktop GIS software including Mapinfo, Intergraph (GeoMedia) and Manifold. Such systems can only view the map or do some simple information viewing tasks like query and measure. They do not allow any complex process to be made to the map by the web client. The third type is a specialized and often functional-rich and powerful system which has been designed and developed as an Internet map server system from the beginning, such as ESRI ArcGIS. This kind of system provides not only all basic functions of a map server, but also many complex GIS operations, e.g., using the ArcGIS server geoprocessing service. ESRI ArcGIS was selected as the one most suitable for building the prototype system.

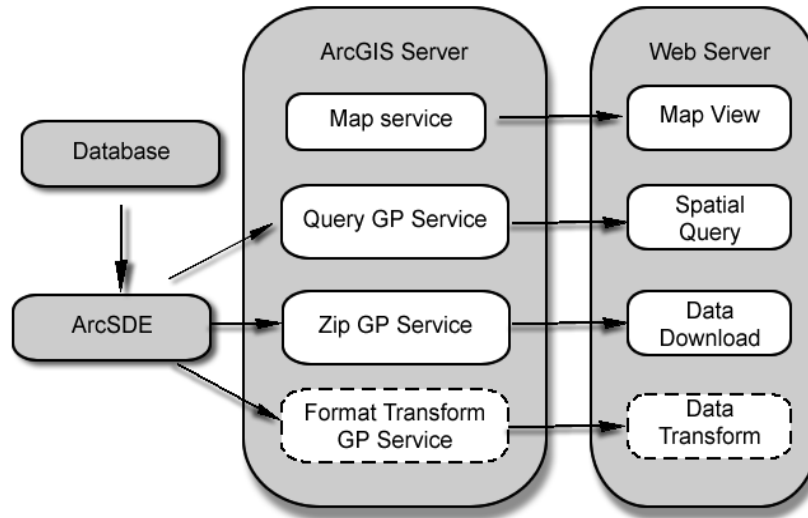
ArcSDE Technology is a gateway between GIS clients and the RDBMS. It enables easy storage, access, and management of spatial data within a database like DB2, Informix, Oracle, PostgreSQL, SQL Server and SQL Server Express. ArcSDE technology has become a part of the ArcGIS server product since version 9.3 (ESRI, 2009b). It is designed to provide spatial data services and support many databases and data formats. Compared to other spatial database products like the Oracle Spatial, ArcSDE is very compatible with an ArcGIS server and has a low learning curve in building up a system. These two features make it an ideal part of spatial data management components.

Oracle database 11g R1 is a famous relational database management system. An ArcGIS server has been selected as the map server and the Oracle is one of the database software it supports through ArcSDE. Another reason the Oracle was selected is due to its license availability at Ryerson. Ryerson does not have other proprietary database software licenses available that ArcSDE supports at the time of building the prototype program. The Oracle database was mainly used to manage the data. Other databases may also be selected and used since ArcSDE does most of the transforming work itself.

Microsoft Internet Information Server (IIS), as part of the Windows Server operating system 2003 (32 bits), was selected as the web server of the WebSIIS. Version 6.0 is a built-in component of the Windows server 2003 that can be directly used without any configuration after the operating system is installed. IIS 6.0 is selected because it is simple and has enough functions to be a web server when handling flash documents. Of course, other web server

programs like Apache Tomcat can also be used with no problems, but they need to be installed first and some require complex settings.

Server-side scripts and components include services, scripts on the ArcGIS server and flash programs on the web server. As shown in Figure 4.2, three Geoprocessing (GP) services and some map services are planned to set up the ArcGIS server. The “Format Transform” service is not a function mentioned in the CIS list of requirements, but it is of great value for the system users to be able to download ice data with more options of data formats. However, due to time limitations and difficulty in building the service model, it has not been completed yet. This has no impact to the main objectives of this thesis. Each service could be published using either ArcCatalog or through the ArcGIS server management page. The GP services are built using ArcGIS Model Builder, and some scripts in the models were written using python scripting language. To consume the services using a Flash document, a substantial amount of codes were written by Flex. Although the main Flex program was developed based on the basic ESRI Sample Flex Viewer, over 1000 lines of codes were written to build up the functions required.



**Figure 4.2 GIS services in prototype system**

#### ***4.1.1.2 Other Considerations in Deciding Software Components***

There were two issues when deciding the software components mentioned above. One is the selection between proprietary and open source software and the other is the selection between RIA platforms. Both considerations are based on the perspective of the requirements of the CIS.

##### **1. Proprietary vs. Open Source**

There is always proprietary and open source software in every market area. Which type should be used is the first thing that has to be considered when choosing suitable software after the system structure is decided. As their names suggest, proprietary software always requires the user to pay the fees, but often has good technical support as well as security updates. In contrast, most open source software is free to use and modify but usually has limited documentation to rely on. In addition, there is no guarantee that the original authors also keep on upgrading, fixing bugs and providing continuous support.

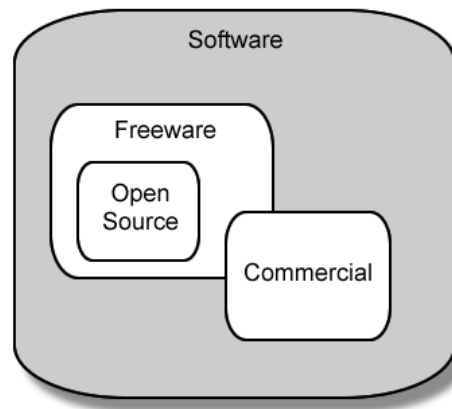
For the sake of prototype development, the developing speed and program functions are the most important points to consider. As shown in Table 4.1, proprietary software has fast learning speeds and good functional support. Although it may require license fees, well-tested functions and better documentation in tangent with its reliability make it worth considering in many applications. Thus, it was decided that the prototype program be built mainly upon a sort of proprietary software.

**Table 4.1 Differences between proprietary and FOS software**

<b>Item</b>	<b>Proprietary software</b>	<b>Free and Open source Software</b>
<b>License fee</b>	Usually need fee for full function	Free
<b>Use area and modification</b>	Follow the warranty and license agreement	Unrestricted use
<b>Documentation</b>	Perfect	Depends on the author
<b>Functions</b>	Usually rich and fully functional	Sometimes not fully functional and may be simple or focus on single function
<b>Update and debug</b>	Frequently update and usually have few bugs	Depends on the author
<b>Security</b>	Usually high	Usually low
<b>Learning curve</b>	Fast and easy but may need additional training cost	Slow and difficult

With respect to the differences between the "free" software and "open source" software, the software objectives of "free" software and "open source" software are in general different and they belong to different groups of software. As illustrated in Figure 4.3, all the open source software is free but not all free software is open source. And even some simple, basic beta version of proprietary software is free. This kind of software usually has good documentation and some limited support that is also handy to use. In addition, well-documented open source

software with good support can also be used to overcome some limitations that the proprietary software may have encountered.



**Figure 4.3 Relationship between different kinds of software**

In conclusion, the development of the prototype system mainly used proprietary software (ESRI ArcGIS server) with a mix of some well-documented free (Adobe Flex) or open source (Sample Flex Viewer) software to build the system.

## **2. RIA platform selection**

There are two major RIA platforms currently on the market: Flex and Silverlight. Flex is provided by Adobe and is built upon the Flash virtual machine. Silverlight is provided by Microsoft and is built upon .Net Framework. There are many differences between these three platforms. These differences are summarized in Table 4.2.

**Table 4.2 RIA platforms feature comparison matrix (re-written from (Roossien, 2009))**

	<b>Flex/Flash</b>	<b>Silverlight</b>
<b>Animation</b>	Richly supported	Limited support and difficult to program than Flex
<b>Bitmap Manipulation</b>	Supported natively	Supported
<b>HTML Rendering</b>	Limited support	Limited support
<b>Video and Audio Streaming</b>	Supported	Supported
<b>Runtime</b>	A single browser plug-in provides a single cross browser cross platform runtime	A single browser plug-in provides a single cross browser cross platform runtime
<b>Strongly Typed / OOP Language</b>	E4X Compliant	Supported through .NET framework
<b>Vector Graphics</b>	Supported natively	Supported
<b>Security</b>	Utilize the browser's security sandbox to ensure the security of the desktop.	Utilize the browser's security sandbox to ensure the security of the desktop.
<b>Server Communication</b>	Allows communicate using web services, REST, simple XML and Binary sockets	The Silverlight client uses a custom AMF interface based on the FluorineFX library.
<b>Search Engine Optimization (SEO)</b>	Limited support. SEO is achieved through meta tagging and alternate content publishing	Limited support. SEO is achieved through meta tagging and alternate content publishing
<b>Accessibility</b>	Fully support 508 compliant	Limited support through third party plug-in
<b>Automated Testing</b>	Flex components include an automated testing framework that tie into tools like QTP or homegrown solutions.	Limited support
<b>Openness</b>	Flex and Action Script are open source the Flash Virtual Machine (FVM) is not.	Fully open source since .Net Framework 3.5
<b>Extensibility</b>	The Flex components can be easily extended and new ones written. The Eclipse based Flex Builder is also significantly extensible.	Fully extendable through .Net Framework

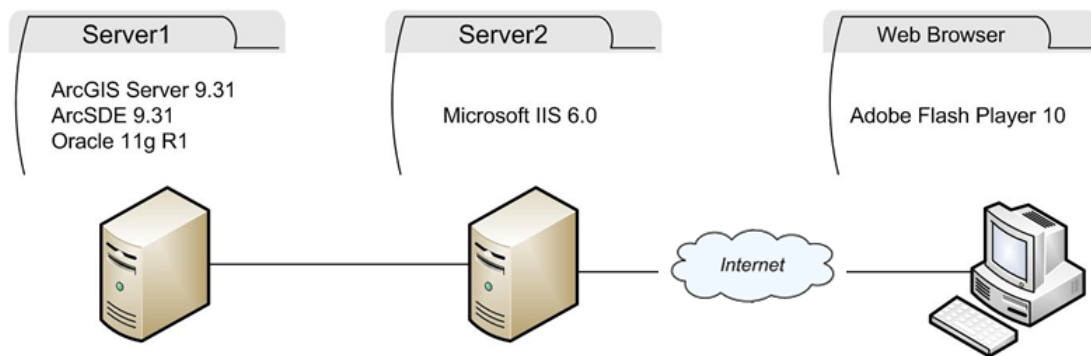
As shown in Table 4.2, Silverlight's architecture is less compact and (ironically) more flexible than Flex (Roossien, 2009). As a result, more architectural pieces need to be assembled to accomplish the same tasks in Silverlight. Silverlight's "UserControl" contains fewer important methods, events and properties than Flex's "UIComponent". Though it may change, at this time, Silverlight is not a platform for rapid application development. It is missing too many basic features and leaves too many basic user interface problems unsolved. Flex has lower startup costs than Silverlight in terms of infrastructure. This is mainly due to Flex's compact architecture with extensive features built into controls. Many common user interface problems are solved for developers and this makes developing Flex applications intuitive and fast. This is the main reason that Flex was selected because it can be quickly built up and also demonstrates good performance.

In conclusion, Oracle database, Adobe Flex RIA platform and ESRI production serials were finally selected to build the prototype program.

#### ***4.1.1.3 Hardware and Networking Components***

A number of hardware components were configured and networked to support the development and testing of the prototype. The hardware components include two computer servers and a personal computer (Figure 4.4). The servers are linked with each other through local Ethernet and linked to the Internet through the Ryerson campus network while the client computer is linked directly to the Internet through a Rogers cable modem.





**Figure 4.4 Prototype system hardware components**

Server computers were used to build the system. One is a high performance server, with an Oracle database, ArcSDE and an ArcGIS server installed on it. Because this is only a prototype program and does not need very high performance, installing all these software on one computer makes it easy to configure and manage. The other server computer is configured as a web server only. Both computers are running Microsoft Windows Server 2003 Standard Operating System. Table 4.3 specifies these hardware components.

**Table 4.3 Hardware specification of the prototype system**

Specifications	Server1	Server2	Client PC
<b>Operating system</b>	Win 2003 std SP2	Win 2003 std SP2	Win XP SP3
<b>Model</b>	DELL PowerEdge2600	DELL Precision360	DELL Dimension4600
<b>CPU</b>	Dual Intel Xeon 3.2GHz	Intel Pentium4 3.4GHz	Intel Pentium 4 2.8GHz
<b>RAM</b>	4096MB	2048MB	1024MB
<b>Storage</b>	300GB	160GB	80GB
<b>Network</b>	Ryerson campus network to Internet	Ryerson campus network to Internet	Ryerson campus network to Internet

These components are selected mainly based on availability, capability and learning curve in using them. The client computer actually does not need such high performance. Any client PC

that fits the hardware requirements for Adobe Flash Player 9 is acceptable. Because the Flash document is compiled by Flex SDK 3.5, any Flash player of version 9 or above should play back the client web page without any problem. The following section briefly describes the way to configure these components and make them work together.

#### **4.1.2 System Configuration and Programming**

The prototype development involved setting up the selected platform and programming server-side components and services using a number of development tools and programming languages. When selecting these tools and languages, a number of factors were considered, including availability, capability with the selected components, and the level of learning curve needed in order to use them. The selected development tools and programming languages include ESRI Sample Flex Viewer, Adobe Flex Builder 3, ArcGIS Desktop 9.3.1 and Python script language. Appendix E presents examples of some code segments. The following sections briefly describe these tools, languages and the work related to system configurations.

##### **4.1.2.1 Overall System Configuration**

All the software is installed mostly using default settings. However, the Oracle database should be installed before ArcSDE as some automatic configurations in the Oracle database are required in order to get them to participate with each other.

As mentioned in Section 2.3, an ArcGIS Server is an all-in-one style GIS server. It has both data and map functions. A Geoprocessing (GP) service provides the data service and a variety of GP services are built to complete different tasks (see Figure 4.2).

A map service is a separate service which only supplies sea ice maps being explored. There is a record file at the web server to save all the maps' service locations. The query service is mostly used by the Ice Graph tool to generate the statistics and may also be used by the identify tool. This service works along with ArcSDE spatial interface to support spatial queries. Zipping and format transforming services are parts of data extracting services that can fetch data from the database and get them ready to be downloaded by the user. All the processing result files are stored at the GIS server and then return a location address to the flash client through the web server when ready.

Briefly, to set up an ArcGIS server based system, one needs first to install a web server such as IIS or Tomcat Web server. Then, the ArcGIS server is installed as well as the setup of the map services and GP services for the service consuming program. Finally, a website can be built with this service consuming program page and published onto the web server.

#### ***4.1.2.2 Setting up Servers***

There is nothing special in installing the Oracle database or ArcSDE spatial database interfaces. Everything is configured by its default settings, except the names of databases. One needs to build separate database instances for text or image data if the system is built to include all the types of CIS archives. After the ArcSDE is set up, the ice coverage map can be easily imported to the database through a direct link using ArcCatalog, which is a component of ArcGIS Desktop.

The next step is to set up the ArcGIS server. Here, the ArcGIS server .Net version is chosen to build the system. As mentioned previously in Figure 4.2, some services should be set up after the installation of the ArcGIS server. Maps can be easily published to the server directly using

ArcCatalog. At the same time, other services should be built using ArcGIS Model Builder and then published to the server as GP services. When the services are set up, a web program is required to consume these services and finally realize the functions designed.

#### 4.1.2.3 ESRI Sample Flex Viewer

ESRI Sample Flex viewer is an ArcGIS server Flex API-based sample program provided by ESRI Situational Awareness. It is an open-source program that mostly fits the design of SIIS in Chapter 3. The web client part of the prototype system is built by modifying this program. Its main structure is shown in Figure 4.5.

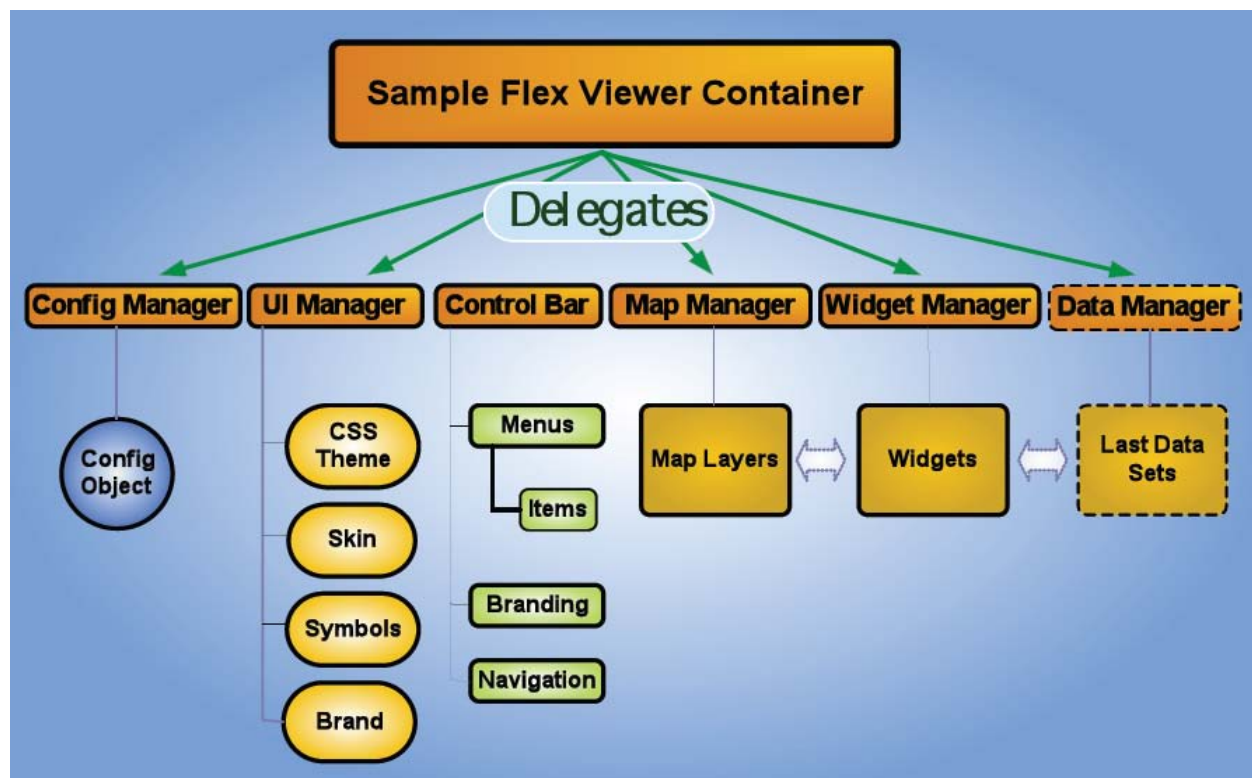


Figure 4.5 Components of Sample Flex Viewer (Zhang, 2009)

The Sample Flex Viewer is based on an Adobe RIA platform. It is written by Flex, which is a combination of the Action Script and MXML language standard setup by Adobe. As it is shown

in Figure 4.5, the program has a configuration manager to apply some internal settings, a UI manager to manage the program appearance, a control bar showing as a menu, a map manager to manage maps, and a widget manager to manage tools. The data manager is actually not included yet, but may be a future expansion of this program. The Sample Flex Viewer interface is shown in Figure 4.6. It has a flexible toolbox-style interface and some basic GIS functions such as map viewing and data query. Each tool is called a widget in Sample Flex Viewer. There is a widget container on the right side that the user can put any widget into for quick retrieval. This structure perfectly fits the design principles of the user interface.



**Figure 4.6 Sample Flex Viewer interface**

Although it looks powerful, there are many shortages in the version 1.3 of Flex Sample Viewer. First, it cannot add or remove maps actively. Maps loaded in the viewer are predefined in the configuration file loaded when the program is started up. Users can only change the visibility of maps. Second, there are many small bugs or functions which do not fit to the system design and need to be changed. For example, the widgets cannot be moved anywhere freely and the

program does not have a way to store the download list. Finally, the main function widget has to be built (like thematic map tool) or completely rebuilt (like view list manager) from scratch. These widgets including Map management widgets, Download widget, Ice Graph widget and other associated tools.

Despite of these drawbacks, this program provides a good base to build upon. Thus, the prototype system is modified after this program, with some widgets and other functions added to fit the requirements.

#### ***4.1.2.4 Languages used for Server-side***

Adobe Flex Builder and Flex language: Adobe Flex is “a free, open-source framework for building highly interactive, expressive web applications that deploy consistently on all major browsers, desktops, and operating systems” (Adobe, 2010c). Adobe Flex builder is an Eclipse-based (Eclipse is an open source, uniform language development software) development tool that can do coding, debugging and visual design of the user interface layout, appearance, and behavior of Flex (Adobe, 2010c). Flex builder version 3 is used with the Flex 3.5 compiler to build the Flash part of the system. At the time of this thesis writing, version 4 of Flex and Flex builder (at version 4, the Flex builder is renamed as Flash builder 4) has been announced. However the Flash document compiled under the standard of Flex 4 has to use Flash Player 10 to get maximum support and capability. As mentioned previously, Flash Player 10 does not support some older operating systems and browsers. This narrows the number of available computers to run the system. That is why the prototype system still uses Flex 3 standard to build.

ArcGIS Model Builder is a tool in ArcGIS Desktop that can quickly build up a customized GIS tool using existing tools or scripts. The tool built by this builder can be used directly in the ArcGIS server to build up a GP service. All of the GP services in the prototype system were built by GIS tools formed using ArcGIS Model Builder.

Python script language is “a free, cross-platform, open-source programming language that is both powerful and easy to learn... and is widely used and supported” (ESRI, 2010). In ArcGIS Model Builder, GIS scripts written by Python can be directly used. It is an easy-to-learn and easy-to-use scripting language like JavaScript. Some GP services use some scripts written by Python.

## **4.2 Prototype Implementation**

The development of prototype implementation functions are focused on the following functions:

- Mapping Sea Ice Coverage;
- Visualizing and Analyzing Ice Data; and
- Accessing and Downloading Data.

This part begins with an overview of the prototype program interface and is then followed by demonstrations of each functional part to draw a complete picture of the current functions in the system.

### 4.2.1 Overview

As described in Section 3.4, a flexible toolbox-based UI is designed in the prototype system. The grouped toolboxes are at the top left of the UI in the prototype system. The activated tools are docked in the tool container on the right side, and any activated tool can simply be dragged out by moving the cursor to the title of the tool and clicking the left mouse button. The prototype system UI does not only have “Ice Info”, “Analyze” and “Ice Data” toolboxes as designed in Section 3.4 but also has some associate menus like “Help” to bring better user experience. Figure 4.7 shows the default UI when the Flash document is loaded.



**Figure 4.7 The main interface of the prototype system**

The whole UI looks like a map explorer and in fact it supports many map exploring operations like mouse drag panning and mouse wheel zooming by consuming the service of an ArcGIS server. The difference between this explorer and other web maps like Google Maps is that it has many tools in the menu and they all can be activated to the right side of the interface and



even can be dragged anywhere freely inside the program interface. Next, the main functions of this system are introduced one-by-one.

#### 4.2.2 Mapping Sea Ice Coverage

The first function to be described here is the mapping function. As mentioned in Section 3.4, the system is designed to be a combination of map explorer and several GIS-based tools. The navigation menu provides all the basic mapping functions like zooming and panning. In addition, users can easily zoom in or out and panning the map. As shown in Figure 4.8, if users choose a navigation tool such as pan map, the status of the current tool will appear after the tip: “current action” above the “Help” menu button.



**Figure 4.8 Prototype system mapping functions**

In addition, an overview map tool and a coverage maps tool are also provided as associate tools. The overview tool provides a contextual view of the map content currently in the display, while the coverage maps tool allows users to manage the view of ice coverage maps in display, such as visibility and transparency of maps, adding map to, or deleting maps from current view.

### 4.2.3 Visualizing and Analyzing Ice Data

As designed in Section 3.2, the prototype system realized three main visualizing and analyzing functions: animation, statistics and thematic mapping. These functions are implemented in separate tools.

The animation tool can automatically synchronize with the view list (current opened map list) and change the current top showing map in a certain timeframe to make it appear animated. In Figure 4.9, the animation tool is paused at the map showing ice coverage in the week of Jan 29, 2007. The play speed is changing at one map layer per second rate.



**Figure 4.9 Prototype system animation function**

The ice graph tool is a tool which looks very similar to the results of the CIS ice graph. However, even though it looks like it has all the functions of the ice graph, only the “weekly ice coverage graph” of “total concentration” is working. This is because, as this is only a prototype program, one type of graphing is sufficient to prove all of these functions are realizable. To be more

specific, to realize one kind of graph, the database link, the calculation and the graphing program should all be working properly. Thus one type of the graphing is enough to prove the design. Figure 4.10 shows the analysis results of the period from Oct 22 to Dec 25, 2009.



**Figure 4.10 Prototype system Ice graph function**

The Thematic Mapping tool is a tool which can apply a customizable theme to specific ice coverage map in order to analyze ice coverage based its attributes. It is also a prototype which can apply some predefined themes to ice maps. Nevertheless, it is only a problem of time to create a sufficiently complex thematic mapping tool to provide users with flexibility in customizing the theme. As shown in Figure 4.11, users can set the applying target and style of theme to generate a customized thematic map.



Figure 4.11 Prototype system thematic map function

#### 4.2.4 Accessing and Downloading Data

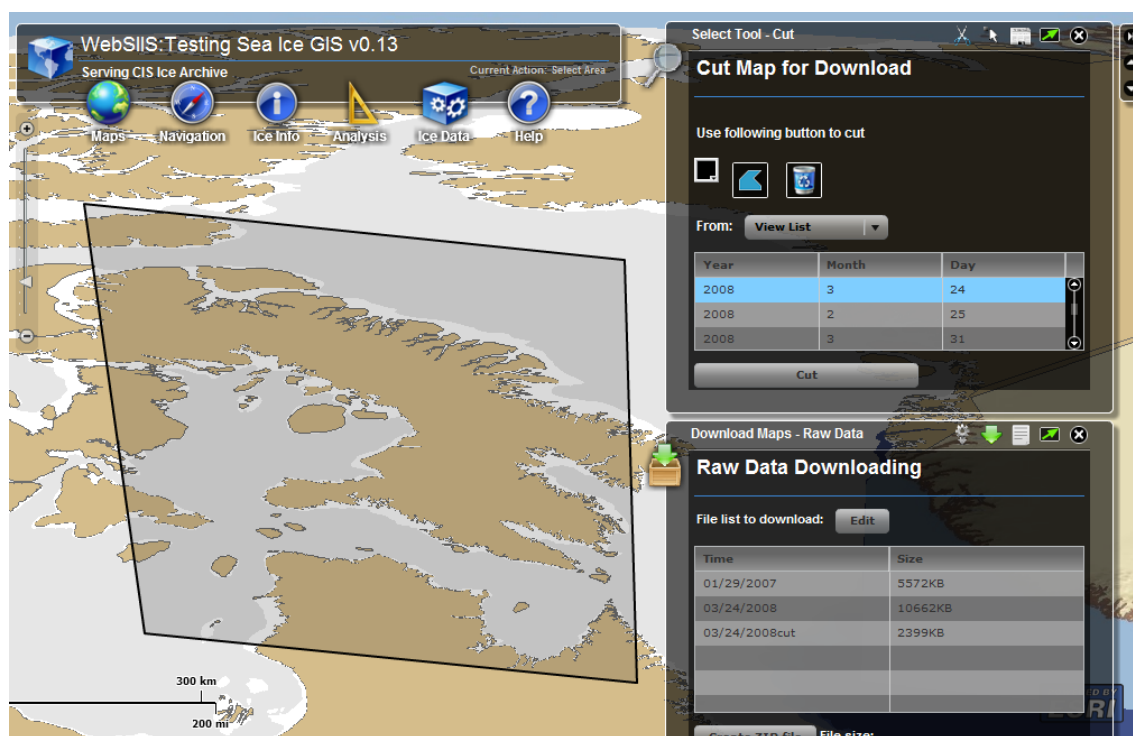
Following the design in Section 3.3, two tools are designed to separately manage access and download functions. One is a select tool that not only can query the database for a certain time frame (TF) but also can cut/extract a customized part of data (by AOI) showing on a current viewing coverage map for downloading. The other is a download management tool with zip file packing function.



Figure 4.12 Prototype system map management tools

Through the select tool, users can query for any timeframe of data from the database and add them to their view list or download list (this design is mentioned in Section 3.4). The coverage

maps tool mentioned above can manage the view list to check the search results. The download list management tool is in the download tool. Figure 4.12 shows a map search result between 2007 and 2009. Two of the resulting maps are added to the download list. Another function of the select tool is AOI cutting. It can generate a segment of map using customized AOI and add it to the download list. Figure 4.13 illustrates the cut and downloads packing function of the two tools. The default downloading format is in geodatabase format (\*.mdb). This is the default format produced by the GP service on ArcGIS server. Further data format transforming is still under development.



**Figure 4.13 Prototype system AOI cut and download tools**

### 4.3 Simple Evaluation

To demonstrate the real improvement of this prototype system in comparing with current CIS web site, some simple evaluation tests were conducted and are described in this section.

There are two functions that current CIS also has: data downloading and packing and ice statistics graphing. To make a simple evaluation test, a computer outside the Ryerson campus network was picked up to download a data package for the same time frame and generate the same statistics graph in order to compare the processing time.

This computer is a desktop PC using Rogers TV cable to connect to the Internet. The network downloading bandwidth is about 600KB/s. The download test was designed to download a sea ice map data package including the weekly regional ice data in January 2007. The zip package is about 9MB in \*.e00 format and 11MB in \*.mdb format. The statistics graphing test was to generate a graph during a period from Oct 12 to Dec 25, 2009 with the settings showing in Figure 4.10. All the processing times were recorded by a stopwatch except for the “Internet latency”, which was calculated by the windows command “ping.exe”. Table 4.4 shows the test results.

**Table 4.4 Simple evaluation of the prototype system**

	System	Total time	Operation time	Download time	Operation number	Internet latency
<b>Download test</b>	WebSIIS	26s	8s	18s	6	15ms
	CIS site	60s	45s	15s	35	35ms
<b>Statistic test</b>	WebSIIS	4s	3s	1s	5	15ms
	CIS site	13s	12s	1s	10	35ms

The operation number represents the total number of operation counts that the tester needed to go through before getting the result. The operation time is the total time cost during the operation. In the download test, current CIS site needed seven operations to generate a zip file for downloading. In addition, the data have to be downloaded region by region. That means user cannot select data from different regions to be packed into on single zip file. To download

all five regions' data, user has to repeat the operations five times. The data in the prototype system has been merged. So it took only six operations to download the data. In statistics graphing test, current CIS site needed ten operations because it needed three steps with some operations in each step to set all the parameters and then generate a graph. Generating a graph using the prototype system only requires five operations at most. That because user only needs to change five kinds of parameters and the graph can be automatically generated after changing. The Internet latency of the CIS site was higher than the WebSIIS because the client computer was closer to the server at Ryerson than the CIS server. But this latency can be ignored because most differences of them in the tests are smaller than 1s.

As it can be seen in the tests, the WebSIIS really improves the data fetching speed in comparing with the current CIS site. The prototype system proved that it is really helpful to build such a system to manage sea ice data.

#### **4.4 Summary**

This chapter describes a prototype system based on the design of Chapter 3. The prototype system realized the main functions described in Chapter 3 and proved that it is a possible solution for the problems given by the CIS. However, there are still some technical problems of the prototype system. For example, with data downloading the zip file sometimes may fail to be created on the server. It is expected that this kind of technical problems will be fixed as the prototype is further refined and tested, and more coding effort is spent on the system development.

## Chapter 5. CONCLUSIONS AND FUTURE WORK

In this research, a web-based sea ice information system (WebSIIS) has been designed and prototyped. The system allows for: (1) easy visualization and visual analysis of the sea ice data from the CIS ice archive; and (2) convenience sea ice data access in terms of data extraction, data packing and data downloading. The following sections summarize the thesis's major contributions, and present some recommendations for future enhancements to the prototype developed so far.

### 5.1 Conclusions

This thesis has first grouped CIS users and analyzed CIS user requirement requirements. These requirements have then been generalized to any sea ice data. In order to design a system with rich visualization, simple analysis and better data extraction/downloading functions, this thesis has tried an improved way of building a web-based sea ice GIS using RIA and HCI principles and guidelines. A prototype system has also been implemented with simple analysis in how GIS may help in improving sea ice information access. To be more specific, the major contributions of this research include:

- Analyzed existing SIIS and discussed user requirements: The thesis analyzed most existing SIIS or similar web sites, described some current conditions regarding their functions, performance, data and services, and also pointed out some noticeable limitations and areas that could be improved.
- Used an improved way to build SIIS: The thesis, following the human-centered information system design principles and the modern SDLC method, proposed a more



user-friendly SIIS design. The human-centered principle was not only applied in the UI design but also in the program structure design so as to enhance user experience.

- *Designed a WebSIIS for CIS with RIA*: The thesis described a possible design of WebSIIS using RIA and Web GIS. RIA brings greater possibilities in building a more complex and user-friendly system compared to traditional Web GIS systems. This design better supplies current functions of the CIS web site and adds some useful functions to enhance user experience.
- *A prototype system based on the design is developed*: A prototype system was developed based on the main functions of the design. It is built on an Adobe Flash platform and ESRI Web GIS solution package using Adobe Flex 3.5 SDK and an ArcGIS server 9.31. The prototype system proved that the designed functions are realizable and have some advantages compared to the current CIS website and other similar services available online.

Overall, the thesis shows that, with the rapid development of the Internet and personal computer, RIA-based web GIS Software has many advantages over other kinds of system in building SIIS. This solution may improve user experience of the CIS website and make more effective use of the CIS historical data.

## **5.2 Future Work**

Although the goal of this thesis has been reached, there are still more works that could be done after this thesis research. The future works relate to two aspects.

From the system development perspective, only an incomplete research prototype has been developed to prove the methodology and the designs. Further refinement and development need to be done before it can become a fully functional WebSIIS that can be used by the CIS to serve historical sea ice information to the public. Some possible works are listed below:

- A restructured geodatabase can be developed to enhance query speed and data management, ideally this should be an spatiotemporal database;
- Build more components to totally realize and replace all the services available on the CIS website;
- Develop a guide and help system for new users;
- Deploy an API system for a customized mapping component design;
- Mashup with other data systems; and
- Add multi-language support (e.g., French).

For further research in web-based sea ice GIS, many topics are not discussed in this thesis research. However, they are worth further studying. Most of these topics are related to the possibility and values of building a WebSIIS completely using open source software, applying collaborative GIS principle into WebSIIS, and adding 3D perspective and 3D GIS data processing support in WebSIIS.

## APPENDIX

### Appendix A: Requirement From the CIS

#### List of Requirements/Deliverables

(Regarding the modeling and development of a spatiotemporal ice database and web-based application)

---

A Prepared for: **Dr. Songnian Li**, Dept. of Civil Engineering, Ryerson University  
(<http://www.ryerson.ca/civil/facstaff/Faculty/songnian.html>)

By: the **Canadian Ice Service** (Trudy Wohlleben, Steve McCourt, Richard Chagnon, Ziqiang Ou)

---

- Become familiar with CIS archived data
- Decide on a data format for the database and develop a process for data conversion
- Using two CIS Areas of Interest (AOI), design and develop a prototype database model
- Design and develop a comprehensive web-based querying tool for this database (similar to Ice Graph, and similar to the NOAA CDC pages:  
<http://www.cdc.noaa.gov/cgi-bin/PublicData/getpage.pl> ,  
<http://www.cdc.noaa.gov/cgi-bin/Composites/printpage.pl> )
  - Design and develop effective visualization techniques for web-based analysis and mapping of the ice data (both static and animated)
  - Similar to the Ice Graph tool, allow for simple analysis, data extraction, etc. allowing for user-defined time-frames, but also allow for user-defined AOIs (chart areas)

## Appendix B: Weekly Regional Ice Chart Attributes

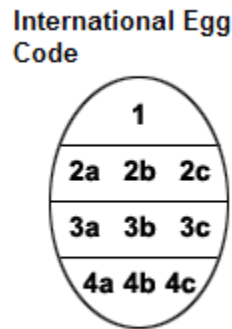
**Table A.1 Attributes of the CIS data table (CIS, 2009)**

Field Name	Fast Ice Polygon	Ice Egg Polygon	Land Polygon	Description
AREA	802496.00	5.82 e^10	10.89 e^10	area of polygon
PERIMETER	3085864.25	2254366.00	8640489.00	perimeter of polygon
N_980126#	3	8	58	internal arc code
N_980126-ID	3	8	58	internal arc code
A_LEGEND	Fast ice	Remote egg	Land	legend name
REGION	CE	CE	CE	name of region:
				BH = Baie d'hudson (Hudson bay)
				AE = Arctique de l'est(East Arctic)
				AO = Arctique de l'ouest(West Arctic)
				CE = Côte est(East Coast)
DATE_CARTE	19981201	19981201	19981201	ice chart date (YYYYMMDD)
SOURCE		EGG		egg or ratio
MOD				data modified ("X" = modified)
EGG_ID	0	0	0	internal egg id
PNT_TYPE	106	118	900	unique label id for polygon features
EGG_NAME		Q		name of egg
EGG_SCALE	0	0	0	internal code
EGG_ATTR		9+_3_4_3_@_7_1_7_5_@_@_3_3_3_@_@_@		string value for egg attributes
USER_ATTR		9+_3_4_3_@_7_1_7_5_@_@_3_3_3_@_@_@		string value for egg attributes
ROTATION	0	0	0	internal code
E_CT		9+		total concentration
E_CA		3		partial concentration of column a
E_CB		4		partial concentration of column b
E_CC		3		partial concentration of column c
E_CD				partial concentration of column d
E_SO		7		stage of development of ice found in traces
E_SA		1		stage of development of column a
E_SB		7		stage of development of column b
E_SC		5		stage of development of column c
E_SD				stage of development of column d
E_SE				stage of development of column e
E_FA		3		floe size of column a
E_FB		3		floe size of column b
E_FC		3		floe size of column c
E_FD				floe size of column d

E_FE					floe size of column e
E_CS					concentration of ice in strips and patches
R_CT					ratio code (prior to 1983)
R_CMY					ratio code (prior to 1983)
R_CSY					ratio code (prior to 1983)
R_CFY					ratio code (prior to 1983)
R_CGW					ratio code (prior to 1983)
R_CG					ratio code (prior to 1983)
R_CN					ratio code (prior to 1983)
R_PMY					ratio code (prior to 1983)
R_PSY					ratio code (prior to 1983)
R_PFY					ratio code (prior to 1983)
R_PGW					ratio code (prior to 1983)
R_PG					ratio code (prior to 1983)
R_PN					ratio code (prior to 1983)
R_CS					ratio code (prior to 1983)
R_SMY					ratio code (prior to 1983)
R_SSY					ratio code (prior to 1983)
R_SFY					ratio code (prior to 1983)
R_SGW					ratio code (prior to 1983)
R_SG					ratio code (prior to 1983)
R_SN					ratio code (prior to 1983)
N_CT	10.0	10.0	0.0		numeric value for total concentration
N_COI	0.0	0.0	0.0		numeric value for concentration of old ice
N_CMY	0.0	0.0	0.0		numeric value for concentration of multiyear ice
N_CSY	0.0	0.0	0.0		numeric value for concentration of second year ice
N_CFY	10.0	7.0	0.0		numeric value for concentration of first year ice
N_CFY_TK	0.0	0.0	0.0		numeric value for concentration of first year thick ice
N_CFY_M	0.0	3.0	0.0		numeric value for concentration of first year medium ice
N_CFY_TN	0.0	4.0	0.0		numeric value for concentration of first year thin ice
N_CYI	0.0	3.0	0.0		numeric value for concentration of young ice
N_CGW	0.0	3.0	0.0		numeric value for concentration of grey white ice
N_CG	0.0	0.0	0.0		numeric value for concentration of grey ice
N_CN	0.0	0.0	0.0		numeric value for concentration of new ice
N_CB	0.0	0.0	0.0		numeric value for concentration of icebergs

## Appendix C: Sea Ice Egging Code

As it listed in Appendix B, “EGG\_ATTR” attribute is representing the values in “Egg Code”. The attributes which are started with “E\_” are also the data in “Egg Code” representing the specific polygon area.



**Figure A.1 Sea ice egg code definitions (CIS, 2009)**

As it is shown in Figure A.1, “Egg Code” has four sections that directly relate to each other. The top section is representing total concentration. That means the concentration of ice coverage in that area. It is expressed in tenths. For example, if the number in the top row is 9, the polygon area represented by this “egg” is covered 90% by ice. The second row in the “egg” is expressing partial concentration, which means the break-down of the total ice coverage. It is expressed in tenths too, but also graded by thickness (thickest-2a, medium-2b, thinnest-2c.) These grades directly relate to the type of ice described in row 3. The third row is recording the ice stage of development. That means it is the type of ice in each of the grades divided in row 2 above. The data in row 3 is determined by the ice age and expressed as a number. Table A.2 shows the meaning of each possible number in this row.

**Table A.2 Egg coding for sea ice stages of development**

Description	Thickness	Code
New ice	<10	1
Ice rind	0 – 10	2
Young ice	10 – 30	3
Grey ice	10 – 15	4
Grey-white ice	15 – 30	5
First year ice	= or > 30	6
Thin first-year ice	30 – 70	7
Thin first-year ice, first stage	30 – 50	8
Thin first-year ice, second stage	50 – 70	9
Medium first-year ice	70 – 120	1.
Thick first-year ice	>120	4.
Old ice	-	7.
Second-year ice	-	8.
Multi-year ice	-	9.
Ice of land origin	-	▲
Unknown, undetermined	-	X

As it shows in Table A.2, not all possible numbers have an approximate thickness range. The thickness of old ices is in a wide range and hard to statistic. The bottom row is recording the floe size of sea ice. The floe size of the ice is determined by the relatively flat piece of ice (e.g., 20 m or more across) for each section and expressed as a code number. Table A.3 is the possible numbers that may appear in the fourth row of the “egg”.

**Table A.3 Egg coding for floe sizes**

Description	Width	Code>
Pancake ice	-	0
Small ice cake; brash ice	< 2m	1
Ice cake	2 - 20m	2
Small floe	20 – 100m	3
Medium floe	100 – 500m	4
Big floe	500 – 2000m	5
Vast floe	2 – 10 km	6
Giant floe	> 10 km	7
Fast Ice	-	8
Icebergs (Ice of land origin)	-	9
Undetermined or unknown	-	X
Ice in strips	-	

## Appendix D: Table of Sea Ice Information Website Addresses

**Table A.4 Web addresses for currently running ice information system**

<b>Name</b>	<b>Address</b>
Canadian Ice Service	<a href="http://ice-glaces.ec.gc.ca./App/WsvPageDsp.cfm?id=1andlang=eng">http://ice-glaces.ec.gc.ca./App/WsvPageDsp.cfm?id=1andlang=eng</a>
Baltic Sea Ice Services	<a href="http://www.bsis-ice.de/index.shtml">http://www.bsis-ice.de/index.shtml</a>
JCOMM Sea Ice Service	<a href="http://www.jcomm-services.org/Sea-Ice-Page.htm">http://www.jcomm-services.org/Sea-Ice-Page.htm</a>
Danish Meteorological Institute	<a href="http://www.dmi.dk/dmi/en/index/gronland/iskort.htm">http://www.dmi.dk/dmi/en/index/gronland/iskort.htm</a>
SOK Ice Service	<a href="http://forsvaret.dk/SOK/eng/National/Ice/Pages/default.aspx">http://forsvaret.dk/SOK/eng/National/Ice/Pages/default.aspx</a>
Baltic Icebreaking Management	<a href="http://portal.fma.fi/sivu/www/baltice">http://portal.fma.fi/sivu/www/baltice</a>
SMHI Polar View	<a href="http://www.smhi.se/polarview/index.html">http://www.smhi.se/polarview/index.html</a>
FMI Ice Conditions	<a href="http://www.fmi.fi/weather/index_9.html">http://www.fmi.fi/weather/index_9.html</a>
Japan Ice Information Center	<a href="http://www1.kaiho.mlit.go.jp/KAN1/drift_ice/eng.html">http://www1.kaiho.mlit.go.jp/KAN1/drift_ice/eng.html</a>
Arctic and Antarctic Research Institute	<a href="http://www.aari.nw.ru/clgmi/sea_charts/sea_charts_en.html">http://www.aari.nw.ru/clgmi/sea_charts/sea_charts_en.html</a>
National Snow and Ice Data Center	<a href="http://nsidc.org/index.html">http://nsidc.org/index.html</a>
International Ice Patrol	<a href="http://www.uscg-iip.org/cms/">http://www.uscg-iip.org/cms/</a>
National Ice Center	<a href="http://www.natice.noaa.gov/">http://www.natice.noaa.gov/</a>
NZ Antarctic GIS	<a href="http://www.anta.canterbury.ac.nz/gis/">http://www.anta.canterbury.ac.nz/gis/</a>
University of the Arctic Atlas	<a href="http://uarctic.org/AtlasMapLayer.aspx?m=643andamid=5992">http://uarctic.org/AtlasMapLayer.aspx?m=643andamid=5992</a>
Arctic Research Mapping Application	<a href="http://www.armac.org/">http://www.armac.org/</a>



## Appendix E: Example Code Segments

### Example of Flex code

This is a code segment in order to add a map to view list.

```
private function addtoViewList(toAdd:Array):void
{
    var i:int;
    var tmp:MapData;
    var labelLast:String;
    var label:String;
    var j:int;

    for (i=0; i < toAdd.length; i++)
    {
        tmp=toAdd[i] as MapData;
        if (tmp.mon > 9)
            label=tmp.year.toString() +

tmp.mon.toString();

        else
            label=tmp.year.toString() + "0" +

tmp.mon.toString();

        if (label == labelLast)
            continue;

        var tKey:Date=new Date(tmp.year, tmp.mon - 1,

tmp.day);

        var vKey:Array=viewList.getKeySet();
        for (j=0; j < vKey.length; j++)
        {

            if (ObjectUtil.dateCompare(vKey[j], tKey)

== 0)

            {
                Alert.show("Map " + label + " has

been added into view list!", "Error");
                break;
            }
        }
        if (j != vKey.length)
            continue;
        var dynlayer:ArcGISDynamicMapServiceLayer=new
ArcGISDynamicMapServiceLayer(tmp.url);
        dynlayer.id=label;
        dynlayer.name=label;
        dynlayer.visible=true;
        dynlayer.alpha=1;
        map.addLayer(dynlayer);
        labelLast=label;
    }
    syncViewList();}
```

### Example of Python code

This is a code segment in order to create a zip file.

```
import sys, zipfile, arcgisscripting, os, traceback

gp = arcgisscripting.create()

def zipws(path, zip):

    isdir = os.path.isdir

    for each in os.listdir(path):

        fullname = path + "/" + each

        if not isdir(fullname):

            if not each.endswith('.lock'):

                try: zip.write(fullname, each)

                except IOError: None

            else:

                for eachfile in os.listdir(fullname):

                    if not isdir(eachfile):

                        if not each.endswith('.lock'):

                            try: zip.write(fullname + "/" + eachfile, \

                                os.path.basename(fullname) + "/" + eachfile)

                            except IOError: None

if __name__ == '__main__':

    try:
```

```

inworkspace = sys.argv[1]

outfile = sys.argv[2]

zip = zipfile.ZipFile(outfile, 'w', zipfile.ZIP_DEFLATED)

zipwps(inworkspace, zip)

zip.close()

gp.setParameterastext(1, outfile)

gp.AddMessage("Zip file created successfully")


except:

    tb = sys.exc_info()[2]

    tbinfo = traceback.format_tb(tb)[0]

    pymsg = "PYTHON ERRORS:\nTraceback Info:\n" + tbinfo + "\nError Info:\n  " + \
        str(sys.exc_type)+ ": " + str(sys.exc_value) + "\n"

    gp.AddError(pymsg)


msgs = "GP ERRORS:\n" + gp.GetMessages(2) + "\n"

gp.AddError(msgs)

```

## REFERENCES

- ACM SIGCHI. (2009). *ACM SIGCHI curricula for human-computer interaction*. Retrieved 4/5, 2010, from [http://old.sigchi.org/cdg/cdg2.html#2\\_1](http://old.sigchi.org/cdg/cdg2.html#2_1)
- Adobe. (2010a). *Adobe flex FAQ*. Retrieved 07/31, 2010, from <http://www.adobe.com/products/flex/faq/>
- Adobe. (2010b). *Flash platform benefits*. Retrieved 07/31, 2010, from <http://www.adobe.com/uk/flashplatform/benefits/>
- Adobe. (2010c). *Flex overview*. Retrieved 08/02, 2010, from <http://www.adobe.com/products/flex/overview/>
- Anthony, F. (2008). *Flex vs. AJAX friends or foes?* EffectiveUI.
- Bell, M. (2008). *Service-oriented modeling: Service analysis, design, and architecture*. Hoboken, N.J.: John Wiley and Sons.
- Busch, M., and Koch, N. (2009). *Rich Internet applications - state-of-the-art* (Technical Report No. 0902). Ludwig-Maximilians-Universität München (LMU), Germany: Institute for Informatics, Programming and Software Engineering (PST).
- CIS. (2003a). *About CIS*. Retrieved 12/18, 2009, from <http://ice-glaces.ec.gc.ca/App/WsvPageDsp.cfm?Lang=engandInid=2andScndLvl=noandID=142>
- CIS. (2003b). *On-line archive*. Retrieved 01/18, 2010, from <http://ice-glaces.ec.gc.ca/App/WsvPageDsp.cfm?ID=11391andLang=eng>
- CIS. (2003c). *Subscribers*. Retrieved 04/12, 2010, from <http://ice-glaces.ec.gc.ca/app/WsvPageDsp.cfm?ID=216andlang=eng>
- CIS. (2007). *How to generate products using IceGraph*. Retrieved Dec. 19th, 2009, from <http://ice-glaces.ec.gc.ca/App/WsvPageDsp.cfm?ID=11873andLang=eng#about>
- CIS. (2009). *Canadian ice service - IceGraph version 1.03*. Retrieved 03/21, 2010, from <http://ice-glaces.ec.gc.ca/IceGraph103/?id=11874andlang=eng>
- CIS. (2010). *Sea ice*. Retrieved July 12th, 2010, from <http://www.ec.gc.ca/glaces-ice/default.asp?lang=Enandn=C7B7B935-1>
- Climate4you.com. (2010). *Sea ice*. Retrieved 0806, 2010, from <http://www.climate4you.com/SeaIce.htm>
- ESRI. (1998). *ESRI shapefile technical description* (White Paper) ESRI. Retrieved from <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

- ESRI. (2009a). *ArcGIS Server 9.3 help: Components of an ArcGIS Server system*. Retrieved Dec 12, 2009, from [http://webhelp.esri.com/arcgisserver/9.3/java/components\\_of\\_server.htm](http://webhelp.esri.com/arcgisserver/9.3/java/components_of_server.htm)
- ESRI. (2009b). *Geodatabase*. Retrieved 08/01, 2010, from <http://www.esri.com/software/arcgis/geodatabase/storage-in-an-rdbms.html>
- ESRI. (2010). *What is python*. Retrieved 08/01, 2010, from [http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/What\\_is\\_Python/002z00000001000000/](http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/What_is_Python/002z00000001000000/)
- Fraternali, P., Rossi, G., and Sánchez-Figueroa, F. (2010). Rich Internet applications. [Rich Internet Applications] *IEEE Internet Computing*, 9-12.
- Hoffer, J., George, J., and Valacich, J. (2005). *Modern systems analysis and design* (Forth Ed.). NJ, USA: Prentice Hall Press Upper Saddle River.
- IICWG. (2007). *Ice information services: socio-economic benefits and earth observation requirements* International Ice Charting Working Group.
- Lawler, J. P. (2008). In Howell-Barber H. (Ed.), *Service-oriented architecture: SOA strategy, methodology, and technology*. Boca Raton: Auerbach Publications.
- Li, S. (2008). Web mapping/GIS services and applications. *Advances in Photogrammetry, Remote Sensing and Spatial Information Science: 2008 ISPRS Congress Book*, 335.
- Lo, C. P., and Yeung, A. K. W. (2007). In Clarke K. C. (Ed.), *Concepts and techniques of geographic information systems* (Second Ed.). USA: Pearson Education Inc.
- Murugesan, S. (2007). Understanding web 2.0. *IT Professional*, 9(4), 34-41.
- NSIDC. (2010). *Sea ice animation tool*. Retrieved 08/05, 2010, from [http://nsidc.org/data/seaice\\_index/archives/image\\_select.html](http://nsidc.org/data/seaice_index/archives/image_select.html)
- OGC. (2004). *OGC web map service interface* (1.3rd Ed.) Open GIS Consortium Inc. Retrieved from [http://portal.opengeospatial.org/files/index.php?artifact\\_id=4756](http://portal.opengeospatial.org/files/index.php?artifact_id=4756)
- Roossien, D. (2009). *Microsoft Silverlight photography framework comparing component based designs in adobe flex and Microsoft Silverlight* (Tech. Rep) Retrieved from <http://www.uop.edu.jo/download/PdfCourses/Silverlight/Silverlight%20Photography%20Framework.pdf>
- Sharma, P. (2008). *Core Characteristics of Web 2.0 Services*. Retrieved 9/12, 2010, from <http://www.techpluto.com/web-20-services/>
- Tang, Y., and Wong, W. D. (2006). Exploring and visualizing sea ice chart data using java-based GIS tools. *Computers and Geosciences*, 32, 846-858.

- Valacich, J., George, J., and Hoffer, J. (2009). *Essentials of system analysis and design* (Forth Ed.). NJ, USA: Prentice Hall Press Upper Saddle River.
- W3C. (2004). *Web services glossary*. Retrieved 12/09, 2009, from <http://www.w3.org/TR/ws-gloss/>
- W3C. (2007). *SOAP version 1.2 part 1: Messaging framework (second edition)*. Retrieved Dec 12, 2009, from <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>
- WMO. (2006). *Sea-ice information services in the world* (Third Ed.) World Meteorological Organization.
- Zhang, M. (2009). *Flex viewer developers guide* (Tech. Rep) ESRI.
- Zhang, P., Carey, J., Te'eni, D., and Tremaine, M. (2005). Integrating human-computer interaction development into the systems development life cycle: A methodology. *Communications of the Association for Information Systems (Volume 15, 2005)*, 15, 512-543.
- zur Muehlen, M.; Nickerson, J.V.; Swenson, K.D.: Developing Web Services Choreography Standards – The Case of REST vs. SOAP. *Decision Support Systems* 37 (2004), Elsevier, North Holland.