

# Journal of Geography

(Jirihak Nonchong)

Monography Series 59

Aug. 2004

**Design and Implementation of a Geocomputing  
Platform for LBS and Modular Mobile Mapping :  
An Approach Based on .NET**

Yang-Won Lee

Institute for Korean Regional Studies, Seoul National University

서울대학교 국토문제연구소 논문집

# 地理學論叢

별호59

2004. 8

LBS와 기능성 모바일매핑을 위한  
지오컴퓨팅 플랫폼의 설계와 구현

이 양 원

서울대학교 국토문제연구소

地理學論叢 별호는 서울대학교 국토문제연구소에서  
펴내는 단행본 시리즈이다. 이 연구물은 편집위원회의  
심의를 거쳐 실린다.

## 편집위원회

편집인 박수진

편집위원회

권용우	김영성	김 인	박기호	박삼욱	박수홍	박영한	손 일
안영진	유근배	유우익	이재덕	이재하	이정록	이정만	이희열
주성재	최병두	허우궁					
Chang, Heejun		Choi, Jong Nam		Kim, Doo-Chul		Park, Bae-Gyoon	
Seong, Jeong-Chang		Yoon, Hong-ki					

편집간사

이호윤 안재성

◇ 題揮

滄海 金昌煥 先生

## Editorial Board

**Editor** Park, Soo Jin

### Editorial Board Members

Ahn, Young-Jin

Choo, Sungjae

Kim, Young-Sung

Lee, Jaeha

Park, Key-Ho

Seong, Jeong-Chang

Yu, Woo-Ik

Chang, Heejun

Huh, Woo-Kung

Kwon, Yongwoo

Lee, Jeongman

Park, Sam-Ock

Son, ILL

Choi, Byung-Doo

Kim, Doo-Chul

Lee, Hee-Yul

Lee, Jeong-Rock

Park, Soo-Hong

Yoon, Hong-Ki

Choi, Jong Nam

Kim, Inn

Lee, Jaeduk

Park, Bae-Gyoon

Park, Young-Han

Yu, Keun Bae

# Abstract

Design and Implementation of a Geocomputing Platform for  
LBS and Modular Mobile Mapping: An Approach Based on .NET

by

Yang-Won Lee

This thesis discussed the design and implementation of a geocomputing platform for LBS and modular mobile mapping. This geocomputing platform targets LBS application development, and focuses on modular mobile mapping as a core element of LBS. Modular mobile mapping which support LBS consists of the functionalities of location information processing, mobile embedded mapping, and mobile internet mapping which play a unique role in the LBS application development. Modular mobile mapping becomes not only a core element of LBS, but also a useful application for mobile GIS.

Major contributions of this thesis include a new approach based on .NET technology and an implementation of the interoperability with generic map servers. During the process of analysis, design, and implementation of the geocomputing platform, the implications can be derived as below.

First, design and implementation of our geocomputing platform was performed on the basis of “supportability of LBS” and “modularity of mobile mapping,” and these principles can play the role of a basic strategy for the development process of a software platform for mobile geocomputing.

The functionalities such as location-awareness, mobile embedded mapping, and mobile internet mapping are necessary for LBS application development, and these functionalities were modularized in our geocomputing platform for the convenient access to a specific functional component and the diverse compatibility with external entities in the process of application development. Functional modularity (or decomposability) is important for arranging complex functionalities for LBS application development in the form of component, and compatible modularity (or composability) is important for the interoperability with the external entities necessary for LBS application development such as map server and database system.

Secondly, design and implementation of our geocomputing platform was performed through the analysis of the major technologies of mobile GIS and LBS, and it provided a clue to the future direction of the development of software platform for mobile geocomputing: “the need of .NET-based programmable component for modular mobile mapping” and “the need of the interoperability with generic map servers.”

The need of .NET-based programmable component arises from the need of complementing the current mobile GIS technology. A close investigation into the characteristics of major mobile GIS products indicates that a programmable component, which allows application developers to build an independently executable program, is not sufficient, and .NET-based technology for modular mobile mapping has not been developed yet. Therefore, it is necessary to consider the component-based programming interface based on .NET environment for the development of mobile geocomputing platform. It provides

the convenience and high performance of .NET mobile computing, as a result of introducing and applying a new technology to GIS domain.

The need of the interoperability with generic map servers for LBS platform is rooted in the compatibility between components of a system and the extensibility of application development. Although major LBS server products may not provide an option for choosing map server for commercial reasons, the interoperability with generic map servers should be taken into consideration for the development of mobile geocomputing platform. The interoperability with generic map servers provides flexible choice of mapping engine to achieve the specific goal of LBS application, thereby allowing the extension of various map data in the region the application requires.

Thirdly, the components of our geocomputing platform can be practically used for the application development of LBS and modular mobile mapping. Those built in the form of DLL and those built in the form of XML Web Services can be referenced in the .NET development environment. These components provide API methods and properties which cover necessary functionalities for application development, and developers can write additional codes for the business logic of a specific domain for their purpose. With the convenience of PNP (Plug and Play), the components save a lot of efforts the developers otherwise have to make to compose the core logic such as GPS functionality with coordinate transformation, embedded mapping, internet mapping, and location information processing.

Keywords: Location Based Services (LBS), Mobile GIS, Mapping Middleware,  
.NET Technology



# Contents

1. Introduction .....	1
1.1. Background .....	1
1.2. Research Objectives.....	7
1.3. Research Methods .....	9
1.4. Thesis Organization .....	11
2. Literature and Technology Review.....	13
2.1. Location-Aware Computing .....	13
2.1.1. Location as Context .....	14
2.1.2. Technology Elements .....	14
2.2. Mobile GIS.....	16
2.2.1. Extension of GIS Environment.....	17
2.2.2. Coupling of Mobile GIS and GPS .....	18
2.2.3. Current Technologies .....	20
2.3. Wireless Map Service .....	23
2.3.1. Request/Response Structure of Map Service .....	24
2.3.2. Technology Standard of Wireless Map Service .....	24
2.4. Location Based Services .....	26
2.4.1. Definition and Meaning .....	26
2.4.2. Key Technology .....	28
2.4.3. Major LBS Server Products.....	32
2.4.4. Service Contents .....	33
2.4.5. Comparison of LBS to Mobile GIS .....	34
2.5. Software Engineering and Software Development.....	35
2.5.1. Object-Oriented Software Engineering.....	36
2.5.2. Unified Modeling Language.....	36
2.5.3. Component-Based Software Engineering.....	37
2.5.4. COTS Interoperability .....	38

2.5.5. Middleware Technology .....	39
2.5.6. .NET-based Technology.....	40
3. Framework Design of Geocomputing Platform.....	47
3.1. Overview .....	47
3.2. Module Composition.....	51
3.2.1. Location-Awareness .....	51
3.2.2. Mobile Embedded Mapping .....	52
3.2.3. Mobile Internet Mapping .....	53
3.3. Development Environment .....	57
3.3.1. Mobile Device and GPS Receiver.....	57
3.3.2. Development Tool.....	59
4. Implementation of Geocomputing Platform .....	60
4.1. Location-Awareness .....	60
4.1.1. Serial Communication .....	61
4.1.2. GPS Data Handling .....	66
4.1.3. Coordinate Transformation .....	72
4.1.4. API Development .....	78
4.1.5. Feasibility Test .....	81
4.2. Mobile Embedded Mapping.....	83
4.2.1. Map Data Pre-processing.....	83
4.2.2. Visualization .....	87
4.2.3. Topological Operation .....	101
4.2.4. API Development .....	106
4.2.5. Feasibility Test .....	109
4.3. Mobile Internet Mapping.....	112
4.3.1. Location Middleware .....	112
4.3.2. Mapping Middleware.....	115
4.3.3. API Development .....	123
4.3.4. Feasibility Test .....	125
5. Application Prototyping .....	128
5.1. Overview .....	128

5.2. Field Data Collection.....	133
5.2.1. Utilization of Field Data Collection .....	133
5.2.2. Sampling Method .....	137
5.2.3. Application Prototype .....	139
5.3. Cadastral Information Service .....	146
5.3.1. LIS and Cadastral Information .....	146
5.3.2. Application Prototype .....	148
6. Conclusion .....	152
6.1. Research Summary .....	152
6.2. Research Implications.....	157
6.3. Future Work .....	159
Bibliography .....	162

# List of Tables

[Table 1] Programming Environment Comparison of Major Mobile GIS Products .....	21
[Table 2] URL Component of HTTP/GET Request.....	24
[Table 3] Programming Environment Comparison of Major LBS Server Products .....	32
[Table 4] LBS Service Contents According to Location Accuracy.....	34
[Table 5] Two Types of Mobile Mapping Environment.....	49
[Table 6] Role of Model, View, and Controller in Graphics Programming.....	53
[Table 7] Workflow of Mobile Internet Mapping .....	54
[Table 8] Western European Handheld Device Shares .....	58
[Table 9] Specification of HP iPAQ h5550.....	58
[Table 10] Specification of Navman GPS 3450 .....	59
[Table 11] Ellipsoid Comparison of WGS84, TM, and ITRF2000 Coordinate System ...	73
[Table 12] 7 Parameters for Moldensky-Badekas Transformation .....	77
[Table 13] Typical API Methods in ClapComm Class .....	78
[Table 14] Typical API Methods in ClapGPS Class .....	79
[Table 15] Typical Properties in ClapGPS Class .....	79
[Table 16] Typical API Methods in ClapCoord Class .....	79
[Table 17] Typical Properties in ClapCoord Class.....	80
[Table 18] Result Comparison of Coordinate Transformation.....	82
[Table 19] Organization of Main File in Shapefile .....	88
[Table 20] Header Description of Main File in Shapefile .....	88
[Table 21] Record Header of Main File in Shapefile .....	89
[Table 22] Contents of Point in Shapefile .....	89
[Table 23] Contents of PolyLine in Shapefile .....	90
[Table 24] Contents of Polygon in Shapefile .....	91
[Table 25] Organization of Index File in Shapefile .....	91
[Table 26] Record Header of Index File in Shapefile.....	92
[Table 27] Characteristics of Header Part in dBase File.....	92

[Table 28] Characteristics of Record Part in dBase File .....	92
[Table 29] Typical API Methods in ClapMap Class .....	107
[Table 30] Typical API Methods in ClapMapRepresenter Class .....	108
[Table 31] Typical Properties in ClapMapRepresenter Class .....	108
[Table 32] Typical API Methods in ClapRect and ClapFeature Class .....	108
[Table 33] Typical Properties in ClapFeature Class .....	108
[Table 34] Example of MapServer CGI Request .....	118
[Table 35] Example of OpenMap HTTP Request .....	120
[Table 36] Description of Object Array for ArcIMS WebMethod .....	123
[Table 37] Typical WebMethod for Basic Visualization.....	124
[Table 38] Typical WebMethod for Advanced Rendering.....	125

# List of Figures

[Figure 1] Circa’s Pizza Story: A Famous Example of LBS .....	2
[Figure 2] Example Usage of Mobile GIS on PDA.....	3
[Figure 3] Research Method of Developing the Geocomputing Platform for LBS and Modular Mobile Mapping .....	11
[Figure 4] Thesis Organization .....	12
[Figure 5] SA Effect on Location Accuracy .....	15
[Figure 6] Extension of GIS Depending on Computing Environment.....	17
[Figure 7] Workflow of “Map Service for Portable Device”.....	26
[Figure 8] GeoMobility Server and OpenLS Architecture .....	29
[Figure 9] Layer Structure of LBS Platform by Korea LBS Standardization Forum ....	31
[Figure 10] Comparison of LBS to Mobile GIS .....	34
[Figure 11] Conceptual Structure of COTS Interoperability .....	39
[Figure 12] Platform Architecture of .NET Compact Framework .....	43
[Figure 13] Comparison of Web Service Peak Throughput .....	45
[Figure 14] Modular Mobile Mapping and LBS Support of the Geocomputing Platform	47
[Figure 15] Overall Structure of the Geocomputing Platform.....	51
[Figure 16] Workflow of GPS Data Management.....	51
[Figure 17] Architecture of Mobile Embedded Mapping in the Geocomputing Platform .....	52
[Figure 18] Architecture of MVC (Model-View-Controller).....	53
[Figure 19] Architecture of Mobile Internet Mapping in the Geocomputing Platform	54
[Figure 20] Workflow of Location Middleware in the Geocomputing Platform .....	55
[Figure 21] Workflow of Mapping Middleware in the Geocomputing Platform .....	56
[Figure 22] Connection between PDA and GPS Receiver .....	59
[Figure 23] Structure of NMEA-0183 Common Sentences .....	67
[Figure 24] Workflow of Coordinate Transformation between WGS84 and TM .....	74
[Figure 25] Testing Screenshot of WGS84 to TM Transformation with or without 10.405	

Seconds Addition.....	81
[Figure 26] Testing Screenshot of GPS Data Monitoring .....	82
[Figure 27] GPS Data Archiving for Traffic Pattern Analysis .....	83
[Figure 28] Vector Simplification of Polygon without Adjustment.....	84
[Figure 29] Shared Boundary Part between Two Polygons .....	85
[Figure 30] Vector Simplification by “Disassemble-Simplify-Reassemble” Method ...	86
[Figure 31] Structure of Double Buffer .....	96
[Figure 32] Default Color Ramp for Choropleth Map .....	98
[Figure 33] Extensibility for Advanced Rendering in the Geocomputing Platform ....	99
[Figure 34] Application of MBR Operation for Viewport Determination .....	102
[Figure 35] Sedgewick’s Test for Point-in-Polygon .....	104
[Figure 36] Testing Screenshot of Color Setting of Polygon.....	109
[Figure 37] Testing Screenshot of Labeling a Map.....	109
[Figure 38] Testing Screenshot of Zooming and Panning .....	110
[Figure 39] Testing Screenshot of Vector-on-Raster Overlay .....	110
[Figure 40] Testing Screenshot of Attribute Query.....	110
[Figure 41] Testing Screenshot of Symbolizing ( I ) .....	111
[Figure 42] Testing Screenshot of Symbolizing ( II ) .....	111
[Figure 43] Testing Screenshot of Choropleth Map ( I ) .....	111
[Figure 44] Testing Screenshot of Choropleth Map ( II ) .....	112
[Figure 45] MBR-Based Hierarchical Operation for Reverse-Geocoding .....	114
[Figure 46] Point-in-Polygon Operation for Reverse-Geocoding .....	114
[Figure 47] Framework of ArcIMS .....	115
[Figure 48] Workflow of Image Compression in the Mapping Middleware .....	121
[Figure 49] Result Comparison of Image Compression in the Mapping Middleware..	123
[Figure 50] Screenshot of Zooming Map on Mobile Web Browser .....	126
[Figure 51] Screenshot of Panning Map on Mobile Web Browser .....	126
[Figure 52] Screenshot of Labeling and Highlighting Map on Mobile Web Browser ..	126
[Figure 53] Screenshot of Choropleth Map on the Mobile Web Browser .....	127
[Figure 54] Screenshot of Symbol Map on the Mobile Web Browser.....	127
[Figure 55] Screenshot of Chart Map on the Mobile Web Browser.....	127

[Figure 56] Geocomputing Platform Extensibility for Geographic Applications .....	128
[Figure 57] Brief Process of Using Components of the Geocomputing Platform for Location-aware Mobile Mapping Application in the Embedded Environment ...	129
[Figure 58] Brief Process of Using Components of the Geocomputing Platform for Mobile Internet Mapping and LBS Applications.....	131
[Figure 59] Screenshot of Existing Field Data Collection Applications.....	136
[Figure 60] Workflow of Field Data Collection Application .....	140
[Figure 61] Use Case Diagram for Field Data Collection Application.....	142
[Figure 62] Screenshot of Field Data Collection Application .....	145
[Figure 63] Web Interface of Seoul Land Information Service .....	148
[Figure 64] Sequence Diagram for Parcel Information Service.....	150
[Figure 65] Screenshot of Cadastral Information Service .....	151
[Figure 66] Extended Architecture of the Geocomputing Platform .....	161

# 1. Introduction

## 1.1. Background

One of the interesting changes the improvement of IT (Information Technology) has brought about is the increased mobility. Now we live in the era of “one cell phone per person” in Korea, and there is a strong tendency towards the widespread use of wireless internet<sup>1</sup>. In addition, the handheld devices like PDA (Personal Digital Assistant)<sup>2</sup> enable us to carry computing functionality with us (Lyytinen & Yoo, 2002).

Amid this, the term LBS (Location Based Services) or mobile GIS (Geographic Information System) is emerging. LBS is a wireless information service that uses the location information of a mobile user. The convergence of multiple technologies including GIS, internet, wireless communication, location determination, and portable device has given rise to the LBS (Koeppel, 2000). To provide various information based on the location of a mobile user, LBS requires GIS functionality for mapping user’s current location and user’s interest area through the wireless internet. This location information is acquired by the location determination technology. The geographic information created with the location information is represented on the screen of a portable device.

“Circa’s Pizza Story” suggested in the 1<sup>st</sup> LIF<sup>3</sup> Meeting is one of the most

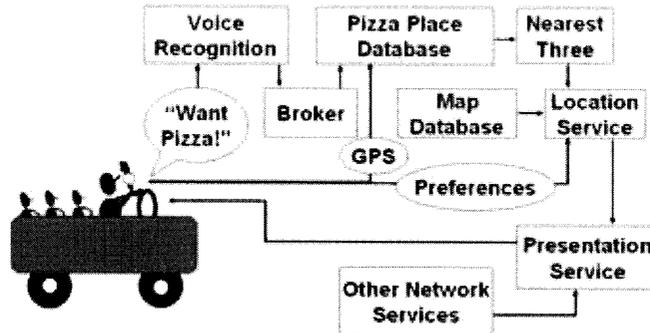
---

<sup>1</sup> According to Korea Network Information Center (2003), 37 % out of the 33 million cell phone users in Korea in 2003 have experienced wireless internet. In particular, this trend prevails among young generation or high-educated people.

<sup>2</sup> PDA is a term for any small mobile handheld device that provides computing and information storage and retrieval capabilities for personal or business use (<http://whatis.com>).

<sup>3</sup> The LIF (Location Interoperability Forum) has consolidated into the Open Mobile Alliance

famous examples of LBS (Hecht, 2000). In this example, a mobile user, moving by a car, is provided with the information about pizza stores in his/her immediate surroundings [Figure 1].



[Figure 1] Circa's Pizza Story: A Famous Example of LBS

As Beinat (2001) observes, there is a general consensus amongst analysts that LBS will be the most widely used mobile services. For example, the ARC Group<sup>4</sup> predicts that location based services will have become the most popular mobile services by 2005. IDC<sup>5</sup> estimates that almost 50% of European subscribers will use LBS by 2005, implying a potential user base close to 150 million. The Strategies Group<sup>6</sup> predicts that LBS will surpass the GIS market in terms of size in the period 2002-2004. Even though these predictions may not

---

(OMA), and no longer exists as an independent organization. The OMA Location Working Group (LOC) continues the work originated in the former Location Interoperability Forum (LIF) and Location Drafting Committee of the former WAP (Wireless Application Protocol) Forum. (<http://www.openmobilealliance.org/tech/affiliates/lif/lifindex.html>)

<sup>4</sup> The ARC Group (Analysis, Research and Consultancy Group) has built its reputation on analyzing mobile communications and multimedia strategic issues, market trends and technologies from a realistic perspective (<http://www.arcgroup.com>).

<sup>5</sup> IDC is the premier global market intelligence and advisory firm in the information technology and telecommunications industries, which has 700 analysts in 50 countries (<http://www.idc.com>).

<sup>6</sup> <http://www.strategiesgroup.com>

be fully realized, it is true that the interest in LBS is growing worldwide.

In Korea, the Ministry of Information and Communication (2003) announced LBS Industry Support Plan, which focuses on promoting ( i ) base technology, ( ii ) analysis technology, and ( iii ) industry support, aiming to construct “U-Korea (Ubiquitous Korea)<sup>7</sup>.”

In addition to the growing interest in LBS, the role of mobile GIS is becoming increasingly important in the GIS field. GIS technology can be categorized into desktop GIS, internet GIS, and mobile GIS depending on the computing environment it is working in. Each of them works closely with each other, and has its own role: desktop GIS is for spatial analysis, internet GIS for geographic information service on the web, and mobile GIS for the geographical work which requires mobility. Mobile GIS is a compact GIS which brings the functionality of desktop or internet GIS into a mobile device like PDA (Stockus, 1999) [Figure 2].



[Figure 2] Example Usage of Mobile GIS on PDA

---

<sup>7</sup> U-Korea is the motto of LBS Industry Support Plan, which has the goal of building a ubiquitous computing environment (anytime/anywhere computing) in Korea.

Mobile GIS can be combined with GPS (Global Positioning System) to acquire real-time location information of a mobile user, if necessary. Field data collection is one of the most typical geographic applications that rely on mobile GIS and GPS (Joyce, 2003). Environmental or socioeconomic research and facility inspection for pipes, valves and meter gauges of water, gas, and electricity in the enterprise GIS<sup>8</sup> often require mobile GIS and GPS for field data collection (Chen & Lee, 2001; Harrington, 2003a).

With the emergence of LBS and mobile GIS, various technologies have been developed by major IT service providers and organizations. Microsoft released MapPoint Location Server (MLS), the most widely used LBS server product, and ESRI released ArcPad, the most widely used mobile GIS product. The international non-profit organizations such as OGC (Open GIS Consortium)<sup>9</sup> and ISO (International Organization for Standardization)<sup>10</sup> have established LBS-related standard specifications.

In Korea, three telecommunication service providers, SK Telecom<sup>11</sup>, KTF<sup>12</sup>, and LG Telecom<sup>13</sup> provide location based services on their cellular network. They include services such as buddy-finding, store-finding, and traffic information (Haeock Choi, 2003). Korea Wireless Internet Standardization

---

<sup>8</sup> Enterprise GIS manages the visible/invisible resources of companies or public organizations in an integrated system.

<sup>9</sup> The Open GIS Consortium (OGC) is an international industry consortium of 259 companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications (<http://www.opengis.org>).

<sup>10</sup> <http://www.iso.org>

<sup>11</sup> <http://www.sktelecom.com>

<sup>12</sup> <http://www.ktf.co.kr>

<sup>13</sup> <http://www.lgtelecom.com>

Forum (KWISF)<sup>14</sup> and Korea LBS Standardization Forum<sup>15</sup> aim to develop Korea's own wireless internet standard and LBS standard which are compatible with international standards.

These efforts to improve LBS and mobile GIS have produced many fruitful results. However, from the viewpoint of application developers of LBS and mobile GIS, there are still a few things which are necessary, but not covered by the major IT service providers and organizations. One of them is the development of ".NET-based technology" and another is the "interoperability with generic map servers."

.NET-based technology .NET is a new programming platform<sup>16</sup> which covers Windows, internet, and mobile environment. Windows-based mobile device such as Pocket PC<sup>17</sup> and Smartphone<sup>18</sup> is recently emerging because of the convenience and familiarity associated with Windows. The .NET-based technology provides excellent performance and compatibility with the Windows-based mobile device. In addition, .NET-based XML Web Services<sup>19</sup> provide very efficient web services which are necessary for the communication between client and server in the LBS and mobile GIS. Several benchmarking

---

<sup>14</sup> Korea Wireless Internet Standardization Forum, composed of research organizations and corporations, performs research and development of the Korean standards and specifications for wireless internet. (<http://129.254.10.56/index.html>)

<sup>15</sup> Korea LBS Standardization Forum is composed of telecommunication-related research organizations and hardware/software corporations (<http://www.lbskorea.or.kr>).

<sup>16</sup> Generally in computers, a platform is an underlying computer system on which application programs can run (<http://whatis.com>). Particularly in software, platform is an underlying software system which can serve for application development.

<sup>17</sup> PDA with Windows-based operating system

<sup>18</sup> Cell phone with Windows-based operating system

<sup>19</sup> XML (Extensible Markup Language) is an advanced standard for information exchange format on the web. XML Web Services play the role of managing web applications using XML-based communication between client and server.

studies show the .NET-based XML Web Services outperform other web application services (Edwards, 2002; The Middleware Company, 2003).

Until now, Java has been the mainstream development platform for cell phone application, and eMbedded Visual Tools have been the mainstream development platform for PDA application. However, .NET, as a new software platform, is also required for the diversification of the base technology for mobile computing. This diversification provides application developers with more flexibility to choose alternative development platform for mobile application development.

.NET-based technology is as important in the area of LBS and mobile GIS as it is in generic mobile computing. Introducing .NET-based technology to LBS and mobile GIS mean not only adopting a new technology, but also applying the new technology to LBS and GIS domain so as to provide more efficient and diverse application development environment.

Another aspect, the need of “interoperability with generic map servers” is based on the role of mobile mapping in LBS application. Mobile mapping, which allows intuitive grasp of geographic information and on-the-spot use of map on the mobile device, is not only an independent functionality which is applicable to various mobile applications including mobile GIS, but also one of the key functionalities of LBS application.

Though the major LBS server products provide useful functionalities of mobile mapping, they are only compatible with their own map servers. They do

not support generic map servers<sup>20</sup>. The interoperability with generic map servers is important because the interoperability means the compatibility between components and the extensibility of an entire system in building applications.

In this context, the interoperability with generic map servers provides flexibility in choosing mapping engines according to the specific goal of LBS application, thereby allowing the extension of various map data such as administrative district map, road map, land parcel map, and topological map by adopting the right server in the right application. With this interoperability, application developers do not need to be confined to a specific map server: on the contrary, they can utilize the workflow of the map server they choose, which plays an important role in mobile mapping for LBS application.

This study starts from the above two problems: “the need of .NET-based technology” and “the need of the interoperability with generic map servers” for the application development environment of LBS and mobile mapping. This study will focus on developing an appropriate software platform which can be used as an experimental model for the application development environment of LBS and mobile mapping by solving these problems.

## 1.2. Research Objectives

Based on the examination of current technologies of LBS and mobile GIS, we will develop a “geocomputing platform” which serves for LBS applications and

---

<sup>20</sup> It may be for the commercial reason.

mobile mapping functionalities. This geocomputing platform, as a kind of software platform, means structured software which underlies the applications of LBS and mobile mapping. It provides the base environment for application development in the form of programming interface.

The specific objectives of the research are:

- To build a system framework of the geocomputing platform based on the examination of major mobile GIS products and LBS server products in addition to the generic requirements of the application development environment of LBS and modular mobile mapping. Meeting the generic requirements will cover basic functionalities for LBS and modular mobile mapping. Analyzing the key features of current major products will identify technical niches to be filled for the application development environment of LBS and modular mobile mapping.
- To develop components of the geocomputing platform for LBS and modular mobile mapping suitable for the system framework. The component development will particularly focus on the technical niches that major mobile GIS products and LBS server products do not cover: the need of .NET-based technology and the need of the interoperability with generic map servers.
- To integrate the components into the geocomputing platform and to provide the application development environment of LBS and modular mobile mapping through the integrated geocomputing platform. The application development can be performed by extending necessary components of the geocomputing platform and composing additional codes

of the business logic for a specific domain.

### 1.3. Research Methods

The geocomputing platform to be discussed in this study is for LBS and mobile mapping. Mobile mapping provides intuitive grasp of geographic information for LBS application and on-the-spot use of map for mobile GIS. Mobile mapping is an independent functionality for various mobile applications, and also one of the key elements for LBS.

Our geocomputing platform is based on “supportability of LBS” and “modularity of mobile mapping.” As a key element of LBS, mobile mapping requires several functionalities which should be wrapped in modules. To achieve the LBS support, each functional module for mobile mapping is organized in the application development process. Thus, the geocomputing platform should be modularized by functionality to provide necessary functionality for application development of LBS and mobile mapping.

In broad terms, modularity is an approach for organizing efficiently the design and production of complex products and processes (Baldwin & Clark, 1997). Complex tasks are decomposed into simpler elements so that they can be managed independently and yet operate together as a whole. A motivation behind decomposition of a complex system into more manageable parts is to gain flexibility and cost savings. (Mikkola, 2003)

Particularly in the software engineering, modularity serves to produce a software system made of autonomous elements (modules) connected by

coherent, simple structures<sup>21</sup>. “Modularity of mobile mapping” in our geocomputing platform is related to two aspects: functional modularity within the platform and compatible modularity with other entities in the application development environment. The functional modularity<sup>22</sup> means that necessary functionalities (e.g., location-awareness of a mobile user, thematic mapping, etc.) are grouped by module in the platform, and the compatible modularity<sup>23</sup> means the modules of the platform can communicate with external entities or components (e.g., generic map servers, database system, etc.) in application development environment. The functional modularity and compatible modularity of the geocomputing platform supports the application development of LBS by providing necessary functional modules and by coupling with necessary external components [Figure 3].

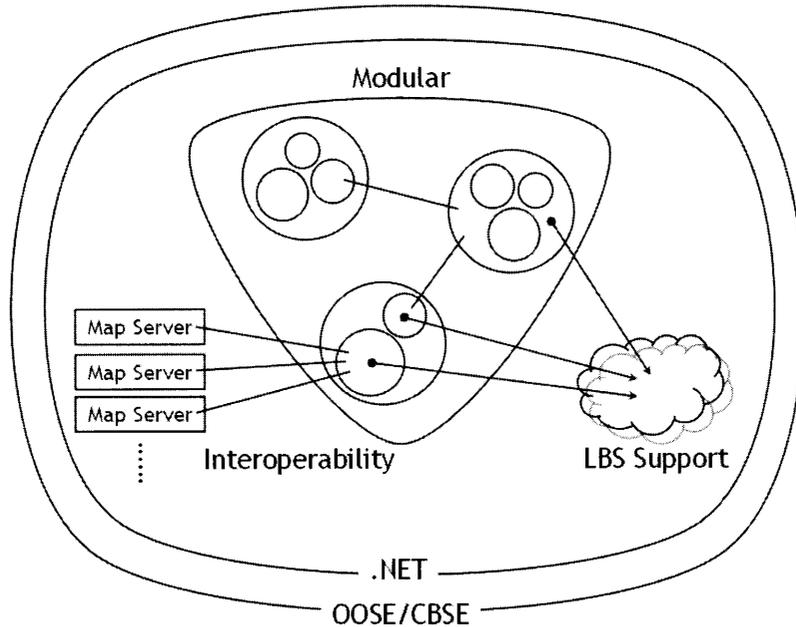
---

21

[http://www.cmis.brighton.ac.uk/Research/Archive/noolearning/facilits/exemplars/oovb/schedule/section14/ab21\\_2.htm](http://www.cmis.brighton.ac.uk/Research/Archive/noolearning/facilits/exemplars/oovb/schedule/section14/ab21_2.htm)

<sup>22</sup> Our term “functional modularity” corresponds to “decomposability” in the criteria for modularity evaluation.

<sup>23</sup> Our term “compatible modularity” corresponds to “composability” in the criteria for modularity evaluation.



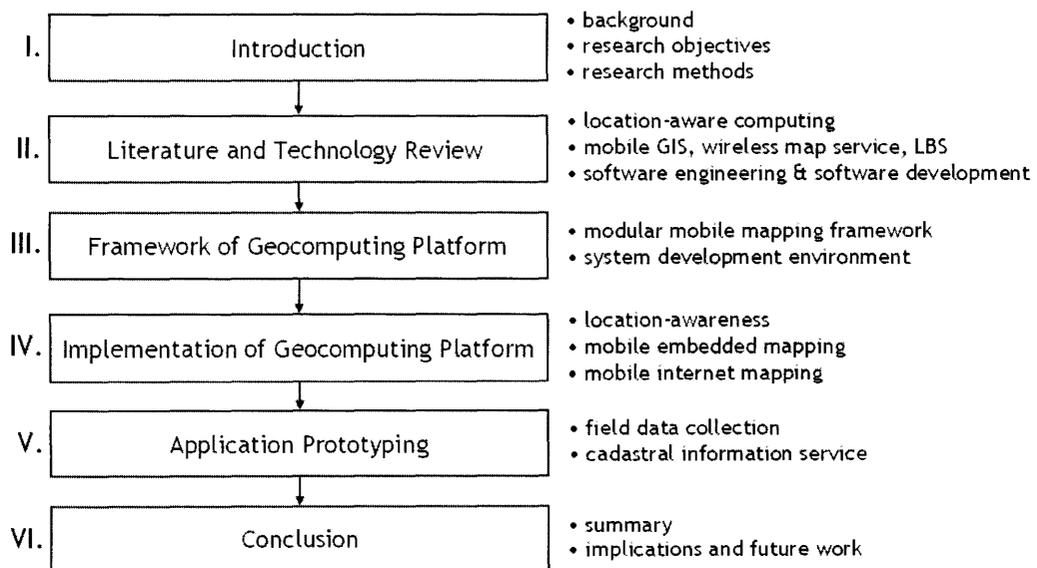
[Figure 3] Research Method of Developing the Geocomputing Platform for LBS and Modular Mobile Mapping

.NET will be the development environment for building our geocomputing platform, and OOSE (Object-Oriented Software Engineering) and CBSE (Component-Based Software Engineering) methods will be used for the overall development process. OOSE provides object-based reusability and extensibility; CBSE provides component-based compatibility with necessary external entities.

## 1.4. Thesis Organization

We have observed the background, objectives, and methods of the study in this chapter. In chapter 2, we will explore the literature and technologies with respect to location-aware computing; mobile GIS; wireless map service; location based services; software engineering and software development. Based

on the literature and technology survey, the system framework of our geocomputing platform will be built in chapter 3. Chapter 4 will describe the implementation process of the geocomputing platform suitable for the system framework. Chapter 5 will show the application prototype which verifies the feasibility of the geocomputing platform. The final chapter, chapter 6 will present a summary and conclusion; the conclusion will include the implications, limitations, and future work of this study. [Figure 4] illustrates thesis organization.



[Figure 4] Thesis Organization

## 2. Literature and Technology Review

This chapter covers the survey of location-aware computing; mobile GIS; wireless map service; location based services; software engineering and software development. These technological items are closely related to the geocomputing platform we will develop. The geocomputing platform, which serves for the application development of LBS and mobile mapping, requires location-awareness of a mobile user for providing geographic information according to the user's location. Mobile GIS and wireless map service deal with mobile mapping for map representation on the mobile device. The recommended architecture and service of LBS platform are presented by technological review of location based services. In addition, the software engineering part deals with necessary methodologies for the development of the geocomputing platform.

### 2.1. Location-Aware Computing

The target geocomputing platform aims to provide the functionalities necessary for LBS and mobile mapping applications which utilize the real-time location information of mobile user. Location-awareness is made possible by the convergence of three distinct technical capabilities: location sensing, wireless communication, and mobile computing system (Computer Science and Telecommunications Board, 2003). The location of mobile user equipped with "mobile computing system" is determined by "location sensing" technology, and transmitted by "wireless communication" between client and server.

### 2.1.1. Location as Context

The term location-awareness is based on context-awareness, in which context means the physical and social situation in which computational devices are embedded. To be more specific, context is defined as any information that can be used to characterize the situation of an entity, where an entity can be a person, time, place, or computational object (Dey & Abowd, 1999). Therefore, the awareness of a person, time, place, and computational object is referred to as user-awareness, time-awareness, location-awareness, and device-awareness, respectively.

The goal of context-aware computing is to acquire and utilize information about the context (Moran & Dourish, 2001). One of the most critical aspects of the context is location: location-aware computing focuses on location as context (Computer Science and Telecommunications Board, 2003). Location-aware computing responds to a user's location, either spontaneously or when activated by a user request (Patterson et al., 2003), and involves the automatic tailoring of information based on the current location of the user (Banerjee et al., 2001).

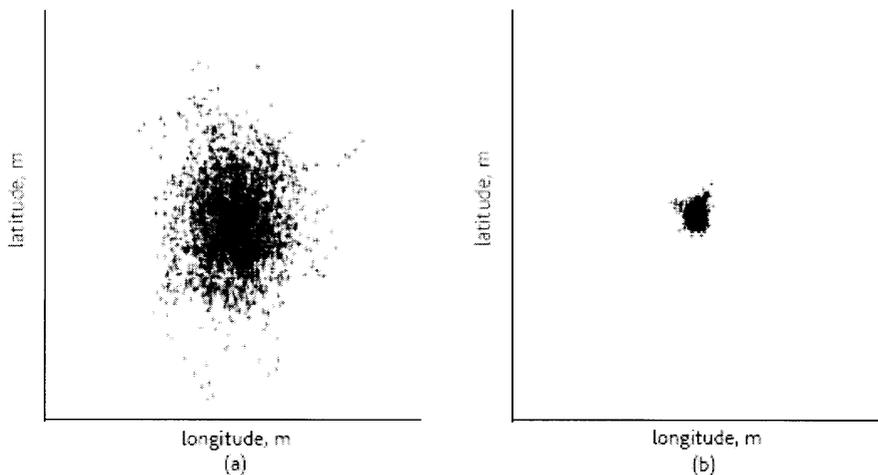
### 2.1.2. Technology Elements

#### 2.1.2.1. Location Sensing

The Global Positioning System (GPS) is the most widely known location sensing system today. This system runs with four or more satellites dedicated to global positioning in the earth. GPS is composed of three main parts: space segment, control segment, and user segment. Space segment corresponds to GPS satellites; control segment, also called ground segment, consists of master

control station, monitor stations and ground antennas; user segment is related to the equipment like GPS receiver (Parkinson, 1996; Kyung-Chan No, 2001).

One of the most important events on the GPS development is the SA removal. In the early days of civilian GPS, the signal was intentionally scrambled through a process known as “SA (Selective Availability)” - degrading the accuracy to around 100 meters for the military reason. On May 1<sup>st</sup> 2000, President Clinton ordered that SA be turned off<sup>24</sup>, enabling typically 3-15 m location accuracy without any other correction. This decision was made to encourage the already growing GPS applications. The effect of SA is described in [Figure 5]: the location accuracy is illustrated in (a) with SA, and in (b) without SA. (D’Roza & Bilchev, 2003; Jae-Kwan Yun, 2003)



[Figure 5] SA Effect on Location Accuracy

<sup>24</sup> [http://www.navcen.uscg.gov/gps/selective\\_availability.htm](http://www.navcen.uscg.gov/gps/selective_availability.htm)

### 2.1.2.2. Wireless Communication

Wireless communication means data communication through wireless network. One of the most important events in the development of wireless communication was the creation of the IEEE 802.11 standard in 1997. The IEEE 802.11 standard sets the protocols used between a wireless client (mobile device) and a base station (access point) or between two wireless clients. The 802.11b, also called Wi-Fi (Wireless Fidelity), increases bandwidth to 11 Mbps, thus is currently the most widely used standard. (Drew, 2003)

The next generation technology for high-speed wireless internet will be OFDM (Orthogonal Frequency Division Multiplexing), a method of digital modulation in which a signal is split into several narrowband channels at different frequencies. (Electronics and Telecommunications Research Institute, 2003; Hanaro Telecom, 2003; Thrunet Corporation, 2003)

### 2.1.2.3. Mobile Computing System

Mobile computing system for location-aware computing means the hardware of handheld device such as PDA and cell phone. These devices provide the capabilities of information processing through the wireless communication. (Patterson et al., 2003)

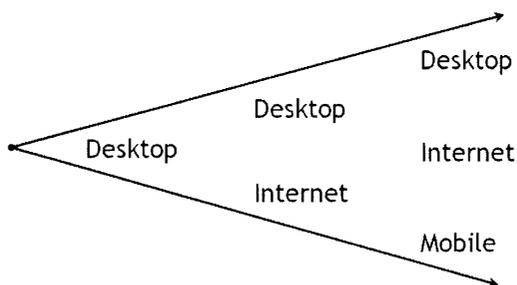
## 2.2. Mobile GIS

Mobile GIS is a kind of compact GIS of desktop or internet GIS. One of the most important functionalities of mobile GIS is providing geographic information and

representing a map on the mobile device. The technological elements of mobile GIS are utilized in the LBS platforms and applications in which mobile mapping plays an important role for information delivery.

### 2.2.1. Extension of GIS Environment

Depending on its computing environment, GIS technology is categorized into desktop GIS, internet GIS, and mobile GIS [Figure 6]. Coexisting with close cooperation with each other, each type of GIS has its own role; desktop GIS is used for spatial analysis, internet GIS for geographic information service on the web, and mobile GIS for the geographical work which requires mobility.



[Figure 6] Extension of GIS Depending on Computing Environment

As a state of the art in the GIS environment, mobile GIS means the geographic information system that works on the mobile device such as PDA and cell phone. Seoul Development Institute (2002) defines Mobile GIS as a system that is equipped with various GIS applications including mapping system, and gathers geographic information without any spatial restriction. It covers offline and online mobile GIS as well that is operated on the devices like PDA or cell phone with the support of wireless network. The Seoul Metropolitan

Government is preparing “Mobile GIS Solution for Civil Administration” that will be used for the business domains like tax, industry, tourism, social welfare, medical care, disaster management, environment, urban planning, cadastre, water service, sewage system, construction, housing, road, transportation, etc. (Seoul Development Institute, 2002; 2003)

Mobile GIS can bring the functionality of desktop GIS or internet GIS into mobile devices. It gives the possibility of using a similar kind of application within both kind of computing environments: stand-alone and client/server. (Stockus, 1999) Today, mobile GIS bridges the gap between the office and field with a direct physical link to enterprise GIS. Limitations exist related to the memory capacity of mobile devices, but this will not always be the case. (Harrington, 2002) For the improvement of mobile GIS, Si (2003) underscores the importance of GIS Database construction suitable for mobile computing environment and common API (Application Programming Interface)<sup>25</sup> standard.

### 2.2.2. Coupling of Mobile GIS and GPS

If mobile GIS is combined with GPS, the functionality of mobile GIS will become richer. The structure of coupling mobile GIS with GPS data consists of ( i ) GPS data acquisition on the mobile devices, ( ii ) GPS data handling on the mobile devices, and ( iii ) GPS data mapping on the mobile devices.

- GPS data acquisition: Most GPS receivers adopt serial communication<sup>26</sup>

---

<sup>25</sup> API is the specific method prescribed by a computer operating system or by an application program by which a programmer writing an application program can make requests of the operating system or another application (<http://whatis.com>).

<sup>26</sup> data communication using a serial port which sends and receives data one bit at a time over one wire

based on RS-232 standard<sup>27</sup> for exchanging data with other devices. The serial communication between GPS receiver and mobile device is necessary to acquire GPS data.

- GPS data handling: The data acquired from a GPS receiver is composed of NMEA-0183 standard<sup>28</sup> that has 6 types of common sentences: GPGLL (Global Positioning System Fixed Data), GPGLL (Geographic Position - Latitude/Longitude), GPGSA (GNSS DOP and Active Satellites), GPGSV (GNSS Satellites in View), GPRMC (Recommended Minimum Specific GNSS Data), and GPVTG (Course over Ground and Ground Speed) (National Marine Electronics Association, 2002). The structure of these sentences should be analyzed to handle GPS data on the mobile devices.
- GPS data mapping: GPS data adopts WGS84 (World Geodetic System 1984) that is the recent, widely accepted coordinate system with GRS80 ellipsoid<sup>29</sup> (Kennedy, 2002). In case of Korea, however, national base map for civilians adopts TM (Transverse Mercator) coordinate system with Bessel 1841 ellipsoid. WGS84 to TM transformation is necessary for applying GPS data in WGS84 coordinate system to the maps of Korea in TM coordinate system.

---

<sup>27</sup> RS-232 is a long-established standard that describes the physical interface and protocol for relatively low-speed serial data communication between computers and related devices (<http://whatis.com>).

<sup>28</sup> NMEA-0183 Interface Standard defines electrical signal requirements, data transmission protocol and time, and specific sentence formats for serial communication. It was originally devised by the United States NMEA (National Marine Electronics Association) for marine equipments. (<http://www.nmea.org/pub/0183>)

<sup>29</sup> GRS80 is a global geocentric system based on the ellipsoid adopted by the International Union of Geodesy and Geophysics (IUGG) in 1979. GRS80 is acronym for the Geodetic Reference System 1980. (Kennedy, 2002)

## 2.2.3. Current Technologies

### 2.2.3.1. Mobile GIS Products

Currently, the major mobile GIS products include ESRI ArcPad, MapInfo MapX Mobile, and Pocket Systems PocketGIS, which run on the Windows CE-based operating system<sup>30</sup>. ArcPad is a package program which provides database access, mapping, GIS, and GPS integration to users in the field via handheld and mobile devices (ESRI, 2002). It provides the functionalities as follows, and users can customize ArcPad environment and compose VBScript applets using these functionalities in the ArcPad Application Builder.<sup>31</sup>

- Support for industry-standard vector and raster image display
- Creating and editing spatial data using input from the mouse pointer, pen, or GPS
- Mobile geodatabase editing to check out your GIS data using ArcGIS, edit in the field with ArcPad, and post changes back to the central GIS database
- GPS integration including GPS navigation
- Map navigation including pan and zoom, spatial bookmarks, and center on the current GPS position
- Data query to identify features, display hyperlinks, and locate features
- Map measurement of distance, area, and bearings
- ArcIMS client for data access via wireless technology
- Application development to automate GIS fieldwork

---

<sup>30</sup> Windows CE, Pocket PC, and Windows Mobile are the typical type of Windows CE-based operating systems for PDA, and Pocket PC Phone Edition is for cell phone.

<sup>31</sup> <http://esri.com/software/arcgis/arcpad>

MapX Mobile is a programmable ActiveX control for creating map-based applications for mobile developers using eMbedded Visual Tools (MapInfo Corporation, 2002). Pocket Systems PocketGIS is a package program for field data capture which can display and edit map geometry and attributes. Users can customize PocketGIS environment using plugin components provided by third party vendors. (Pocket Systems, 2004) Both of them have similar functionality to that of ArcPad.

To evaluate mobile GIS products, we need to focus on two aspects: functionality and programmability. Programmability means the ability of providing programming environment for application developers, and the typical example of programming environment is component library or API function. The case of ArcPad and MapX Mobile shows the pros and cons of current mobile GIS products with respect to the functionality and programmability. In the ArcPad environment, developers can use the richest functionality for composing applications in the form of VBScript applet, whereas every application on the client-side requires ArcPad installation because these applications run only under the ArcPad. In case of MapX Mobile, the available functionality is smaller than ArcPad, whereas it provides the ability of composing independently executable module on Windows. [Table 1] illustrates the comparison of these products in the aspect of programming environment.

[Table 1] Programming Environment Comparison of Major Mobile GIS Products

Product	Type	Customization or application development
ArdPad	Package program	Customization using ArcPad Application Builder
MapX Mobile	ActiveX control	Application development using eMbedded Visual Tools
PocketGIS	Package program	Customization using third party plugin

### 2.2.3.2. Emerging Technologies

Currently emerging technologies applicable to mobile GIS are GML (Geography Markup Language), WFS (Web Feature Service), SVG (Scalable Vector Graphics), etc. GML is XML-based standard data format for geographic information; WFS is a vector map service based on GML for providing feature data on the web; SVG is XML-based data format for vector graphics applicable to web mapping.

GML is an XML encoding in compliance with ISO 19118<sup>32</sup> for the transport and storage of geographic information observing the conceptual modeling framework used in the ISO 19100 series<sup>33</sup> and including both the spatial and non-spatial properties of geographic features. GML version 3.1 represents geospatial phenomena in addition to simple 2D linear features, including features with complex, non-linear, 3D geometry, features with 2D topology, features with temporal properties, dynamic features, coverages, and observations (Open GIS Consortium, 2004a).

While WMS (Web Map Service) provides image map over the web, WFS (Web Feature Service) provides access to geographic feature data in the form of GML over the web. There are two levels of Web Feature Service - Basic and Transaction. Basic Web Feature Service provides read-only access to feature data, and Transaction Web Feature Service extends this service with the ability to create, update, and delete features (Open GIS Consortium, 2002).

In addition, several XML-based specifications for describing vector graphic elements have been developed, including SVG (Scalable Vector Graphics), VML

---

<sup>32</sup> project of ISO/TC211 (Geographic Information - Encoding)

<sup>33</sup> project of ISO/TC211 (Geographic Information)

(Microsoft's Vector Markup Language), and X3D, the XML incarnation of the syntax and behavior of VRML (Virtual Reality Markup Language)<sup>34</sup>.

The efforts to utilize GML for mobile computing environment are now being made as in the case of De Vita et al. (2003), Electronics and Telecommunications Research Institute (2004), and You et al. (2004). In addition, the World Wide Web Consortium (W3C) has established two mobile profiles for SVG. The first profile, SVG Tiny, is defined to be suitable for cell phone; the second profile, SVG Basic, is suitable for PDA. The advantages of SVG are the fancy visualization and small size of vector graphics. The eSVG (embedded SVG) project<sup>35</sup> implements SVG Mobile specifications: it provides eSVG IDE (Integrated Development Environment) and eSVG ActiveX for eMbedded Visual Tools.

## 2.3. Wireless Map Service

Map service means the service which provides geographic information based on the communication protocol between client and server (Korea Wireless Internet Standardization Forum, 2004a). The type of map service is divided into WMS (Web Map Service) and WFS (Web Feature Service) depending on the map providing method. WMS method represents a map in the form of image, whereas WFS method represents a map in the form of vector which has geometry information (Open GIS Consortium, 2002; 2004c).

---

<sup>34</sup> <http://www.w3.org/Mobile/posdep/GMLIntroduction.html>

<sup>35</sup> <http://www.embeddedsvg.com>

### 2.3.1. Request/Response Structure of Map Service

Both WMS and WFS adopt HTTP/GET and HTTP/POST as a request/response method for the data communication between client and server<sup>36</sup>. Request means the invocation of the operation of server; response returns the result of the operation from server to client. HTTP/GET method transmits information in the form of “string” through URL, and uses name-value pair of server-specific parameter list [Table 2]. HTTP/POST uses HTTP message body to transmit the parameter list instead of URL.

[Table 2] URL Component of HTTP/GET Request

URL Component	Description
http://host[:port]/path[?{name[=value]&}]	[ ] - optional; { } - n times iteration
name=value&	Name-value pair for server operation

If a map server gets a valid request, it performs the functionality according to the request and transmits the response object as a result. The response object composed of MIME (Multipurpose Internet Mail Extensions)<sup>37</sup> type is parameterized<sup>38</sup> in the server-specific output format for the transmission to client. (Goode et al., 2002; Open GIS Consortium, 2003; 2004)

### 2.3.2. Technology Standard of Wireless Map Service

Wireless map service has the similar process structure to that of generic map

<sup>36</sup> In addition to HTTP/GET and HTTP/POST, SOAP (Simple Object Access Protocol) is newly being introduced to HTTP request/response structure. SOAP is a way for a program running in one kind of operating system to communicate with a program in the same or another kind of an operating system by using HTTP and its XML as the mechanisms for information exchange.

<sup>37</sup> MIME is an extension of the original internet e-mail protocol that lets people use the protocol to exchange different kinds of data files on the internet: audio, video, images, application programs, and other kinds, as well as the ASCII text (<http://whatis.com>).

<sup>38</sup> In the form of “type/subtype; param1=value1; param2=value2; .....”

service on the web because it is a heritage of the generic map service. The technology standard of “Map Service for Portable Device” of Korea Wireless Internet Standardization Forum (2004a) presents the standards of client request for map and POI data<sup>39</sup>; request acceptance and analysis by gateway<sup>40</sup>; gateway request for data search and response of result data according to the request; map representation technology on the portable device.

The workflow of wireless map service is described in [Figure 7]. If a mobile client such as PDA and cell phone issues a request for map or POI data, gateway accepts and analyzes the request and issues a new request for data search to map server. If the map server operates the functionality according to the request, result data returns to client via the gateway. The gateway produces lightweight map data when the client issues lightweightedness<sup>41</sup> request. The result data transmitted by the gateway to the client is in the form of WKB (Well Known Binary)<sup>42</sup> for vector map, GML for POI data, and BMP, JPEG, GIF for image map. The result data is represented on the portable device by WIPI (Wireless Internet Platform for Interoperability)<sup>43</sup>.

---

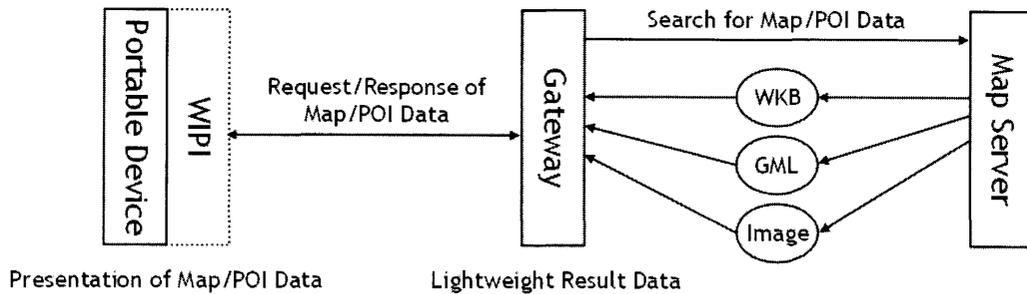
<sup>39</sup> Geographic and attribute information related to the location of Point-of-Interest (POI)

<sup>40</sup> A system connecting client with map server, location server, or portal server

<sup>41</sup> The process of diminishing the size of map data considering the transmission and representation of map data (Korea Wireless Internet Standardization Forum, 2004a)

<sup>42</sup> Geographic information is stored in continuous stream of bytes.

<sup>43</sup> Mobile platform standard for executing an application program downloaded through wireless internet (Korea Wireless Internet Standardization Forum, 2004b)



[Figure 7] Workflow of “Map Service for Portable Device”

## 2.4. Location Based Services

LBS, as an information service which utilizes real-time location information of mobile user, requires multiple technologies such as location-aware computing, mobile GIS, and wireless map service mentioned in the previous sections. In these technologies, necessary functionalities for LBS such as location-awareness, mobile mapping are integrated into LBS platform, and current major LBS server products play the role of LBS platform.

### 2.4.1. Definition and Meaning

The term “LBS (Location Based Services)<sup>44</sup>” means the mobile services in which mobile user’s location information is used in order to add value to the services as a whole<sup>45</sup>. Among many definitions, below are the generally accepted ones (Key-Ho Park et al., 2003).

- 3GPP (3<sup>rd</sup> Generation Partnership Project): standardized services with

<sup>44</sup> Location based services are also called location services, location dependent services, mobile location services, or wireless location services (MapInfo, 2000; Hjelm, 2002).

<sup>45</sup> <http://www.northstream.se/download/LocationBasedServices.pdf>

network support which provide location based applications

- FCC (Federal Communications Commission): providing mobile users with the services based on their geographic location
- ISO (International Standardization Organization): any services, query or process whose return is dependent on the location of the client requesting the service and/or of some other things, objects or persons
- OGC (Open GIS Consortium): a wireless-IP service that uses geographic information to serve a mobile user; any application service that exploits the position of a mobile terminal

Korean researchers (Haeock Choi, 2003; Heui-Chae Jin & Kilpyo Hong, 2003) define LBS as the advanced wireless services which provide useful information related to mobile user's location by tracking mobile devices like cell phone or PDA. According to the standard specification of Korea LBS Standardization Forum (2004), LBS is defined as the system or service which determines the location of person or object and utilizes the location information through the telecommunication network.

Simply knowing where you are, or how far you are from someone or something, is typically not valuable by itself. Relating location to other pertinent information gives it meaning and value. To derive this type of value, two types of elements are required: spatial data and tools to manipulate spatial data. GIS is central to both of these elements. LBS that incorporate GIS tools enable a wide range of spatial transactions that can be delivered in meaningful ways. Location application services are of universal industry service significance and depend upon the availability of relevant spatial information infrastructures

in forms useful for small devices (Koepfel, 2002).

## 2.4.2. Key Technology

According to the OGC's proposal, the key technologies of LBS are categorized into LDT (Location Determination Technology) for positioning a mobile user's location, LEP (Location Enabled Platform) for managing location data, and LAP (Location Application Program) for the actual services (Niedzwiadek, 2002; Ki-Joon Han, 2003). Sometimes LEP is expressed in "LBS Platform Technology," and LAP in "LBS Application Technology" (Seogyun Kim & Joonseok Lee, 2003).

LBS platform is a system composed of internal gateways and their components which perform a series of functionality to connect to the location system inside a mobile network; the support for LBS client and location based applications; and providing location information to external clients and applications (Korea LBS Standardization Forum, 2004). The common functionalities for the LBS platform can be summarized as follows. (National Computerization Agency of Korea, 2002; Korea LBS Standardization Forum, 2003)

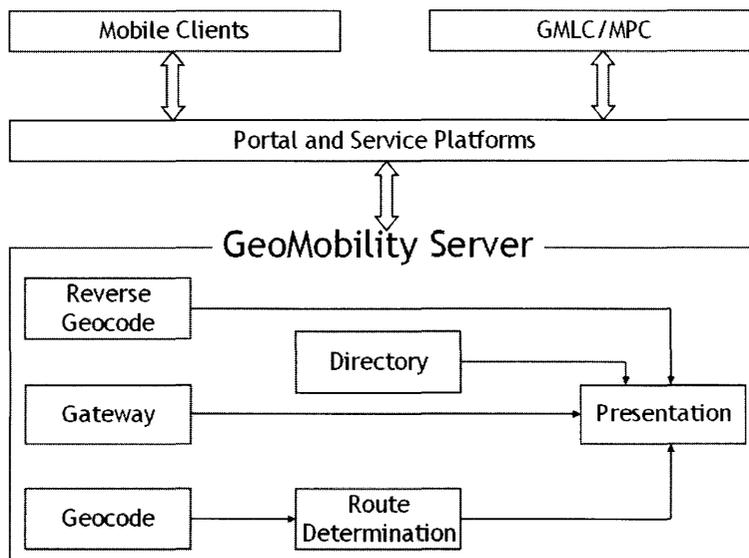
- Request and Response: managing the request of the location information of a mobile user and the response of the result through telecommunication network
- Location Managing: managing location information acquired from the location system of telecommunication network
- Location Based Functions: providing basic functionalities for the location services including pull<sup>46</sup> and push service<sup>47</sup>

---

<sup>46</sup> Pull service refers to the location service provided by user's request.

- Profile Management: protecting privacy and managing users' profile for the personalization<sup>48</sup>
- Authentication and Security: controlling access privilege to protect users' information and maintain system security
- Information Roaming between Carriers: providing standardized common API for the communication among telecommunication service providers

GeoMobility Server developed by OGC is the LBS platform comprising the Core Services developed under the OGC OpenLS (Open Location Services) initiatives<sup>49</sup>, and the architecture of OpenLS is described in [Figure 8]<sup>50</sup>.



[Figure 8] GeoMobility Server and OpenLS Architecture

<sup>47</sup> Push service refers to the location service spontaneously triggered from the mobile network to the user.

<sup>48</sup> Personalization is particularly useful to target marketing.

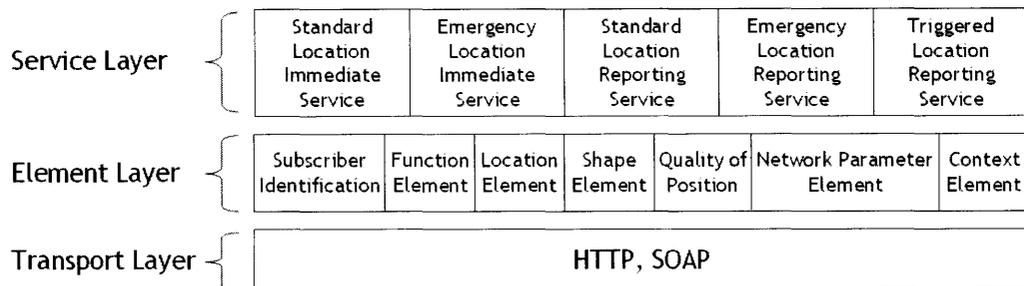
<sup>49</sup> As a functional area of OGC, the Open Location Services Initiative (OpenLS) is devoted to the development of interface specifications that facilitate the use of location and other forms of spatial information in the wireless Internet environment (<http://www.opengis.org/functional/?page=ols>).

<sup>50</sup> GMLC: Gateway Mobile Location Center, MPC: Mobile Positioning Center

The Core Services are composed of directory service; gateway service; location utility service; presentation service; route service; and navigation service (Open GIS Consortium, 2004b).

- Directory Service: A network-accessible service that provides access to an online directory (e.g., Yellow Pages) to find the location of a specific or nearest place, product or service.
- Gateway Service: A network-accessible service that fetches the position of a known mobile terminal from the network.
- Location Utility Service: Geocoder Service is a network-accessible service that transforms a description of a location, such as a place name, street address or postal code, into a normalized description of the location with a Point geometry. Reverse Geocoder Service is a network-accessible service that transforms a given position into a normalized description of a feature location (Address with Point), where the address may be defined as a street address, intersection address, place name or postal code.
- Presentation Service: A network-accessible service that portrays a map made up of a base map derived from any geospatial data and a set of ADT (Abstract Data Type) as overlays.
- Route Service: A network-accessible service that determines travel routes and navigation information between two or more points.
- Navigation Service: An enhanced version of the Route Service.

Korea LBS Standardization Forum plays the similar role to OGC OpenLS. [Figure 9] describes layer structure of the LBS platform suggested by Korea LBS Standardization Forum.



[Figure 9] Layer Structure of LBS Platform by Korea LBS Standardization Forum

In the transport layer which defines the transportation between mobile client and LBS platform, HTTP as a classical standard and SOAP as an emerging technology are adopted. In the element layer which defines necessary technology element for LBS platform, defines the programming interface for five standard services such as standard location immediate service, emergency location immediate service, standard location reporting service, emergency location reporting service, and triggered location reporting service. Moreover, several advanced services such as map service, yellow page, routing service, and traffic information can be added to the standard services. (Korea LBS Standardization Forum, 2004)

In Korea, the efforts for LBS standardization and LBS platform development are very important and urgent mission because LBS core technologies such as location determination and location platform are currently dependent on the advanced technology of foreign countries (Seogyun Kim & Joonseok Lee, 2003). Therefore, the standardization and development of open LBS platform that is independent of telecommunication network and software environment should be achieved in the near future (Ki-Joon Han, 2003; Heui-Chae Jin & Kilpyo Hong, 2003).

### 2.4.3. Major LBS Server Products

Currently, the major LBS server products include Microsoft MapPoint Location Server<sup>51</sup> (MLS), Autodesk LocationLogic<sup>52</sup>, and Intergraph IntelliWhere LocationServer<sup>53</sup>. The architecture of these products is similar to that of OpenLS, and they resemble one another in the functionality of LBS server. The comparison of the products in the aspects of application development and map server is described in [Table 3].

[Table 3] Programming Environment Comparison of Major LBS Server Products

Product	Application development	Map server/service
MapPoint Location Server	Visual Studio .NET	MapPoint Web Service
LocationLogic	Java 2 Micro Edition	(Internal)
IntelliWhere LocationServer	XML Parser	GeoMedia WebMap

Microsoft MapPoint Location Server works with Microsoft MapPoint Web Service (MWS). MapPoint Web Service is a web map service managed by Microsoft Corporation, which provides its own map data and does not allow application developers to add the map data they want<sup>54</sup>. It currently provides geographic street-level detail for 24 countries/regions: 21 countries/regions in Europe; 3 countries/regions in North America and South America<sup>55</sup>. Autodesk LocationLogic has an internal mapping engine for the Geo-Services. Intergraph IntelliWhere LocationServer provides the geospatial processing capabilities of Intergraph GeoMedia WebMap. As seen from the map server/service

<sup>51</sup> <http://www.microsoft.com/mappoint/mls>

<sup>52</sup> <http://locationservices.autodesk.com/platform>

<sup>53</sup> <http://imgs.intergraph.com/iwls>

<sup>54</sup> ESRI ArcWeb Services are the same case as MapPoint Web Service (<http://www.esri.com/software/arcwebservices/about/data.html>).

<sup>55</sup> <http://www.microsoft.com/mappoint/webservice/regional.msp>

compatibility, one of the most interesting things is that each vendor only supports its own map server/service.

#### 2.4.4. Service Contents

Currently in Korea, three telecommunication service providers, SK Telecom, KTF, and LG Telecom provide location services on the cellular network. Their major service contents are personal location tracking like buddy-finding, POI (Point of Interest)<sup>56</sup> like store-finding, traffic information, emergency service, etc. (Haeock Choi, 2003). Most services are provided on a text basis excluding sketch map service, and the buddy-finding is the most frequently used service for now (Korea Geospatial Information & Communication Company, 2003).

According to the LBS Industry Support Plan announced by the Ministry of Information and Communication (2003), LBS will be a “killer-application” for mobile service industries: this plan suggests LBS for public sector, corporations, and citizens as follows.

- LBS for public sector: emergency service
- LBS for corporations: m-commerce (advertising, reservation, online payment, etc.) and logistics (freight monitoring, stock management, etc.)
- LBS for citizens: day-to-day information (entertainment, tour information, transportation information, etc.)

The type of service contents according to the location accuracy is summarized in [Table 4] (Korea Information Strategy Development Institute,

---

<sup>56</sup> A location where one can find a place, product or service, typically identified by name rather than by address and characterized by type, which may be used as a reference point or a target in a location based service request (Open GIS Consortium, 2004)

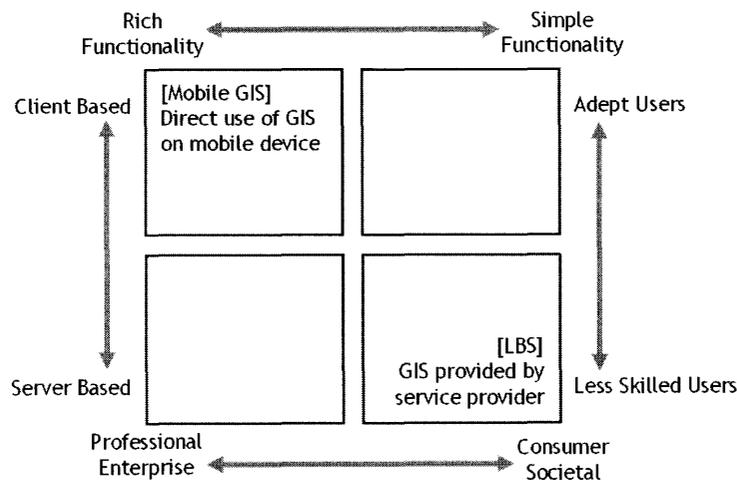
2001; Haeock Choi, 2002).

[Table 4] LBS Service Contents According to Location Accuracy

Location accuracy	Service contents
Up to 200 km	Weather forecasting
Up to 20 km	Local news, local traffic information
Up to 1 km	Traffic controlling, traffic jam monitoring
500 m - 1 km	Rural emergency service, service man management
75 m - 125 m	Urban emergency service, mobile advertising, tracking service
10 m - 50 m	Car navigation service, routing service

#### 2.4.5. Comparison of LBS to Mobile GIS

LBS and mobile GIS have a common aspect in that both of them utilize geographic information for mobile users. While LBS is a kind of information service based on a mobile user's location, mobile GIS is a kind of GIS running on mobile device. Though a dichotomy between LBS and mobile GIS is meaningless, comparison of LBS and mobile GIS can be illustrated in [Figure 10] (Eun-Hyung Kim & Jun-Gu Park, 2004).



[Figure 10] Comparison of LBS to Mobile GIS

The difference between mobile GIS and LBS lies in the purpose of each system. The purpose of mobile GIS is introducing the functionalities of desktop GIS and/or internet GIS to a mobile device, whereas that of LBS is utilizing location information of a mobile user. In case of mobile GIS, mapping is indispensable, but location-awareness is not; in case of LBS, location-awareness is indispensable, but mapping is not. However, the recent trends of them show that mobile GIS is adopting GPS functionality for the acquisition of accurate location information, and LBS is adopting mapping functionality for the efficient delivery of geographic information.

## 2.5. Software Engineering and Software Development

Software engineering methodology provides systematic structure and efficient throughput when developing a system. The classical object-oriented method is still useful for the mobile computing environment, and the modeling through UML (Unified Modeling Language) supports the object-oriented design and implementation. The component-based method is necessary for building a programming environment for application developers. Software component becomes more value-added if it interoperates with generic COTS (Commercial-Off-The-Shelf) products. In addition, mobile internet computing requires middleware technology for the brokerage between mobile client and server. .NET, as a new software development platform, provides such technologies as .NET Compact Framework and XML Web Services for more efficient mobile and internet software development.

### 2.5.1. Object-Oriented Software Engineering

OOSE (Object-Oriented Software Engineering) is based upon the concepts of encapsulation, inheritance, and polymorphism. Encapsulation packages data and the operations that manipulate the data into a single named object. Inheritance enables the attributes and operations of a class to be inherited by all subclasses and the objects that are instantiated from them. Polymorphism enables a number of different operations to have the same name, reducing the number of lines of code required to implement a system. (Wang, 2000) OOSE has attracted attention because of its promise for code reusability, maintainability, and extensibility in addition to the natural and intuitive language it promotes for discussion of software problems and their solutions (Taylor, 1990).

### 2.5.2. Unified Modeling Language

From the perspective of software engineering, modeling means the work of designing overall software structure before coding. A model plays an analogous role in software development and a role of examining the business logic is complete and correct. (Object Management Group, 2004) As an industry standard, UML (Unified Modeling Language) provides a good chance for “communication” and “documentation” both for software developers and end-users (Filev et al., 2003). UML has a set of graphical and textual modeling tools that aim to provide a common understandable language for developers and users. In particular, UML methodology provides a semantics document which aims to precisely describe the structure and meaning of the language. (Evans & Welling, 1999) As a tool of object-oriented analysis and design, UML represents

standardized meta-model. Below are the most commonly used diagrams (Object Management Group, 2003).

- Use case diagram: A use case diagram shows the relationship among use cases within a system or other semantic entity and their actors.
- Sequence diagram: A sequence diagram presents an interaction, which is a set of messages between objects.
- Class diagram: A class diagram is a graph of Classifier elements connected by their various static relationships. Note that a class diagram may also contain interfaces, packages, relationships, and even instances, such as objects and links. Perhaps a better name for it would be “static structural diagram” but “class diagram” is shorter and well established.

### 2.5.3. Component-Based Software Engineering

Component-based approaches for solving problems are common to many sciences or technical domains (Hofmann et al., 1999). A component is an integral logical constituent in a system, which is relatively independent of its surroundings concerning the functions and behaviors (Mei et al., 2001). It is an independently deployable implementation of some functionality, to be reused as-is in a broad spectrum of applications (Goulão & Abreu, 2002). A component is also defined as “a self-contained entity that exports functionality to its environment and may also import functionality from its environment using well-defined and open interfaces.” In this context an interface defines “the syntax and semantics of the functionality it comprises” and “components may support their integration into the surrounding environment by providing mechanisms, such as introspection or configuration functionality.” (Emmerich, 2002)

CBSE (Component-Based Software Engineering) intends to build large software systems by integrating pre-built software components. The high productivity is achieved by using standard components. The principles of CBSE can be best described by the following two guiding principles: reuse but do not reinvent; assemble pre-built components rather than coding line by line. (Wang, 2000) By enhancing the flexibility and maintainability of systems, CBSE approach can potentially be used to reduce software development costs, assemble systems rapidly, and reduce the spiraling maintenance burden associated with the support and upgrade of large systems (Carnegie Mellon Software Engineering Institute, 1997a).

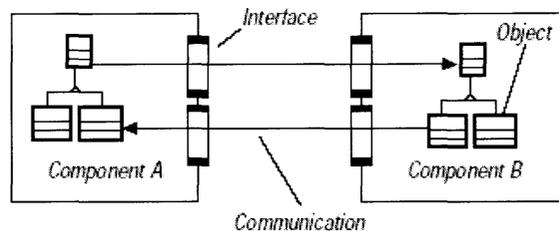
CBSE involves the technical steps for designing and implementing software components (component developers' view) and assembling systems from pre-built components (application developers' view) (Goulão & Abreu, 2002).

- Component developers' view: when designing reusable components, one has to take into account what type of usage the components are expected to support.
- Application developers' view: from an integrator's point of view, a component is a unit of composition of software architectures.

#### 2.5.4. COTS Interoperability

The term COTS (Commercial-Off-The-Shelf) means a software product, supplied by a vendor, that has specific functionality as part of a system, a piece of pre-built software that is integrated into the system and must be delivered with the system to provide operational functionality or to sustain maintenance efforts. (Morisio et al., 2002)

One of the important aspects for developing a new software component is the communication with generic COTS products. This kind of communication is considered interoperability (Tu et al., 2002), and [Figure 11] shows the COTS interoperability through an interface (Hofmann et al., 1999). In case of LBS and mobile mapping, COTS map server may be necessary for map representation and geographic information delivery.



[Figure 11] Conceptual Structure of COTS Interoperability

## 2.5.5. Middleware Technology

In the computer industry, middleware is a general term for any programming that serves to “glue together” or mediate between two separate and often already existing programs<sup>57</sup>. In fact, middleware for mobile computing is also a kind of component to interconnect the components that form a mobile application (Meier, 2002), or to glue a server and mobile clients together.

In fact, the middleware has its ground on event-based communication. In the event-based communication, a mobile client issues an event to a server in the form of request, and then the server returns a result to the mobile client in the form of response. Even-based communication through a middleware is well

<sup>57</sup> <http://whatis.com>

suitable to addressing the requirements of wireless mobile computing applications (Hughes & Cahill, 2003). It avoids centralized control as well as long-lasting and hence potentially expensive connections and requires a less tightly coupled communication relationship between application components compared to the traditional client/server communication model. Event-based communication through a middleware interconnects the components that comprise an application in a potentially distributed and heterogeneous environment, and has recently become widely used in application areas such as large-scale internet services and mobile programming environments. (Meier, 2002)

## 2.5.6. .NET-based Technology

### 2.5.6.1. .NET and Its Advantages

As Bill Gates put it at the launch of Visual Studio.NET on February 13, 2002 “The .NET vision incorporates more than just web services. It talks about how people use these things, being able to get their information at any time, any place, on any device, pocket-sized devices, tablet-sized devices, in the car, or the TV set, you name it, all connected up to these capabilities.” (Wigley, 2003)

Visual Studio .NET 2003 is the comprehensive, multi-language development tool for rapidly building and integrating XML Web Services and applications<sup>58</sup>.

Being different from the previous versions (Visual Studio 6 and Visual Studio .NET 2002), it has enhanced support for the mobile device developer including<sup>59</sup>:

---

<sup>58</sup> <http://msdn.microsoft.com/vstudio/productinfo/faq/default.aspx>

<sup>59</sup> <http://msdn.microsoft.com/vstudio/productinfo/overview/whatsnew.aspx>

- Building applications for smart devices such as Pocket PC, Pocket PC Phone Edition, and Smartphone
- Building applications for more than 200 mobile web devices including WAP (Wireless Application Protocol) phones, PDA's, and pagers

Visual Studio .NET is based on the .NET Framework as a development platform, and SDE (Smart Device Extension) for Visual Studio .NET provides a robust development environment for creating mobile applications that target the .NET Compact Framework (Roof & Fergus, 2003). The .NET Compact Framework is the smart device development platform for realizing the goal of accessibility any time, any place, and on any device. It is a rich subset of the .NET Framework, thereby providing the same benefits as the .NET Framework. The .NET Compact Framework significantly simplifies the process of creating and deploying applications to mobile devices while also allowing the developer to take full advantage of the capabilities of the device<sup>60</sup>. The major advantages of the .NET Compact Framework include;<sup>61</sup>

- Familiar programming model for desktop developers: The .NET Compact Framework uses the same programming model as the desktop .NET Framework. This familiarity makes it easier for developers to write new applications for the .NET Compact Framework and makes it much easier for them to migrate portions of their existing .NET Framework applications to smart devices.
- Shared code and increased efficiency: Because the .NET Compact

---

<sup>60</sup> <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/NETCompactFrame.asp>

<sup>61</sup> <http://msdn.microsoft.com/mobility/prodtechinfo/devtools/netcf/overview/default.aspx>

Framework delivers the same programming model across a range of devices, it simplifies the process of developing an application that will run on multiple devices.

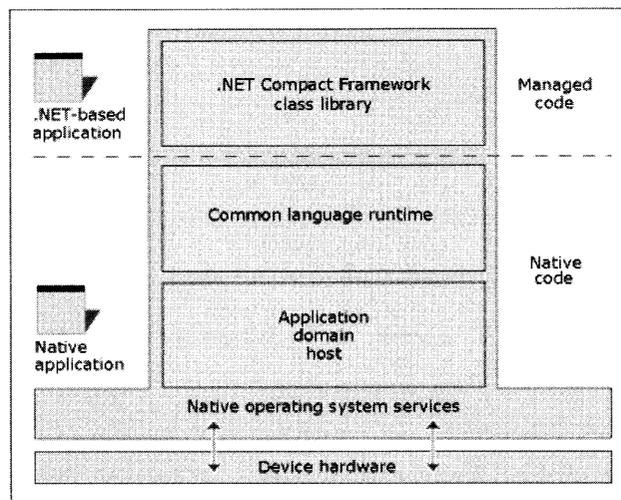
- Designed from the ground up for XML Web Services: XML Web Services are a very useful application model for smart devices. Networked devices need to communicate with a wide variety of other systems, and the standardized XML Web Services protocols allow applications to communicate with each other, regardless of operating system or programming language. As with the desktop .NET Framework, the .NET Compact Framework is designed from the ground up to be the best development platform for writing and consuming XML Web Services.

#### 2.5.6.2. Comparison of .NET to eVTs

eVTs (eMbedded Visual Tools) is a stand-alone development environment that includes eVB (eMbedded Visual Basic) and eVC (eMbedded Visual C++). eVB provides quick development time, but applications do not perform as well as those built with eVC or the .NET Compact Framework. eVC provides the highest performance particularly for the applications that need direct control over device hardware and operating system services. (Wigley, 2003)

The .NET Compact Framework significantly enhances the development capabilities of the current Microsoft eVTs (eMbedded Visual Tools). One of the goals for the .NET Compact Framework is to create a development environment as easy to use as eVB and as powerful as eVC so that you get the best of both tools. (Milroy, 2003) Visual Studio .NET 2003 with the .NET Compact Framework

is easy and powerful as well because it provides RAD (Rapid Application Development) environment through the controls and class libraries and P/Invoke (Platform Invocation Services) for interoperating with native code<sup>62</sup> [Figure 12].



[Figure 12] Platform Architecture of .NET Compact Framework

The existing application developers using Visual Basic, Visual C++, or Java can easily migrate to Visual Studio .NET with the .NET Compact Framework (Cornelius, 2002) because it adopts Visual Basic .NET and Visual C# .NET as a programming language based on the CLR (Common Language Runtime)<sup>63</sup>. The

<sup>62</sup> [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dv\\_evtuv/html/etconplatformarchitecture.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dv_evtuv/html/etconplatformarchitecture.asp)

<sup>63</sup> As a part of Microsoft .NET Framework, the Common Language Runtime (CLR) is programming that manages the execution of programs written in any of several supported languages, allowing them to share common object-oriented classes written in any of the languages. Microsoft refers to its CLR as a “managed execution environment.” A program compiled for the CLR does not need a language-specific execution environment. (<http://whatis.com>)

CLR consists of execution engine and class libraries which are shared by different programming languages (Wigley, 2003), thereby providing the same performance to the applications composed in different programming languages. In addition, compared with older device development language like eMbedded Visual Tools, .NET has a large pool of qualified developers (Yuan, 2003).

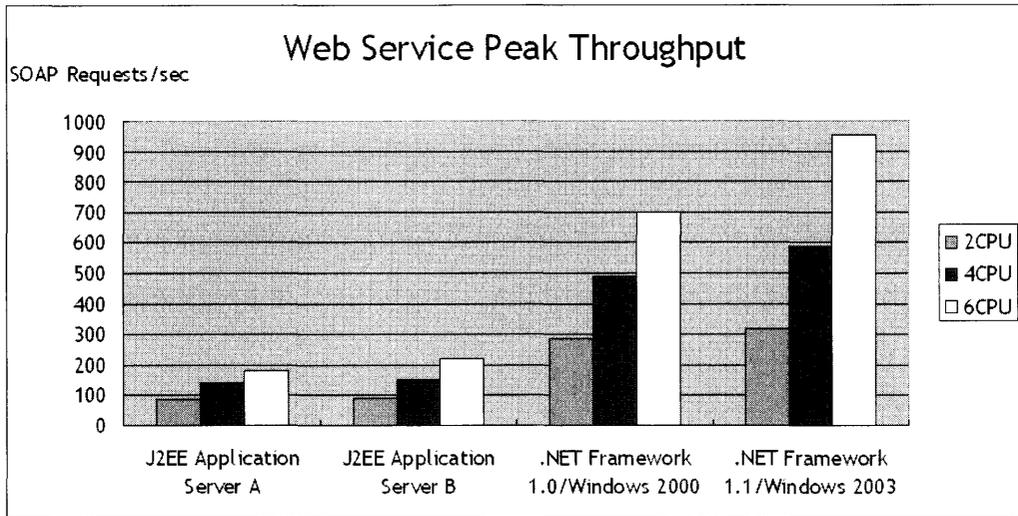
### 2.5.6.3. Comparison of .NET to Java

J2ME (Java 2 Micro Edition) is a Java platform for mobile computing. Despite the Java's advantage of supporting multiple operating systems, Visual Studio .NET with the .NET Compact Framework shows the best performance on the Windows CE-based PDA (Chappell, 2002; Stanski & Wiharto, 2003) which is our target device.

Particularly for the middleware technology, .NET-based XML Web Services show higher performance than Java-based web application. The benchmark conducted on the .NET Framework and J2EE (Java 2 Enterprise Edition) has found out the .NET Framework significantly outperform J2EE, as shown in [Figure 13]<sup>64</sup> (Edwards, 2002; The Middleware Company, 2003).

---

<sup>64</sup> <http://msdn.microsoft.com/netframework/technologyinfo/overview/default.aspx>



[Figure 13] Comparison of Web Service Peak Throughput

#### 2.5.6.4. XML Web Services

XML Web Services are the fundamental building blocks in the move to distributed computing on the internet. Open standards and the focus on communication and collaboration among people and applications have created an environment where XML Web Services are becoming the platform for application integration. Applications are constructed using multiple XML Web Services from various sources that work together regardless of where they reside or how they were implemented. The common features of XML Web Services are as follows. (Wolter, 2001)

- XML Web Services expose useful functionality to web users through a standard web protocol such as HTTP and SOAP.
- XML Web Services provide a way to describe their interfaces in detail to allow a user to build a client application. This description is usually provided in an XML document called WSDL (Web Services Description

Language)<sup>65</sup> document.

- XML Web Services are registered so that potential users can find them easily by UDDI (Universal Discovery Description and Integration)<sup>66</sup>.

XML Web Services support HTTP/GET, HTTP/POST, and SOAP for the request/response on the internet, and the characteristics of each method are as follows (Goode et al., 2002).

- HTTP/GET: Simple, unstructured information is bundled in with the page as a sequence of name-value pairs. These pairings are a simple way to combine all the values into a single string.
- HTTP/POST: While HTTP/GET uses the end of the URL to pass its information from resource, HTTP/POST uses the body of the transmission to carry the same name-value pairs. Using POST means a less-cluttered URL, as well as slightly tighter security.
- SOAP: It provides an effective way to call all sorts of remote functions. It wraps up any call-specific information inside an XML element.

---

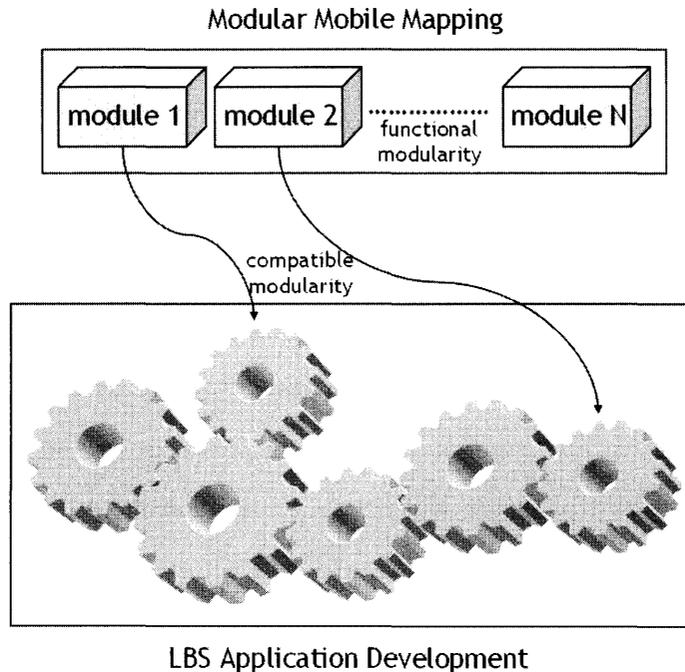
<sup>65</sup> WSDL is an XML-based language used to describe the services a business offers and to provide a way for individuals and other businesses to access those services electronically (<http://whatis.com>).

<sup>66</sup> UDDI is an XML-based registry for businesses worldwide to list themselves by name, product, location, or the web services they offer. Its ultimate goal is to streamline online transactions by enabling companies to find one another on the web and make their systems interoperable. (<http://whatis.com>)

### 3. Framework Design of Geocomputing Platform

#### 3.1. Overview

As mentioned in “Research Methods” in chapter 1, the framework of our geocomputing platform is based on “modular mobile mapping” and “LBS support.” Necessary functionalities for LBS application development such as location-awareness, wireless map service, and geographic information delivery will be modularized in the geocomputing platform: for the modularity allows convenient access to a specific functional component in the application development process and diverse compatibility with external entities such as map server and database system [Figure 14].



[Figure 14] Modular Mobile Mapping and LBS Support of the Geocomputing Platform

Our geocomputing platform primarily targets LBS and focuses on the mobile mapping with location-awareness which is at once a necessary module for LBS and an independent application as it is. The information provided by LBS can be more value-added when coupled with geographic information by the mobile mapping which provides a map related to mobile user's location or interest area.

The mobile mapping with location-awareness serves for the mobility and portability of mobile users to acquire geographic information with real-time location. We suppose two typical cases for which the mobile mapping with location-awareness is necessary: the case of field data collection in a small area of countryside and the case of map service covering the whole city.

In the first case, if a soil surveyor collects soil samples in a small area of countryside, he needs to prepare his own maps according to the particular area to carry with him in a mobile device. And if the countryside is out of the wireless network, he has to work with the embedded map for information source and the GPS coordinate for location-awareness in the stand-alone computing environment.

In the second case, if a citizen searches cadastral or housing information in a city, he/she needs to request a map according to his area of interest. The land parcel map should be stored in a remote database server because it should be in huge volume to process the ad hoc request of anonymous mobile users. He can work with the map provided by a database server for information source and the GPS coordinate for location-awareness in the client/server computing environment.

These two cases provide a clue to the categorization of the mobile mapping environment. In a conventional way, the first case corresponds to stand-alone system and the second case to client/server system. Stand-alone system means that a program independently runs on mobile device without network connection of remote server which provides necessary data or application service. Client/server system means that a program runs on mobile device with network connection of remote server which provides necessary data or application service.

In this study, we adopt the terms “mobile embedded mapping” and “mobile internet mapping” based on the way of providing geographic information and the connection of wireless network. Mobile embedded mapping is for the stand-alone system in which geographic information is provided from the mobile device itself without wireless internet connection, and mobile internet mapping is for the client/server system in which geographic information is provided from the remote database server with wireless internet connection. [Table 5] summarizes our two types of mobile mapping environment.

[Table 5] Two Types of Mobile Mapping Environment

Category	Key feature
Mobile Embedded Mapping	<ul style="list-style-type: none"> <li>. Stand-alone system</li> <li>. Geographic information from a mobile device itself</li> <li>. Without wireless internet connection</li> </ul>
Mobile Internet Mapping	<ul style="list-style-type: none"> <li>. Client/server system</li> <li>. Geographic information from a remote database server</li> <li>. With wireless internet connection</li> </ul>

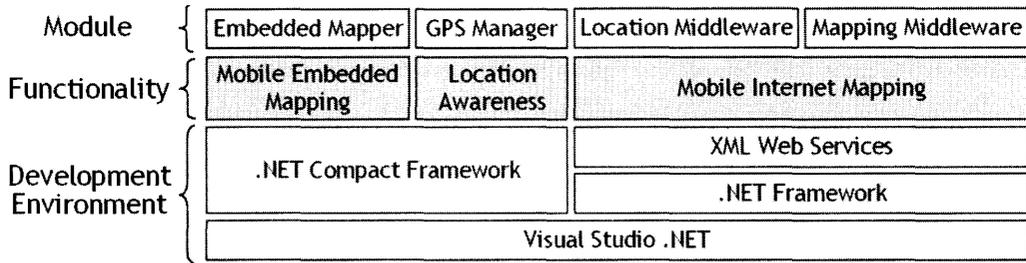
Our geocomputing platform will cover both mobile mapping environments: mobile embedded and mobile internet mapping. Particularly, mobile internet

mapping plays an important role in LBS through the capability of providing a wide range of geographic and attribute data from a remote database server. Though our geocomputing platform does not cover all the requirements of OGC OpenLS or Korea LBS Standardization Forum, it provides the core parts of LBS platform such as GPS functionality, location middleware, and mapping middleware based on the modularized functionalities of mobile mapping.

In the conceptual dimension, this geocomputing platform is composed of three main functionalities of ( i ) location-awareness, ( ii ) mobile embedded mapping, and ( iii ) mobile internet mapping. In the physical dimension, the location-awareness is implemented as the module of GPS Manager, and the mobile embedded mapping as Embedded Mapper. These two modules are built in the form of DLL (Dynamic Link Library)<sup>67</sup> which is one of the most common deployment formats for Windows application development including mobile environment. In addition, the mobile internet mapping is implemented as the module of Location Middleware and Mapping Middleware. These two modules are built in the form of XML Web Services which provide very high performance for internet application development including mobile environment. For the module of GPS Manager and Embedded Mapper, .NET Compact Framework will be used; for the module of Location Middleware and Mapping Middleware, XML Web Services under .NET Framework will be used. [Figure 15] describes the overall structure of the geocomputing platform.

---

<sup>67</sup> DLL is a collection of small programs, any of which can be called when needed by a larger program that is running in Windows. It is usually referred to as a DLL file. (<http://whatis.com>)

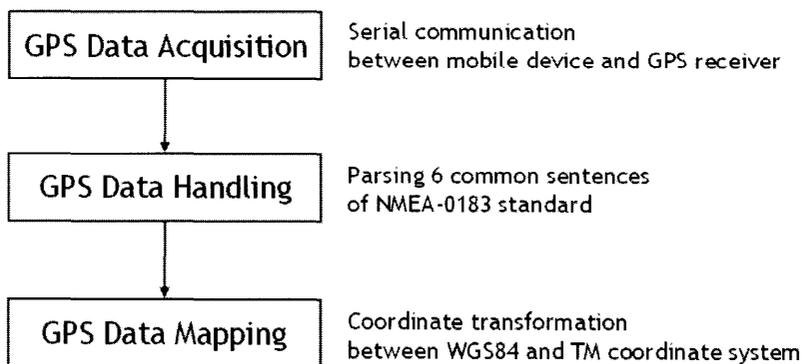


[Figure 15] Overall Structure of the Geocomputing Platform

## 3.2. Module Composition

### 3.2.1. Location-Awareness

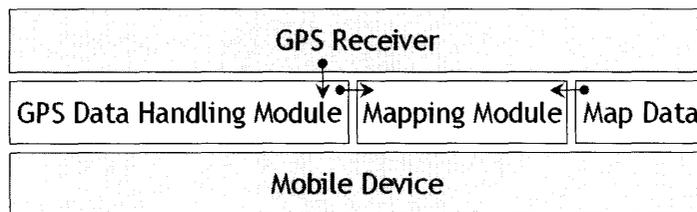
The basic workflow of the location-awareness which will be combined with the mobile mapping in the embedded and internet environment is described in [Figure 16]. This procedure is composed of GPS data acquisition on the mobile device, GPS data handling for NMEA-0183 sentences, and GPS data mapping through coordinate transformation.



[Figure 16] Workflow of GPS Data Management

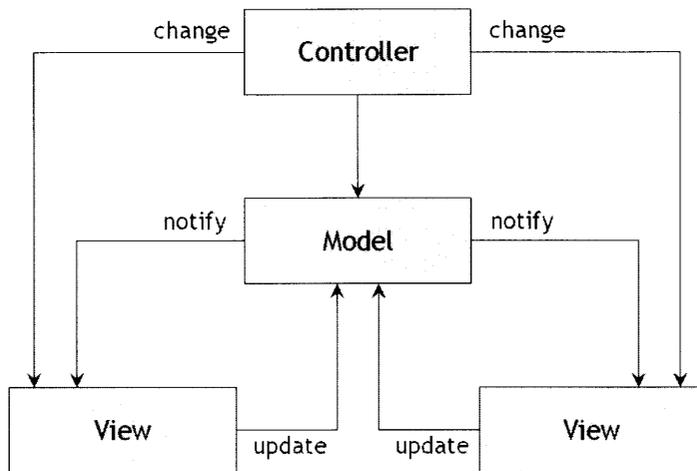
### 3.2.2. Mobile Embedded Mapping

Our architecture for the mobile mapping in the embedded environment is composed of GPS Data Handling Module, Mapping Module, and Map Data which are interconnected between mobile device and GPS receiver [Figure 17]. GPS Data Handling Module gets data from GPS receiver, and Mapping Module visualizes the GPS coordinate on a map using the Map Data embedded in the mobile device. As a stand-alone system, the overall structure may be similar to that of major vendors' mobile GIS products such as ESRI ArcPad, MapInfo MapX Mobile, and Pocket Systems PocketGIS.



[Figure 17] Architecture of Mobile Embedded Mapping in the Geocomputing Platform

This Mapping Module has an MVC (Model-View-Controller) architecture whose goal is to separate the application object (model) from the way it is represented to the user (view) from the way in which the user controls it (controller), thereby allowing flexibility and reusability. The role of model, view, and controller is addressed in [Table 6], and the relationship between model, view, and controller objects is shown in [Figure 18]. (Wheeler, 1996)



[Figure 18] Architecture of MVC (Model-View-Controller)

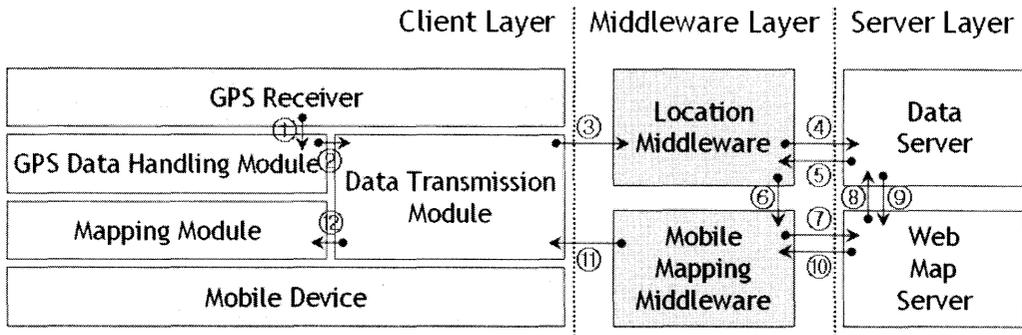
[Table 6] Role of Model, View, and Controller in Graphics Programming

Model	The model object knows about all the data that need to be displayed. It also knows about all the operations that can be applied to transform that object. However, it knows nothing whatever about the GUI, the manner in which the data are to be displayed, nor the GUI actions that are used to manipulate the data. The data are accessed and manipulated through methods that are independent of the GUI.
View	The view object refers to the model. It uses the query methods of the model to obtain data from the model and then displays the information.
Controller	The controller object knows about the physical means by which users manipulate data within the model. In a GUI for example, the controller object would receive mouse clicks or keyboard input which it would translate into the manipulator method which the model understands.

### 3.2.3. Mobile Internet Mapping

Our architecture for the mobile mapping in the internet environment is composed of client, middleware, and server layer [Figure 19]. The core part is middleware layer which acts as a broker between client and server layer. If a mobile client issues a request for a map according to GPS location, the Location Middleware works for processing location data, and the Mobile Mapping

Middleware works for providing a map according to the location. More specific workflow of our mobile internet mapping architecture is as [Table 7].



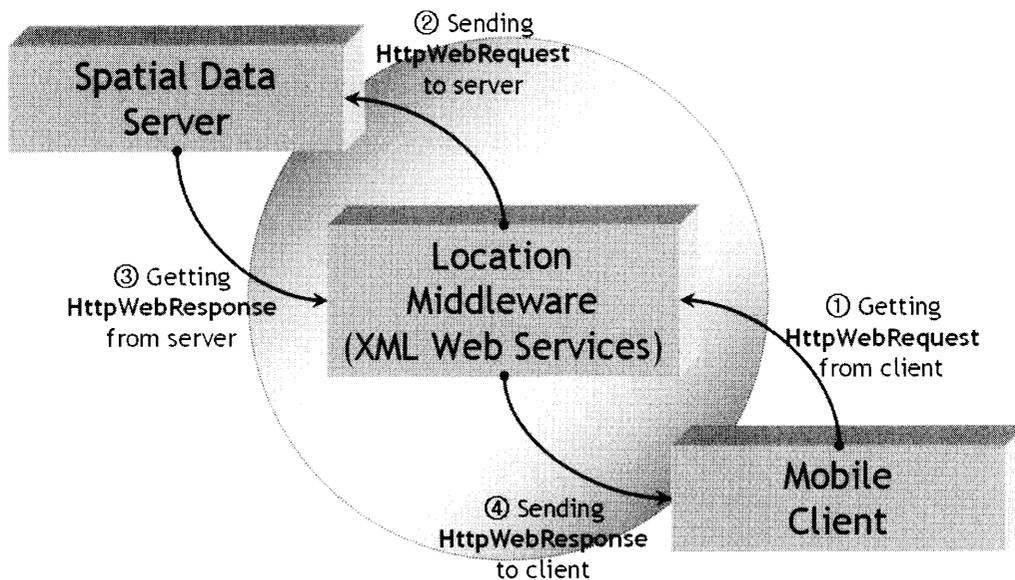
[Figure 19] Architecture of Mobile Internet Mapping in the Geocomputing Platform

[Table 7] Workflow of Mobile Internet Mapping

Process	Description
1	Getting GPS data from a GPS receiver and performing the coordinate transformation
2, 3	Sending {x, y} coordinate to Location Middleware through Data Transmission Module
4, 5	Getting map index according to the coordinate from Data Server
6, 7	Issuing a request to Map Server for the map service according to the coordinate
8, 9, 10	Getting the map service information from Web Map Server
11	Sending the map service information to a client through Data Transmission Module
12	Parsing the map service information and displaying an image map through Mapping Module

### 3.2.3.1. Location Middleware

One of the most important roles of our Location Middleware is the functionality of geocoding and reverse geocoding [Figure 20]. Geocoding is finding a representative {x, y} coordinate according to a specific address, and reverse-geocoding is finding an address according to a specific {x, y} coordinate (Niedzwiadek, 2002).

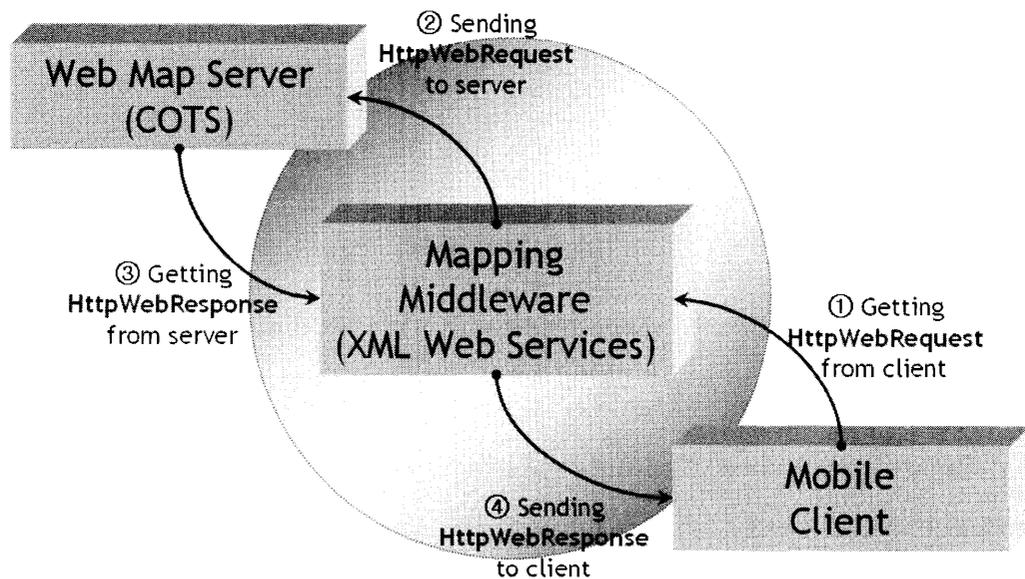


[Figure 20] Workflow of Location Middleware in the Geocomputing Platform

The skeleton process of our location middleware is ( i ) getting a request from Mobile Client and ( ii ) sending it to Spatial Data Server, and then, ( iii ) getting a response according to the request and ( iv ) sending it back to the Mobile Client. In case of geocoding, this process will be applied like: ( i ) getting an address as a request from Mobile Client and ( ii ) sending it to Spatial Data Server, and then, ( iii ) getting a polygon centroid of the address as a response and ( iv ) sending it back to the Mobile Client. In case of reverse-geocoding, this process will be applied like: ( i ) getting a GPS coordinate as a request from Mobile Client and ( ii ) sending it to Spatial Data Server, and then, ( iii ) getting an area unit according to the GPS location as a response and ( iv ) sending it back to Mobile Client.

### 3.2.3.2. Mapping Middleware

The skeleton process of our mapping middleware is ( i ) getting a request from Mobile Client and ( ii ) sending it to the Web Map Server, and then, ( iii ) getting a user-requested map as a response and ( iv ) sending it back to the Mobile Client [Figure 21].



[Figure 21] Workflow of Mapping Middleware in the Geocomputing Platform

One of the most important functionalities of our Mobile Mapping Middleware is the interoperability with COTS map servers. This interoperability provides application developers with the option for map server which may have different strength on the various functionalities (Tu et al., 2002). Although the interoperability with generic map servers is important, even the major LBS server products such as Microsoft MLS, Autodesk LocationLogic, and Intergraph IntelliWhere do not provide map server interoperability, just adopting their own

map server.

Therefore, we selected ESRI ArcIMS, UMN MapServer, and BBN OpenMap to provide interoperability with generic map servers. These map servers are considered to be leaders in the aspect of practical use and operation environment. The ArcIMS is a leading commercial map server, the MapServer is an open-source map server using CGI (Common Gate Interface)<sup>68</sup>, and the OpenMap is an open-source map server under the Java HTTP.

### 3.3. Development Environment

For the hardware composition, we adopt Windows CE-based PDA and serial type GPS receiver because they reflect current trend and standard. For the development tool, we will use Visual Studio .NET as the most prospecting development tool which deals with embedded environment through the .NET Compact Framework and internet environment through the XML Web Services.

#### 3.3.1. Mobile Device and GPS Receiver

It should be better way to work with PDA for now because cell phone has small screen for mobile mapping. Currently, the major types of PDA are Pocket PC series that adopts Windows CE-based operating system and Palm series that adopts Palm OS.

Windows CE-based operating system is an open, scalable, 32-bit operating system that is designed to meet the needs of a broad range of intelligent

---

<sup>68</sup> The common gateway interface (CGI) is a standard way for a Web server to pass a Web user's request to an application program and to receive data back to forward to the user (<http://whatis.com>).

devices<sup>69</sup>. According to IDC report on the 2003 Western European handheld device market, Windows CE-based operating system is growing faster and faster [Table 8]<sup>70</sup>. It's because Windows CE series is a compact edition of the Windows which brings the familiarity of Windows to mobile device users. In this study, we select HP iPAQ h5550<sup>71</sup> for PDA which adopts Windows CE-based operating system. Its specification is as [Table 9].

[Table 8] Western European Handheld Device Shares

Operating System	4Q, 2003	% Share	4Q, 2002	% Share	Growth
Windows CE-based OS	570,670	59.6%	222,835	32.2%	156.1%
Palm OS	372,990	38.9%	443,920	64.1%	-16.0%
Others	14,390	1.5%	26,300	3.8%	-45.3%
Total	958,050	100.0%	693,055	100.0%	38.2%

[Table 9] Specification of HP iPAQ h5550

Processor	400 MHz Intel XScale processor
Operating System	Windows Mobile 2003
Memory	48 MB Flash ROM, 128 MB SDRAM
Display	240 x 320 pixel resolution

In addition, GPS receiver is necessary for the location-awareness. Serial communication has been a standard for data transmission between GPS receiver and external device, and GPS receivers for PDA based on serial communication are being released by vendors these days. We can connect PDA and GPS receiver as shown in [Figure 22]. In this study, we select Navman GPS 3450<sup>72</sup>, and its specification is as [Table 10].

<sup>69</sup> <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wceintro/html/wceintroAboutWindowsCE.asp>

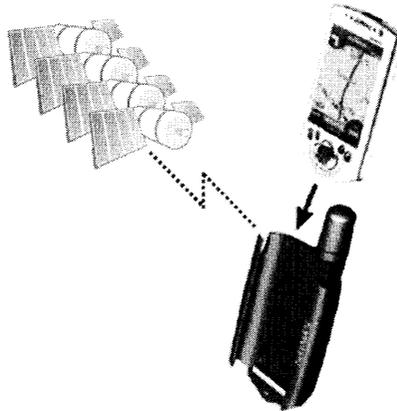
<sup>70</sup> <http://www.windowfordevices.com/news/NS2827737955.html>

<sup>71</sup> <http://h10010.www1.hp.com/wwpc/us/en/sm/WF06a/215348-64929-215381-314903-f44-322916.html>

<sup>72</sup> <http://www.navman.com/go/usa/products/gps3450>

[Table 10] Specification of Navman GPS 3450

Baud Rate	57600
Data Bits	8
Parity	None
Stop Bits	1
Update Rate	Once every two seconds
Output Message	NMEA-0183 GPGGA and GPRMC, 2s / GPGSA and GPSSV, 4s



[Figure 22] Connection between PDA and GPS Receiver

### 3.3.2. Development Tool

In this study, development tool is considered in two aspects: platform development vs. application development. For the development of our geocomputing platform, Visual Studio .NET is chosen: .NET Compact Framework is used for the module of location-awareness and mobile embedded mapping; XML Web Services are used for the module of location middleware and mapping middleware. For the application development of LBS and mobile mapping, our geocomputing platform can be used within Visual Studio .NET and Java environment<sup>73</sup>.

<sup>73</sup> This geocomputing platform allows Java mobile client for XML Web Services in case of the

## 4. Implementation of Geocomputing Platform

The basic strategy for developing the geocomputing platform is “independency from external libraries” and “providing .NET-based programming interface.” We composed all the source code for the necessary functionalities without importing external libraries, and built a .NET-based component library which provides the functionalities for application developers. For convenience, we call this geocomputing platform “CLAP.NET” abbreviated from “Component for Location-based Application on PDA with .NET.”

### 4.1. Location-Awareness

Though the GPS enabling technology is now being generalized, we implemented GPS functionality using the communication between Windows CE-based PDA and serial type GPS receiver under the .NET Compact Framework. It is because the integration of GPS functionality with mobile mapping in a unified .NET environment becomes an important element for our geocomputing platform.

The integration of GPS functionality with mobile mapping is a process composed of ( i ) serial communication, ( ii ) GPS data handling, and ( iii ) coordinate transformation. The first two functionalities are necessary in any case, whereas the third one is necessary for the countries which do not adopt WGS84 as their coordinate system of national base map. For the serial communication and GPS data handling, we implemented two base classes, from which an application code inherits: one class for communicating through a

---

mobile internet mapping through HTTP.

serial port, and the other class for handling NMEA-0183 sentences coming from a serial port. The first base class, named “ClapComm,” opens and configures a serial port; sends and receives bytes of data; interacts with the control input and output. The second base class, named “ClapGPS,” parses the 6 common sentences of NMEA-0183, and extracts necessary information from the sentences. In an application program, the “ClapGPS” class instantiates a “ClapComm” class object and uses it. In addition, we implemented necessary methods and properties in “ClapCoord” class for the transformation between WGS84 and TM coordinate system.

#### 4.1.1. Serial Communication

Under the .NET Compact Framework, P/Invoke (Platform Invocation Services) makes it possible to manipulate serial port through the interaction with “coredll.dll.” Coredll.dll is a DLL (Dynamic Link Library) provided in the Windows CE-based operating system, which has the same role as Win32 API functions of the Windows 95/98/NT/2000/XP. P/Invoke enables managed code in the CLR (Common Language Runtime) to make calls into unmanaged DLL including Win32 API which has the necessary functions for serial communication (Hind, 2002; Wigley, 2002).

Just as in the .NET Framework, using the P/Invoke in the .NET Compact Framework includes three primary steps: declaration, invocation, and error handling.<sup>74</sup>

- Declaration: To begin, you must, at design time, tell the .NET Compact

---

<sup>74</sup> <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetcomp/html/netcfintrointerp.asp>

Framework which unmanaged function you intend to call. You'll need to include the DLL name (also called the module), the function name (also called the entry point) and the calling convention to use.

- **Invocation:** Once you've correctly declared the function to call (typically in a utility class), you can wrap the function call in a method of that class.
- **Handling Errors:** Although developers never expect their code to generate runtime errors, it is important to remember that functions invoked on DLLs using P/Invoke can generate two different kinds of errors. The first is an exception generated by the P/Invoke service itself. This occurs if the arguments passed to the method contain invalid data, or if the function itself is declared with improper arguments. The second type of error that might be produced is an error returned from the DLL function itself.

The communication with serial port using P/Invoke was implemented by declaring each method according to that of "coredll.dll" and invoking the method when needed. We wrapped necessary API functions of "coredll.dll" in our CLR managed class of "ClapComm" for the functionalities of 4.1.1.1 to 4.1.1.4.

#### 4.1.1.1. Opening Serial Port

Because hardware vendors and device driver developers can give any name to a port, an application should list the available ports and enable users to specify the port to open. We implemented "CreateFile" method for opening serial port

[Code 1], considering the tips as follows.<sup>75</sup>

- Insert a colon after the communication port pointed to with the first parameter, lpFileName. For example, specify “COM1:” as the communication port.
- Specify zero in the dwShareMode parameter unless you are using one of the supported drivers that allows multiple opens.
- Specify OPEN\_EXISTING in the dwCreationDisposition parameter. This flag is required.
- Specify zero in the dwFlagsAndAttributes parameter. Windows CE supports only nonoverlapped I/O.

[Code 1] Declaration of CreateFile Method for Opening Serial Port

```
[DllImport (“coredll.dll”)]
private static extern int CreateFile(
    string lpFileName,           // Port name
    int dwDesiredAccess,        // Read/write mode
    int dwShareMode,            // Share mode
    int lpSecurityAttributes,    // Security attributes
    int dwCreationDisposition,  // How to open the serial port
    int dwFlagsAndAttributes,   // Port attributes
    int hTemplateFile);         // Handle to port with attribute to copy
```

#### 4.1.1.2. Configuring Serial Port

The most critical phase in serial communication programming is configuring the port settings with the DCB (Device Control Block) structure which defines the control setting for a serial communications device. A call to the “CreateFile” method opens a serial port with default port settings. By way of precaution against the case an application needs to change the defaults, we should

---

<sup>75</sup> <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wceseril/html/cmconOpeningPort.asp>

prepare the “GetCommState” method to retrieve the default settings and the “SetCommState” method to set new port settings.<sup>76</sup>

Frequently used port settings, which can be adjusted, are baud rate, RTS (Request To Send) control, DTR (Data Terminal Ready) control, parity, and stop bits. In the “ClapComm” class, a constructor method for initializing these settings and each “SetOOO” method for adjusting these settings were implemented. In addition to the port setting with DCB, time-out settings to escape from unnecessary read/write operation were also prepared.

#### 4.1.1.3. Writing to Serial Port

We implemented “WriteFile” method which transfers data through the serial connection to another device [Code 2]. Before calling this method, a serial port must be opened and configured. Because Windows CE does not support overlapped I/O, the primary thread<sup>77</sup> or any thread that creates a window should not try to write a large amount of data to a serial port. Below are tips for using this method.<sup>78</sup>

- Pass the port handle to the WriteFile method in the hFile parameter.

The CreateFile method returns this handle when an application opens a port.

---

<sup>76</sup> <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wceseril/html/cmconconfiguringserialport.asp>

<sup>77</sup> Operating systems use processes to separate the different applications that they are executing. Threads are the basic unit to which an operating system allocates processor time, and more than one thread can be executing code inside that process. (<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconthreadsthreading.asp>)

<sup>78</sup> <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wceseril/html/cmconwritingtoserialport.asp>

- Specify a pointer to the data to be written in lpBuffer. Often this data is binary data or a character array.
- Specify the number of bytes to write in nNumberOfBytesToWrite. The entire buffer can be passed to the driver.
- Specify in lpNumberOfBytesWritten a pointer to a location where WriteFile will store the number of bytes actually written. You can then look at this location to determine if the data transferred.
- Be sure that lpOverlapped is NULL.

[Code 2] Declaration of WriteFile Method for Writing to Serial Port

```
[DllImport("coredll.dll")]
private static extern bool WriteFile(
    int hFile, // Port handle
    byte[] lpBuffer, // Data to write
    int nNumberOfBytesToWrite, // Number of bytes to write
    ref int lpNumberOfBytesWritten, // Pointer to number of byte written
    int lpOverlapped); // overlapped I/O
```

#### 4.1.1.4. Reading from Serial Port

We implemented “ReadFile” method to receive data from a device at the other end of a serial connection [Code 3]. “ReadFile” takes the same parameters as the “WriteFile” method. Typically, a read operation is a separate thread that is always ready to process data arriving at a serial port. A communication event signals the read thread, and there is data to read at a serial port. Then the read thread waits for another communication event. Below are tips for using this method.<sup>79</sup>

- Pass the port handle to ReadFile in the hFile parameter. The CreateFile

<sup>79</sup> <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wceseril/html/cmconreadingfromserialport.asp>

method returns this handle when an application opens a port.

- Specify a pointer to receive the data that is read in lpBuffer.
- Specify the number of characters to read in nNumberOfBytesToRead.
- Specify a pointer to a location where ReadFile will store the number of bytes actually read in lpNumberOfBytesRead.
- Be sure that lpOverlapped is NULL. Windows CE does not support overlapped I/O.

[Code 3] Declaration of ReadFile Method for Reading from Serial

```
[DllImport("coredll.dll")]
private static extern bool ReadFile(
    int hFile,                // Port handle
    byte[] lpBuffer,         // Data to read
    int nNumberOfBytesToWrite, // Number of bytes to read
    ref int lpNumberOfBytesWritten, // Pointer to number of byte read
    int lpOverlapped);       // overlapped I/O
```

The byte type data acquired from “ReadFile” method should be encoded into string type for more intuitive recognition and more convenient use of a sentence. For this encoding, we implemented “ByteArray2String” method using GetString method of ASCIIEncoding class in the .NET Compact Framework.

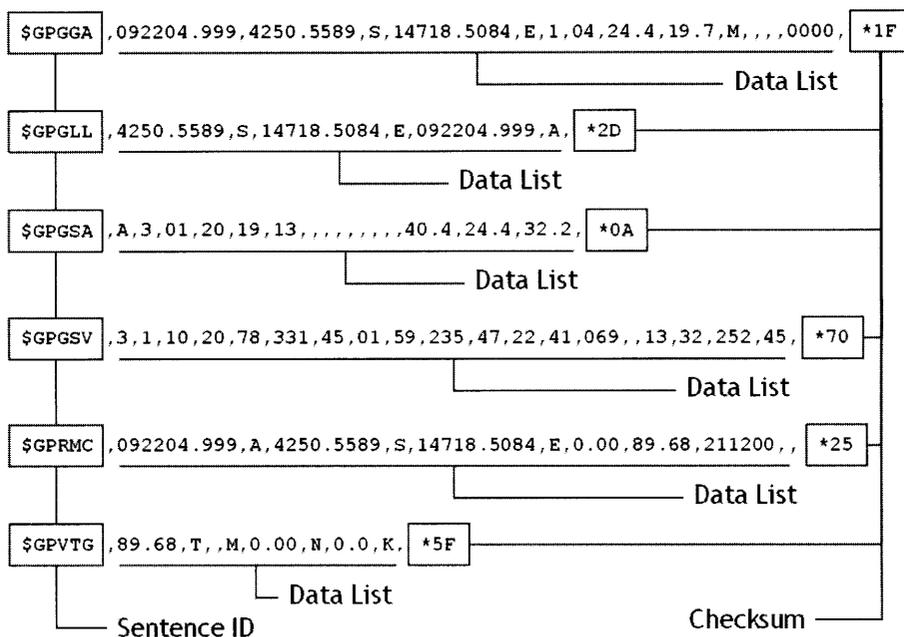
#### 4.1.2. GPS Data Handling

The GPS data acquired by serial communication is composed of 6 common sentences of NMEA-0183 standard such as GPGGA, GPGLL, GPGSA, GPGSV, GPRMC, and GPVTG. Each sentence starts with “\$” character, and ends with CR+LF character<sup>80</sup>. Between the “\$” and CR+LF character, there is an actual

---

<sup>80</sup> Carriage Return + Line Feed

sentence starting with Sentence ID, and ending with Checksum<sup>81</sup>. A sentence has comma delimiters that divide each token such as latitude, longitude, UTC time, etc. The structure of NEMA-0183 is described in [Figure 23]. (Hyoun-Jun Kim, 2003)



[Figure 23] Structure of NMEA-0183 Common Sentences

① GPGGA (Global Positioning System Fixed Data)

● Example

\$GPGGA,092204.999,4250.5589,S,14718.5084,E,1,04,24.4,19.7,M,,0000\*1F

● Structure

No.	Field	Example	Comments
1	Sentence ID	GPGGA	
2	UTC Time	092204.999	hhmmss.sss
3	Latitude	4250.5589	ddmm.mmmm

<sup>81</sup> Checksum is used for checking if the values in a data list are valid.

4	N/S Indicator	S	N = North, S = South
5	Longitude	14718.5084	dddmm.mmmm
6	E/W Indicator	E	E = East, W = West
7	Position Fix Indicator	1	0 = Invalid, 1 = Valid SPS, 2 = Valid DGPS, 3 = Valid PPS
8	Satellite Used	04	Number of satellites used (range 0 to 12)
9	HDOP	24.4	Horizontal dilution of precision
10	Altitude	19.7	Altitude in meters according to WGS-84 ellipsoid
11	Altitude Units	M	M = Meters
12	Geoid Separation		Geoid separation in meters according to WGS-84 ellipsoid
13	Separation Units		M = Meters
14	DGPS Age		Age of DGPS data in seconds
15	DGPS Station ID	0000	
16	Checksum	1F	Result of XOR operation of all the characters between \$ and *
17	Terminator	CR+LF	Carriage Return + Line Feed

## ② GPGLL (Geographic Position - Latitude/Longitude)

- Example

\$GPGLL,4250.5589,S,14718.5084,E,092204.999,A\*2D

- Structure

No.	Field	Example	Comments
1	Sentence ID	GPGLL	
2	Latitude	4250.5589	ddmm.mmmm
3	N/S Indicator	S	N = North, S = South
4	Longitude	14718.5084	dddmm.mmmm
5	E/W Indicator	E	E = East, W = West
6	UTC Time	092204.999	hhmmss.sss
7	Status	A	A = Valid, V = Invalid
8	Checksum	2D	Result of XOR operation of all the characters between \$ and *
9	Terminator	CR+LF	Carriage Return + Line Feed

## ③ GPGSA (GNSS DOP<sup>82</sup> and Active Satellites)

- Example

\$GPGSA,A,3,01,20,19,13,,,,,,,,,40.4,24.4,32.2\*0A

- Structure

<sup>82</sup> GNSS DOP: Global Navigation Satellite System Dilution Of Precision

No.	Field	Example	Comments
1	Sentence ID	GPGSA	
2	Mode 1	A	A = Auto 2D/3D, M = Forced 2D/3D
3	Mode 2	3	1 = No fix, 2 = 2D, 3 = 3D
4	Satellite used 1	01	Satellite used on channel 1
5	Satellite used 2	20	Satellite used on channel 2
6	Satellite used 3	19	Satellite used on channel 3
7	Satellite used 4	13	Satellite used on channel 4
8	Satellite used 5		Satellite used on channel 5
9	Satellite used 6		Satellite used on channel 6
10	Satellite used 7		Satellite used on channel 7
11	Satellite used 8		Satellite used on channel 8
12	Satellite used 9		Satellite used on channel 9
13	Satellite used 10		Satellite used on channel 10
14	Satellite used 11		Satellite used on channel 11
15	Satellite used 12		Satellite used on channel 12
16	PDOP	40.4	Position dilution of precision
17	HDOP	24.4	Horizontal dilution of precision
18	VDOP	32.2	Vertical dilution of precision
19	Checksum	0A	Result of XOR operation of all the characters between \$ and *
20	Terminator	CR+LF	Carriage Return + Line Feed

#### ④ GPGSV (GNSS Satellites in View)

- Example

\$GPGSV,3,1,10,20,78,331,45,01,59,235,47,22,41,069,,13,32,252,45\*70

- Structure

No.	Field	Example	Comments
1	Sentence ID	GPGSV	
2	Number of messages	3	Number of messages in complete message (1-3)
3	Sequence number	1	Sequence number of this entry (1-3)
4	Satellites in view	10	
5	Satellite ID 1	20	Range is 1-32
6	Elevation 1	78	Elevation in degrees (0-90)
7	Azimuth 1	331	Azimuth in degrees (0-359)
8	SNR 1	45	Signal to noise ration in dBHz (0-99)
9	Satellite ID 2	01	Range is 1-32
10	Elevation 2	59	Elevation in degrees (0-90)
11	Azimuth 2	235	Azimuth in degrees (0-359)
12	SNR 2	47	Signal to noise ration in dBHz (0-99)
13	Satellite ID 3	22	Range is 1-32
14	Elevation 3	41	Elevation in degrees (0-90)
15	Azimuth 3	069	Azimuth in degrees (0-359)
16	SNR 3		Signal to noise ration in dBHz (0-99)
17	Satellite ID 4	13	Range is 1-32

18	Elevation 4	32	Elevation in degrees (0-90)
19	Azimuth 4	252	Azimuth in degrees (0-359)
20	SNR 4	45	Signal to noise ration in dBHZ (0-99)
21	Checksum	70	Result of XOR operation of all the characters between \$ and *
22	Terminator	CR+LF	Carriage Return + Line Feed

### ⑤ GPRMC (Recommended Minimum Specific GNSS Data)

- Example

\$GPRMC,092204.999,A,4250.5589,S,14718.5084,E,0.00,89.68,211200,,\*25

- Structure

No.	Field	Example	Comments
1	Sentence ID	GPRMC	
2	UTC Time	092204.999	hhmmss.sss
3	Status	A	A = Valid, V = Invalid
4	Latitude	4250.5589	ddmm.mmmm
5	N/S Indicator	S	N = North, S = South
6	Longitude	14718.5084	dddmm.mmmm
7	E/W Indicator	E	E = East, W = West
8	Speed over ground	0.00	Knots
9	Course over ground	0.00	Degrees
10	UTC Date	211200	DDMMYY
11	Magnetic variation		Degrees (E = East, W = West)
12	Checksum	25	Result of XOR operation of all the characters between \$ and *
13	Terminator	CR+LF	Carriage Return + Line Feed

### ⑥ GPVTG (Course Over Ground and Ground Speed)

- Example

\$GPVTG,89.68,T,,M,0.00,N,0.0,K\*5F

- Structure

No.	Field	Example	Comments
1	Sentence ID	GPVTG	
2	Course	89.68	Measured heading in degrees
3	Reference	T	T = True
4	Course		Measured heading in degrees
5	Reference	M	M = Magnetic
6	Speed	0.00	Horizontal speed
7	Units	N	N = Knots
8	Speed	0.00	Horizontal speed

9	Units	K	K = KM/h
10	Checksum	5F	Result of XOR operation of all the characters between \$ and *
11	Terminator	CR+LF	Carriage Return + Line Feed

We implemented a method named “ParseGPSInput” in the “ClapGPS” class for branching 6 common sentences [Code 5]. This method loops through the incoming GPS data, parsing by CR+LF and allocating each input line to the appropriate parsing method according to the sentence type. In the individual parsing method such as “ParseGGA,” “ParseGLL,” “ParseGSA,” “ParseGSV,” “ParseRMC,” and “ParseVTG,” we tokenized each element of a sentence by using Split method of String class in the .NET Compact Framework, which stores each element (e.g., latitude, longitude, UTC time, etc.) in a string array.

Threading technique is one of the most important parts of the parsing process. In the threading mechanism, at first, a worker thread is created by using a ThreadStart delegate whose parameter is the name of a method to be executed. This ThreadStart delegate is then passed to the constructor method of Thread class. And finally, the instance of Thread class starts to work by the Start method. In our case, “StartGPS” method issues a thread for “GPSThread” method to read NMEA-0183 sentences from a serial port and to parse it using “ParseGPSInput” method [Code 4].

[Code 4] Threading Methods for GPS Data Handling

```
public void StartGPS()
{
    try
    {
        ThreadStart worker = new ThreadStart(GPSThread);
        Thread t = new Thread(worker);
        t.Start();
    }
}
```

```

    }
    catch
    {
    }
}

public void GPSThread()
{
    do
    {
        try
        {
            _data += aClapComm.Input;
            ParseGPSInput();
            Thread.Sleep(_interval);
        }
        catch
        {
        }
    }
    while(!_threading);
}

```

[Code 5] ParseGPSInput Method for Branching 6 Sentences

```

// omission of what precedes
string refLine;
foreach(string line in inlines)
{
    if(line.IndexOf("$GPGGA") >= 0)
        ParseGGA(ref refLine);
    else if(line.IndexOf("$GPGLL") >= 0)
        ParseGLL(ref refLine);
    else if(line.IndexOf("$GPGSA") >= 0)
        ParseGSA(ref refLine);
    else if(line.IndexOf("$GPGSV") >= 0)
        ParseGSV(ref refLine);
    else if(line.IndexOf("$GPRMC") >= 0)
        ParseRMC(ref refLine);
    else if(line.IndexOf("$GPVTG") >= 0)
        ParseVTG(ref refLine);
}
}
// omission of what follows

```

### 4.1.3. Coordinate Transformation

NMEA-0183 data is based on WGS84 coordinate system, whereas the national

base map in Korea is currently based on TM coordinate system. The GRS80 ellipsoid of WGS84 coordinate system has the major axis of 6,378,137 meter and the flattening of 1/298.257223563; the Bessel 1841 ellipsoid of TM coordinate system has the major axis of 6,377,397.155 meter and the flattening of 1/299.15 28128 (Mu-Wook Pyeon, 1991; Young-Bin Nim, 1997; Hjelm, 2002).

Recently in Korea, KGD2002 (Korea Geodetic Datum 2002), which observes ITRF2000 (International Terrestrial Reference Frame), was adopted as the future standard of national coordinate system, and the construction of new base map will be based on KGD2002 coordinate system. The GRS80 ellipsoid of ITRF2000 (KGD2002) coordinate system has the major axis of 6,378,137 meter and the flattening of 1/298.257222101. The comparison of major axis and flattening of the ellipsoid of WGS84, TM, and ITRF2000 (KGD2002) coordinate system is as follows [Table 11].<sup>83</sup>

[Table 11] Ellipsoid Comparison of WGS84, TM, and ITRF2000 Coordinate System

Ellipsoid	Major axis	Flattening
GRS80 of WGS84	6,378,137 m	1/298.257223563
Bessel 1841 of TM	6,377,397.155 m	1/299.15 28128
GRS80 of ITRF2000 (KGD2002)	6,378,137 m	1/298.257222101

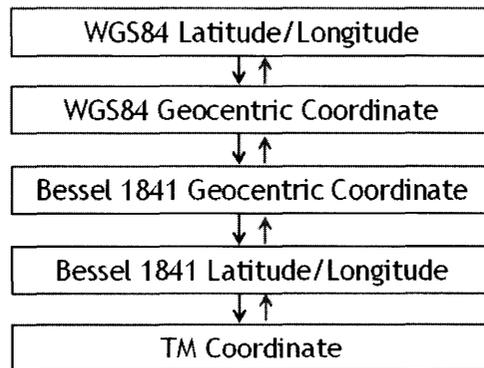
WGS84 is a world geodetic system established by United States, whereas ITRF2000 is a new standard established by international cooperation. Because the coordinate measures of both international standard coordinate systems are almost the same, they are expected to be integrated<sup>84</sup>. However, considering current state of GPS and base map of Korea, the transformation between

<sup>83</sup> [http://www.ngi.go.kr/sub01/sub01\\_03\\_02qna.jsp#1-1](http://www.ngi.go.kr/sub01/sub01_03_02qna.jsp#1-1)

<sup>84</sup> [http://www.ngi.go.kr/sub01/sub01\\_03\\_02qna.jsp#1-3](http://www.ngi.go.kr/sub01/sub01_03_02qna.jsp#1-3)

WGS84 and TM has priority for the practical use of LBS, and ITRF2000-related work should also be performed in the near future.

WGS84 to TM transformation consists of four consecutive processes: ( i ) WGS84 latitude/longitude to WGS84 geocentric coordinate, ( ii ) WGS84 geocentric coordinate to Bessel 1841 geocentric coordinate, ( iii ) Bessel 1841 geocentric coordinate to Bessel 1841 latitude/longitude, and ( iv ) Bessel 1841 latitude/longitude to TM coordinate [Figure 24] (Hyoung-Jun Kim, 2003). The process of TM to WGS84 transformation is vice versa.



[Figure 24] Workflow of Coordinate Transformation between WGS84 and TM

Particularly for the process ( ii ), 7 parameters are necessary for the transformation model of Bursa-Wolf or Molodensky-Badekas. The National Geographic Information Institute of Korea (2003) recommends the optimum value for these 7 parameters. For the process ( iv ), 10.405 second can be optionally added to the longitude of coordinate center point<sup>85</sup>.

<sup>85</sup> Center points in longitude are 125 degree for west, 127 degree for mid, and 129 degree for east region.

We implemented WGS84 to TM transformation for the practical use of GPS coordinate and TM to WGS84 transformation for the possible unification of coordinate system over the world. To implement both transformation processes, we referred to the VB source code of a desktop program by Geo Group Eng<sup>86</sup>, and recomposed it for mobile computing in C#.NET language under the .NET Compact Framework environment. The formulas used for the implementation are as follows.

#### 4.1.3.1. WGS84 Latitude/Longitude to WGS84 Geocentric Coordinate

The formula for transforming WGS84 latitude/longitude  $\{\phi, \lambda, h\}$  to WGS84 geocentric coordinate  $\{X, Y, Z\}$  is as follows [Formula 1].

[Formula 1] WGS84 geocentric coordinate

$$X = (N + h) \cos \phi \cos \lambda$$

$$Y = (N + h) \cos \phi \sin \lambda$$

$$Z = \left( \frac{b^2}{a^2} N + h \right) \sin \phi$$

$$a = 6378137$$

$$f = \frac{1}{298.257223563}$$

$$b = a \left( \frac{1}{f} - 1 \right) / \frac{1}{f}$$

$$N = \frac{a^2}{\sqrt{a^2 \cos^2 \phi + b^2 \sin^2 \phi}}$$

---

<sup>86</sup> <http://geogroup.cc>

#### 4.1.3.2. WGS84 Geocentric Coordinate to Bessel 1841 Geocentric Coordinate

For the transformation of WGS84 ellipsoid to Bessel 1841 ellipsoid and vice versa, the models of Bursa-Wolf and Moldensky-Badekas are the most frequently used. We implemented both models in the “ClapCoord” class, and application developers can choose either way with user-defined 7 parameters.

The generic form of Bursa-Wolf transformation model is presented in [Formula 2] and of Moldensky-Badekas in [Formula 3]. Moldensky-Badekas model is recommended in Korea because it eliminates the high correlation between datum points and parameters. The standardized 7 parameters recommended by National Geographic Information Institute (2003) for Moldensky-Badekas model are presented in [Table 12].

[Formula 2] Bursa-Wolf Transformation Model

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = (1 + \Delta s) \begin{bmatrix} 1 & \kappa & -\phi \\ -\kappa & 1 & \omega \\ \phi & -\omega & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

[Formula 3] Moldensky-Badekas Transformation Model

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} + (1 + \Delta s) \begin{bmatrix} 0 & \kappa & -\phi \\ -\kappa & 0 & \omega \\ \phi & -\omega & 0 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

[Table 12] 7 Parameters for Moldensky-Badekas Transformation

displacement of X, Y, and Z	$\Delta x = -145.907, \Delta y = 505.034, \Delta z = 685.756$
rotation of X, Y, and Z	$\omega = -1.162, \phi = 2.347, \kappa = 1.592$
scale change	$\Delta s = 6.342$

#### 4.1.3.3. Bessel1841 Geocentric Coordinate to Bessel1841 Latitude/Longitude

The formula for transforming Bessel 1841 geocentric coordinate  $\{X, Y, Z\}$  to Bessel 1841 latitude/longitude  $\{\phi, \lambda, h\}$  is as follows [Formula 4].

[Formula 4] Bessel Geocentric to Bessel Latitude/Longitude

$$\phi = \arctan(Z / l)$$

$$\lambda = \arctan(Y / X)$$

$$h = \frac{P}{\cos \phi} - N$$

$$l = \sqrt{\left(\frac{b^2}{a^2} N + h\right)^2 - Z^2}$$

$$p = \sqrt{X^2 + Y^2} = (N + h) \cos \phi$$

#### 4.1.3.4. Bessel 1841 Latitude/Longitude to TM Coordinate

The formula for transforming Bessel 1841 latitude/longitude  $\{\lambda, \phi\}$  to TM coordinate  $\{x, y\}$  is as follows [Formula 5] (Hyoung-Jun Kim, 2003). In case of Korea, the central point of TM coordinate is  $\{500000, 200000\}$ , and there are three referential points in latitude/longitude: west as 38N/125W, mid as 38N/127W, and east as 38N/129E. 10.405 second can be optionally added to the longitude of coordinate center point in order to rectify the distortion TM

coordinate system may have.

[Formula 5] Bessel Latitude/Longitude to TM

$$\begin{aligned} \frac{x}{N} &= \frac{\lambda}{\rho} \cos \rho + \frac{\lambda^3 \cos^3 \varphi}{6\rho^3} (1 - t^2 + \eta^2) + \frac{\lambda^5 \cos^5 \varphi}{120\rho^5} (5 - 18t^2 + t^4 + 14\eta^2 - 58t^2\eta^2) \\ &+ \frac{\lambda^7}{5040\rho^7} \cos^7 \varphi (61 - 479t^2 + 179t^4 - t^6) \\ \frac{y}{N} &= \frac{S_\rho}{N} + \frac{\lambda^2}{2\rho^2} \sin \rho \cos \rho + \frac{\lambda^4}{24\rho^4} \sin \rho \cos^3 \varphi (5 - t^2 + 9\eta^2 + 4\eta^4) \\ &+ \frac{\lambda^6}{720\rho^6} \sin \varphi \cos^5 \varphi (61 - 58t^2 + t^4 + 270\eta^2 - 330t^2\eta^2) \\ &+ \frac{\lambda^8}{40320\rho^8} \sin \rho \cos^7 \varphi (1385 - 3111t^2 + 543t^4 - t^6) \end{aligned}$$

#### 4.1.4. API Development

As a result of the functionality implementation, we provide API functions covering serial communication, GPS data handling, and coordinate transformation. The typical public methods and properties we implemented for application development are presented in [Table 13] to [Table 17].

[Table 13] Typical API Methods in ClapComm Class

Method	Parameter	Return	Description
ClapComm			Default constructor
OpenPort			Opens serial port with CommState <sup>87</sup> setting
ClosePort			Closes serial port
GetCommState		CommState	Gets current CommState setting
SetCommState	CommState		Sets CommState
ByteArray2String	Byte[]	string	Converts byte array to string for GPS input

<sup>87</sup> CommState is an instance of the utility class for getting and setting the state of serial port like baud rate, data bit, parity, stop bit, etc.

[Table 14] Typical API Methods in ClapGPS Class

Method	Parameter	Return	Description
ClapGPS			Default constructor
InitGPS			Initializes GPS with ClapComm instance
OpenPort		bool	Opens serial port with ClapComm instance and return the result of success/fail
ClosePort		bool	Closes serial port with ClapComm instance and return the result of success/fail
StartGPS			Issues a thread of GPSThread method
GPSThread			Reads from serial port for parsing GPS input
ParseGPSInput	string		Branches GPS input according to 6 types of NMEA-0183 sentence
ParseGPGGA	string	string[]	Parses GPGGA sentence and return the string array composed of parsed information
ParseGPGLL	string	string[]	Parses GPGLL sentence and return the string array composed of parsed information
ParseGPGSA	string	string[]	Parses GPGSA sentence and return the string array composed of parsed information
ParseGPGSV	string	string[]	Parse GPGSV sentence and return the string array composed of parsed information
ParseGPRMC	string	string[]	Parse GPRMC sentence and return the string array composed of parsed information
ParseGPVTG	string	string[]	Parse GPVTG sentence and return the string array composed of parsed information

[Table 15] Typical Properties in ClapGPS Class

Property	Type	Description
Interval	int	Gets and sets the interval (in second) of reading from GPS receiver
Status	string	Gets GPS mode (A: valid, V: awaiting position fix)
Latitude	double	Gets latitude in ddmm.mmmm
Longitude	double	Gets longitude in dddmm.mmmm
NorthTM	double	Gets northing of TM transformed from Latitude
EastTM	double	Gets easting of TM transformed from Longitude
Speed	double	Gets the moving speed (in knot)
Course	double	Gets the moving course (in degree)
UTCDate	string	Gets current UTC date
UTCTime	string	Gets current UTC time

[Table 16] Typical API Methods in ClapCoord Class

Method	Parameter	Return	Description
WGS2TM	double[]	double[]	Transforms WGS84 latitude/longitude to TM coordinate with default setting (Bursa-Wolf model with MOCT 7 parameters and the central point of 38N/127E with additional 10.405 second)
WGSLatLon2WGS_GC	double[]	double[]	Transforms WGS84 latitude/longitude to

			WGS84 geocentric coordinate
WGS_GC2Bessel_GC	double[]	double[]	Transforms WGS84 geocentric coordinate to Bessel 1841 geocentric coordinate; Calls TransBursa or TransMolodensky method according to the property value of TransMode
Bessel_GC2BesselLatLon	double[]	double[]	Transforms Bessel 1841 geocentric coordinate to Bessel 1841 latitude/longitude
BesselLatLon2TM	double[]	double[]	Transforms Bessel 1841 latitude/longitude to TM coordinate according to the property value of CentralPoint and Add10_405
TM2WGS	double[]	double[]	Transforms TM coordinate to WGS84 latitude/longitude with default setting (Bursa-Wolf model with MOCT 7 parameters and the central point of 38N/127E with additional 10.405 second)
TM2BesselLatLon	double[]	double[]	Transforms TM coordinate to Bessel 1841 latitude/longitude according to the property value of CentralPoint and Add10_405
BesselLatLon2Bessel_GC	double[]	double[]	Transforms Bessel 1841 latitude/longitude to Bessel 1841 geocentric coordinate
Bessel_GC2WGS_GC	double[]	double[]	Transforms Bessel 1841 geocentric coordinate to WGS84 geocentric coordinate; Calls TransBursa or TransMolodensky method according to the property value of TransMode
WGS_GC2WGSLatLon	double[]	double[]	Transforms WGS84 geocentric coordinate to WGS84 latitude/longitude
TransBursa	double[]	double[]	Transforms WGS84 geocentric coordinate to Bessel 1841 geocentric coordinate according to Bursa-Wolf model
TransMolodensky	double[]	double[]	Transforms WGS84 geocentric coordinate to Bessel 1841 geocentric coordinate according to Molodensky-Badekas model
Set7Params	double[]		Sets 7 parameters for Bursa-Wolf or Molodensky-Badekas model

[Table 17] Typical Properties in ClapCoord Class

Property	Type	Description
TransMode	string	Gets and sets the transformation option for the transformation between WGS84 geocentric and Bessel 1841 geocentric coordinate (B: Bursa-Wolf model, M: Molodensky-Badekas model)
CentralPoint	string	Gets and sets the central point of Korea for the transformation

		between Bessel latitude/longitude and TM coordinate (WEST: 38N/125E, MID: 38N/127E, EAST: 38N/129E)
Add10_405	bool	Gets and sets whether to add or not the 10.405 second to the central point for the transformation between Bessel latitude/longitude and TM coordinate (true: add, false: not)

### 4.1.5. Feasibility Test

To verify our API functions, we implemented feasibility testing applications such as coordinate transformation, monitoring GPS data, and archiving GPS location. [Figure 25] shows the testing screenshot of WGS84 to TM transformation using Molodensky-Badekas model with 7 parameters of National Geographic Information Institute (2003). To compare with the result of other transformation program, we performed the same test using the program of Geo Group Eng. [Table 18].

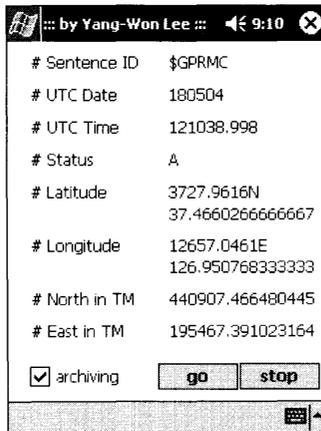


[Figure 25] Testing Screenshot of WGS84 to TM Transformation with or without 10.405 Seconds Addition

[Table 18] Result Comparison of Coordinate Transformation

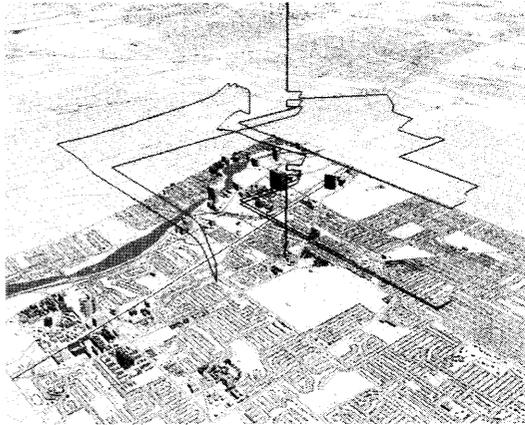
Result	Our Program	Geo Group Eng. Program
North in TM (w/10.405 seconds addition)	444675.960635522	444675.96063
East in TM (w/10.405 seconds addition)	199566.173291048	199566.17329
North in TM (w/o10.405 seconds addition)	444675.951236518	444675.95123
East in TM (w/o10.405 seconds addition)	199821.711697706	199821.71170

[Figure 26] shows the real-time monitoring of GPS data on PDA. In this case, we adopted GPRMC sentence, and presented UTC date/time, latitude/longitude in “dddmm.mmmm” format and north/east in TM every 2 seconds. Location data acquired from GPS receiver is monitored and archived in real-time. The archived data can be applied to space-time GIS like traffic pattern analysis by tracking a person’s movement [Figure 27].



[Figure 26] Testing Screenshot of GPS Data Monitoring

Time	Lat	Long	Alt	Speed	Course	Fix	Age	Mode
131809.485	3727.9483N	12657.2900E	440404.664445854			1	0.0	2D
131809.485	3727.9483N	12657.2901E	440404.695103587			1	0.0	2D
131811.483	3727.9483N	12657.2901E	440404.695103587			1	0.0	2D
131813.482	3727.9487N	12657.2901E	440405.434115047			1	0.0	2D
131815.482	3727.9487N	12657.2902E	440405.434215047			1	0.0	2D
131817.482	3727.9483N	12657.2979E	440397.289526753			1	0.0	2D
131819.482	3727.9443N	12657.2979E	440397.289526753			1	0.0	2D
131821.482	3727.9540N	12657.3081E	440389.331798708			1	0.0	2D
131823.482	3727.9540N	12657.3081E	440389.331798708			1	0.0	2D
131825.482	3727.9540N	12657.3081E	440389.331798708			1	0.0	2D
131827.482	3727.9540N	12657.3081E	440389.331798708			1	0.0	2D
131829.482	3727.9540N	12657.3081E	440389.331798708			1	0.0	2D
131831.481	3727.8729N	12657.2107E	440281.112459020			1	0.0	2D
131833.481	3727.8495N	12657.1115E	440221.913730235			1	0.0	2D
131835.48	3727.8170N	12657.2951E	440161.823501923			1	0.0	2D
131837.48	3727.7851N	12657.2778E	440102.81809428			1	0.0	2D
131839.48	3727.7851N	12657.2778E	440102.81809428			1	0.0	2D
131841.478	3727.7980N	12657.2737E	440071.189642220			1	0.0	2D
131843.478	3727.7980N	12657.2737E	440071.189642220			1	0.0	2D
131845.478	3727.7481N	12657.2719E	440034.380197854			1	0.0	2D
131847.478	3727.7481N	12657.2719E	440034.380197854			1	0.0	2D
131849.478	3727.7272N	12657.2699E	439996.091073846			1	0.0	2D
131851.478	3727.7272N	12657.2699E	439996.091073846			1	0.0	2D
131853.478	3727.7078N	12657.2687E	439959.886129823			1	0.0	2D
131855.478	3727.7078N	12657.2687E	439959.886129823			1	0.0	2D
131857.478	3727.6915N	12657.2767E	439929.678720073			1	0.0	2D
131859.478	3727.6915N	12657.2767E	439929.678720073			1	0.0	2D
131861.478	3727.6842N	12657.2835E	439916.159478880			1	0.0	2D
131863.478	3727.6842N	12657.2835E	439916.159478880			1	0.0	2D
131905.478	3727.6826N	12657.3218E	439911.330601773			1	0.0	2D
131907.478	3727.6826N	12657.3218E	439911.330601773			1	0.0	2D
131909.478	3727.6779N	12657.3487E	439904.46946291			1	0.0	2D
131911.478	3727.6779N	12657.3487E	439904.46946291			1	0.0	2D



[Figure 27] GPS Data Archiving for Traffic Pattern Analysis

## 4.2. Mobile Embedded Mapping

### 4.2.1. Map Data Pre-processing

Considering the limited resource of mobile device, a lightweight map is sometimes necessary for faster display in the mobile embedded environment. In fact, it is a kind of tradeoff between quality and speed. In case of the application which requires a precise map, we have to spend more time for downloading a map; in case of the application which requires fast display of a map, we have to use simplified map.

Vector data, whether it is a polygon, line, or point, basically consists of point elements, and the principle of the vector simplification is based on decreasing the number of point element. The most widely accepted method of vector simplification is Douglas-Peucker algorithm (Chul-Sue Hwang, 1998). Though this algorithm has been used primarily for line feature, its principle can be applied to the polygon feature which has boundary characteristics such as

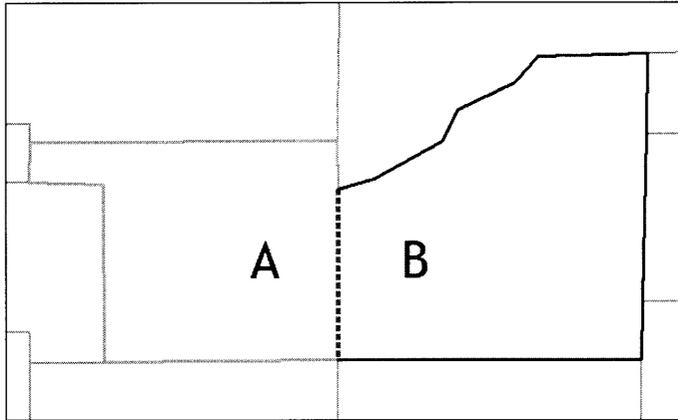
administrative district and land parcel map.

We implemented a vector simplification module using ArcObjects by combining Douglas-Peucker algorithm and topological operator such as Union, Intersect, and Difference. This module is a pre-processing module which is optionally chosen by application developers.

If we apply Douglas-Peucker algorithm to the polygon feature without any adjustment, the boundary discordance problem between each simplified polygon may occur [Figure 28]. This problem arises from the shared boundary part among two or more polygon features. Because the shared boundary part between “polygon A” and “polygon B” is the same, whereas the portion of the part in polygon A is definitely different from the portion of the part in polygon B [Figure 29], the simplification processes for polygon A and for polygon B create different number and composition of point set for this shared part.



[Figure 28] Vector Simplification of Polygon without Adjustment



[Figure 29] Shared Boundary Part between Two Polygons

For the adjustment to overcome the discordance problem, we invented “disassemble-simplify-reassemble” method. This method, at first, disassembles a polygon feature into several line features according to the intersection with its neighboring polygons. Then it simplifies each line feature using Douglas-Peucker algorithm, and finally reassembles the simplified lines features into one polygon feature. This process iterates for each polygon feature.

In this method, maximum offset tolerance can be specified in double value as a simplification option. [Figure 30] is an example lightweight polygon Shapefile without serious loss of quality or discordance of shared boundary. The file size of simplified output varies (10-50% of original file) depending on the option of maximum offset tolerance.



[Figure 30] Vector Simplification by “Disassemble-Simplify-Reassemble” Method

## 4.2.2. Visualization

### 4.2.2.1. Parsing Shapefile

For the visualization of Shapefile which is an industry standard, we, at first, scrutinized the structure of the Shapefile encoded in binary<sup>88</sup> format. A Shapefile consists of a main file (.shp), an index file (.shx), and a dBase table (.dbf). The main file is a direct access, variable-record-length file in which each record describes a shape with a list of its vertices. In the index file, each record contains the offset of the corresponding main file record from the beginning of the main file. The dBase table contains feature attributes with one record per feature. Attribute records in the dBase file must be in the same order as records in the main file. Integers and double-precision integers that make up the data description fields in the header file and record contents in the main file are in little endian (Intel standard) byte order. The integers and double-precision floating point numbers that make up the rest of the file and file management are in big endian (Sun standard) byte order.<sup>89</sup> (ESRI, 1998)

#### ① Organization of Main file

The main file (.shp) contains a fixed-length file header followed by variable-length records. Each variable-length record is made up of a fixed-length record header followed by variable-length record contents. [Table 19] illustrates the main file organization.

---

<sup>88</sup> Binary describes a numbering scheme in which there are only two possible values for each digit: 0 and 1. The term also refers to any digital encoding/decoding system in which there are exactly two possible states. In digital data memory, storage, processing, and communications, the 0 and 1 value are sometimes called “low” and “high,” respectively. (<http://whatis.com>)

<sup>89</sup> Big byte order: Byte 0 - Byte 1 - Byte 2 - Byte 3  
Little byte order: Byte 3 - Byte 2 - Byte 1 - Byte 0

[Table 19] Organization of Main File in Shapefile

File Header	
Record Header	Record Contents
Record Header	Record Contents
.....	.....
.....	.....
Record Header	Record Contents
Record Header	Record Contents

The main file header is 100 bytes long. [Table 20] shows the fields in the file header with their byte position, value, type, and byte order. The values for “Shape Type” are 1 (Point), 3 (PolyLine), 5 (Polygon), etc. The value of “File Length” is the length of the file in 16-bit words (including the fifty 16-bit words that make up the header).

[Table 20] Header Description of Main File in Shapefile

Position	Field	Value	Type	Byte Order
Byte 0	File Code	9994	Integer	Big
Byte 4	Unused	0	Integer	Big
Byte 8	Unused	0	Integer	Big
Byte 12	Unused	0	Integer	Big
Byte 16	Unused	0	Integer	Big
Byte 20	Unused	0	Integer	Big
Byte 24	File Length	File Length	Integer	Big
Byte 28	Version	1000	Integer	Little
Byte 32	Shape Type	Shape Type	Integer	Little
Byte 36	Bounding Box	Xmin	Double	Little
Byte 44	Bounding Box	Ymin	Double	Little
Byte 52	Bounding Box	Xmax	Double	Little
Byte 60	Bounding Box	Ymax	Double	Little
Byte 68*	Bounding Box	Zmin	Double	Little
Byte 76*	Bounding Box	Zmax	Double	Little
Byte 84*	Bounding Box	Mmin	Double	Little
Byte 92*	Bounding Box	Mmax	Double	Little

The header of each record stores the record number and content length for

the record. Record headers have a fixed length of 8 bytes. [Table 21] shows the fields in the file record headers with their byte position, value, type, and byte order. The content length for a record is the length of the record contents section measured in 16-bit words. Each record, therefore, contributes (4 + content length) 16-bit words toward the total length of the file, as stored at Byte 24 in the file header.

[Table 21] Record Header of Main File in Shapefile

Position	Field	Value	Type	Byte Order
Byte 0	Record Number	Record Number	Integer	Big
Byte 4	Content Length	Content Length	Integer	Big

Shapefile record contents consist of a shape type followed by the geometric data for the shape. The length of the record contents depends on the number of parts and vertices in a shape. [Table 22] to [Table 24] show the contents of the shape such as Point, PolyLine, and Polygon.

A Point consists of a pair of double-precision coordinates in the order X, Y.

[Table 22] Contents of Point in Shapefile

```
Point
{
    Double X // X coordinate
    Double Y // Y coordinate
}
```

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	1	Integer	1	Little
Byte 4	X	X	Double	1	Little
Byte 12	Y	Y	Double	1	Little

A PolyLine is an ordered set of vertices that consists of one or more parts. A

part is a connected sequence of two or more points. "Parts" may or may not be connected to one another. Parts may or may not intersect one another. "Box" stores the bounding box in the order of Xmin, Ymin, Xmax, Ymax. NumParts is the number of parts in the PolyLine. NumPoints is the total number of points for all parts. "Parts" is an array of length NumParts. For each PolyLine, it stores the index of its first point in the "Points" array. "Points" is an array of length NumPoints. The points for each part in the PolyLine are stored end to end. The points for points for Part 2 follow the points for Part 1, and so on. There is no delimiter in the points array between parts.

[Table 23] Contents of PolyLine in Shapefile

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	3	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little

(Note: X = 44 + 4 \* NumParts)

A Polygon consists of one or more rings. A ring is a connected sequence of four or more points that form a closed, non-self-intersecting loop. The structure of Polygon is basically the same as PolyLine.

[Table 24] Contents of Polygon in Shapefile

```

Polygon
{
    Double[4]      Box           // Bounding Box
    Integer        NumParts      // Number of Parts
    Integer        NumPoints     // Total Number of Points
    Integer[NumParts] Parts     // Index to First Point in Part
    Point[NumPoints] Points     // Points for All Parts
}
    
```

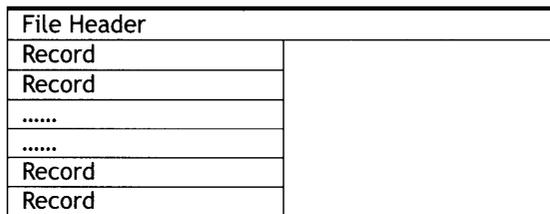
Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	5	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little

(Note: X = 44 + 4 \* NumParts)

② Organization of Index File

The index file (.shx) contains a 100-byte header followed by 8-byte, fixed-length records. [Table 25] illustrates the index file organization.

[Table 25] Organization of Index File in Shapefile



The index file header is identical in organization to the main file header described above. The file length stored in the index file header is the total length of the index file in 16-bit words (the fifty 16-bit words of the header plus 4 times the number of records). The l'th record in the index file stores the offset and content length for the l'th record in the main file. [Table 26] shows

the fields in the file header with their byte position, value, type, and byte order.

[Table 26] Record Header of Index File in Shapefile

Position	Field	Value	Type	Byte Order
Byte 0	Offset	Offset	Integer	Big
Byte 4	Content Length	Content Length	Integer	Big

The offset of a record in the main file is the number of 16-bit words from the start of the main file to the first byte of the record header for the record. Thus, offset for the first record in the main file is 50, given the 100-byte header. The content length stored in the index record is the same as the value stored in the main file record header.

### ③ Organization of dBase File

The dBase file (.dbf) contains any desired feature attributes or attributes keys to which other tables can be joined. The main characteristics of header part are described in [Table 27]; record part in [Table 28].

[Table 27] Characteristics of Header Part in dBase File

Record count	Number of records in the table
Header length	Number of bytes in the header
Field count	$(\text{header length} - 1) / 32 - 1$
Record length	Number of bytes per record
Field descriptor array	32 bytes per field composed of field name, type, length, etc.

[Table 28] Characteristics of Record Part in dBase File

Starting position	$32 + (\text{field count} * 32) + 1$
Each record	The first 1 byte is for delete flag and the remaining bytes are for field value

#### 4.2.2.2. Viewing Shapefile

The geographic coordinate of a Shapefile should be transformed to the screen coordinate of mobile devices. We implemented the methods for the transformation in the “ClapMapFrame” class, using the relationship between geographic and screen coordinate (geographic to screen and screen to geographic).

- Geographic to screen

$$\text{screen\_x} = (\text{geographic\_x} - \text{viewport\_x\_min}) / \text{map\_unit\_per\_pixel\_x}$$
$$\text{screen\_y} = (\text{viewport\_y\_max} - \text{geographic\_y}) / \text{map\_unit\_per\_pixel\_y}$$

- Screen to geographic

$$\text{geographic\_x} = \text{viewport\_x\_min} + \text{screen\_x} * \text{map\_unit\_per\_pixel\_x}$$
$$\text{geographic\_y} = \text{viewport\_y\_max} - \text{screen\_y} * \text{map\_unit\_per\_pixel\_y}$$

Drawing point, line, and polygon on the screen under the .NET Compact Framework is made possible by using System.Drawing.Graphics class with the screen coordinates transformed from geographic coordinate. The methods in the “ClapMapDrawer” class we implemented for map visualization are based on the C#.NET methods as follows:

- DrawEllipse(System.Drawing.Pen pen, int x, int y, int width, int height)
- FillEllipse(System.Drawing.Brush brush, int x, int y, int width, int height)
- DrawLine(System.Drawing.Pen pen, int x1, int y1, int x2, int y2)
- DrawPolygon(System.Drawing.Pen pen, Point[] points)
- FillPolygon(System.Drawing.Brush brush, Point[] points)

Once a map shows up on the screen, users can control the viewport<sup>90</sup> of the map with zooming and panning. We implemented the methods for map controlling in the “ClapMapController” class, using the principle of zooming, panning, and fitting to screen as follows:

- Zoom-in

$\text{viewport\_x\_min} += \text{viewport\_delta\_x} / \text{zoom\_factor}$

$\text{viewport\_y\_min} += \text{viewport\_delta\_y} / \text{zoom\_factor}$

$\text{viewport\_x\_max} -= \text{viewport\_delta\_x} / \text{zoom\_factor}$

$\text{viewport\_y\_max} -= \text{viewport\_delta\_y} / \text{zoom\_factor}$

- Zoom-out

$\text{viewport\_x\_min} -= \text{viewport\_delta\_x} / \text{zoom\_factor}$

$\text{viewport\_y\_min} -= \text{viewport\_delta\_y} / \text{zoom\_factor}$

$\text{viewport\_x\_max} += \text{viewport\_delta\_x} / \text{zoom\_factor}$

$\text{viewport\_y\_max} += \text{viewport\_delta\_y} / \text{zoom\_factor}$

- Panning

$\text{viewport\_x\_min} = \text{center\_x} - \text{viewport\_delta\_x}$

$\text{viewport\_y\_min} = \text{center\_y} - \text{viewport\_delta\_y}$

$\text{viewport\_x\_max} = \text{center\_x} + \text{viewport\_delta\_x}$

$\text{viewport\_y\_max} = \text{center\_y} + \text{viewport\_delta\_y}$

- Fitting to screen

$\text{viewport\_x\_min} = \text{extent\_xmin}$

$\text{viewport\_y\_min} = \text{extent\_ymin}$

$\text{viewport\_x\_max} = \text{extent\_xmax}$

$\text{viewport\_y\_max} = \text{extent\_ymax}$

---

<sup>90</sup> The displayed part of a map: it is always equal to or smaller than the extent of a map.

### 4.2.2.3. Double Buffering

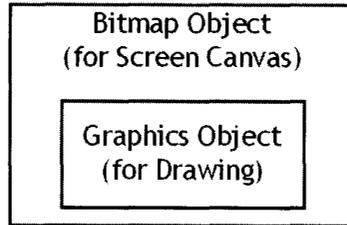
Double buffering is a method for the smooth visualization of a map. When a graphic is complex or is used repeatedly, we can reduce display time by rendering it to an off-screen buffer and then copying the buffer to the screen<sup>91</sup>. We allocate an area according to the size of the screen in the system memory and draw to it, instead of drawing directly to the video memory. When the drawing is finished, the off-screen buffer is written to the video memory and the image shows up. It is a lot faster for a computer to draw to the system memory rather than make an I/O request every time we want to write to the screen. In addition, double buffering smoothens the transition between one image and another on the screen, thereby preventing the flickering of the screen.<sup>92</sup>

We implemented double buffering technique in the “ClapMapDrawer” class. At first, we create an instance of `System.Drawing.Bitmap` class and an instance of `System.Drawing.Graphics` class through the `System.Drawing.Graphics.FromImage` method using the `Bitmap` instance already created as a parameter. And then, we issue `drawOOO` or `fillOOO` method according to the vector feature. The double buffer we prepared is “`Bitmap`” for screen canvas and “`Graphics`” for drawing a map [Figure 31]. The first target of the drawing is “`Graphics`” object and the second target is “`Bitmap`” object. The “`Graphics`” class has useful methods for drawing point, line, polygon, etc., and “`Bitmap`” class prevents flickering of the screen. After drawing on the “`Graphics`,” we copy the buffer on the “`Bitmap`” at once.

---

<sup>91</sup> Double buffering is also called “off-screen imaging.”

<sup>92</sup> <http://java.sun.com/docs/books/tutorial/2d/images/doublebuffering.html> and [http://www.osdever.net/tutorials/pdf/GUI\\_tut.pdf](http://www.osdever.net/tutorials/pdf/GUI_tut.pdf)



[Figure 31] Structure of Double Buffer

#### 4.2.2.4. Advanced Rendering

Advanced rendering such as choropleth map, symbol map, and chart map improves the efficiency of communication for geographic information delivery. Choropleth map, where each spatial unit is filled with a uniform color or pattern, is appropriate for the data that have been scaled or normalized like density information, expressed as “per unit area.”<sup>93</sup>

Symbol map is applied to point features to represent the notable features such as highway shield and important building. It also can be applied to point or polygon features to represent the magnitude of a specific attribute by adjusting the size of symbol according to the attribute value. Chart map shows the size or degree of an attribute value in the form of bar or pie which is located on the polygon centroid, thus is useful for comparing two or more attribute values.

We implemented the map representing methods such as choropleth map, symbol map, and chart map in the “ClapMapRepresenter” class. For example, choropleth map requires the methods to determine the range of classification and the color ramp according to each class. In our methods [Code 6] [Code 7], the classification range is regarded as “min <= ? < max,” and very tiny number

---

<sup>93</sup> [http://www.ncgia.ucsb.edu/education/curricula/cctp/units/unit47/html/mas\\_form.html](http://www.ncgia.ucsb.edu/education/curricula/cctp/units/unit47/html/mas_form.html)

(0.0000000001) is added to the last max value for the classification.

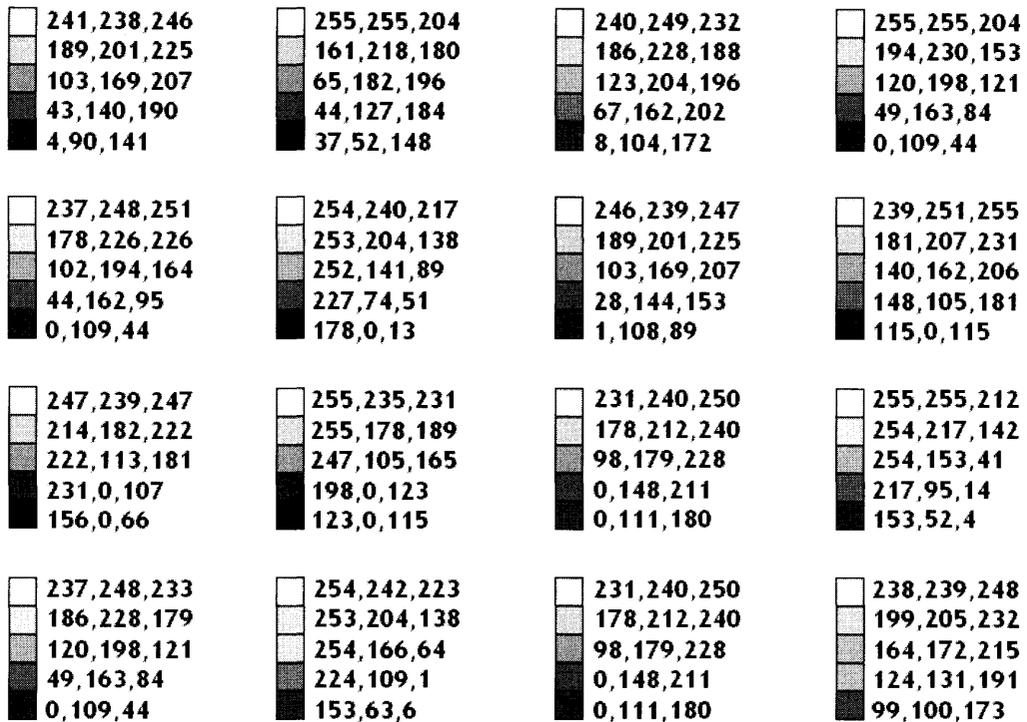
For the default color ramp, we adopted the sequential color ramp recommended by the National Geographic Information Institute of Korea (2002) [Figure 32]. For the dynamic allocation of the color ramp, we implemented the method which arranges color ramp with the RGB value of start color and end color [Code 8].

#### [Code 6] Equal Interval Method for Classification Rendering

```
public void SetEqualInterval()
{
    _breakIncrement = (_maxVal - _minVal) / _numOfClass;
    double increment = _minVal;
    double lastMin = 0;
    for(int i = 0; i < _numOfClass - 1; i++)
    {
        _breakMin[i] = increment;
        increment += _breakIncrement;
        _breakMax[i] = increment;
        lastMin = increment;
    }
    _breakMin[_numOfClass - 1] = lastMin;
    _breakMax[_numOfClass - 1] = _maxVal + 0.0000000001;
}
```

#### [Code 7] Quantile Method for Classification Rendering

```
public void SetQuantile()
{
    _recordIncrement = Convert.ToInt32(Math.Round(Convert.ToDouble(_numOfRecord /
_numOfClass)));
    int increment = 0;
    int lastMin = 0;
    for(int i = 0; i < _numOfClass - 1; i++)
    {
        _breakMin[i] = _fieldValues[increment];
        increment += _recordIncrement;
        _breakMax[i] = _fieldValues[increment];
        lastMin = increment;
    }
    _breakMin[_numOfClass - 1] = _fieldValues[lastMin];
    _breakMax[_numOfClass - 1] = _fieldValues[_numOfRecord - 1] + 0.0000000001;
}
```



[Figure 32] Default Color Ramp for Choropleth Map

[Code 8] Method for Arranging Color Ramp

```

public void SetColorRamp()
{
    int sRed = _startColor.R;
    int sGreen = _startColor.G;
    int sBlue = _startColor.B;
    int eRed = _endColor.R;
    int eGreen = _endColor.G;
    int eBlue = _endColor.B;
    int rRange = eRed - sRed;
    int gRange = eGreen - sGreen;
    int bRange = eBlue - sBlue;
    int rIncr = Convert.ToInt32(rRange / _numOfClass);
    int gIncr = Convert.ToInt32(gRange / _numOfClass);
    int bIncr = Convert.ToInt32(bRange / _numOfClass);

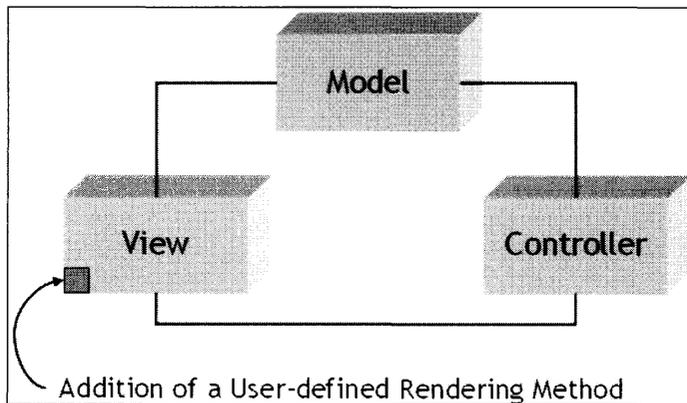
    _colorRamp = new Color[_numOfClass];
    _colorRamp[0] = _startColor;
    _colorRamp[_numOfClass - 1] = _endColor;

    int theRed = sRed + rIncr;
    int theGreen = sGreen + gIncr;
    int theBlue = sBlue + bIncr;

```

```
for(int i = 1; i < _numOfClass - 1; i++)
{
    _colorRamp[i] = Color.FromArgb(theRed, theGreen, theBlue);
    theRed = theRed + rlncre;
    theGreen = theGreen + glncre;
    theBlue = theBlue + blncre;
}
_brush = new Brush[_numOfClass];
for(int i = 0; i < _numOfClass; i++)
    _brush[i] = new SolidBrush(_colorRamp[i]);
}
```

One of the purposes of the component library is to provide extensibility for the application developers who want to add new functionality to the existing functionalities. The component library of our geocomputing platform, which has MVC (Model-View-Controller) architecture, allows application developers to add user-defined rendering methods in their applications for another type of map representation [Figure 33].



[Figure 33] Extensibility for Advanced Rendering in the Geocomputing Platform

#### 4.2.2.5. Viewing Image Map

We support generic image files such as BMP, GIF, JPG, and PNG for the image

map. Two important functionalities for viewing image map are to adjust an image on the screen according to real coordinate, and to expand/shrink the image according to zooming factor. Besides, these functionalities should be combined with vector-on-raster overlay.

To construct a map extent in real coordinate, we employed a coordinate file named “Geographic Coordinate File” (.gcf) that includes the coordinates of Xmin, Ymin, Xmax, and Ymax. An instance of the “ClapImageMap” class imports the coordinate file and constructs a map extent with an instance of the “ClapMapFrame” class. [Code 9] shows the process of tokenizing coordinate file with comma delimiter and assigning the coordinates of bounding box.

[Code 9] Method for Loading the Bounding Box of Image Map

```
private void LoadExtent(string filename)
{
    char delimiter = ',';
    FileInfo gcf = new FileInfo(filename);
    StreamReader sr = gcf.OpenText();
    string bulk = sr.ReadLine();
    sr.Close();

    string[] corners = bulk.Split(delimiter);
    _extent.Xmin = Convert.ToDouble(corners[0]);
    _extent.Ymin = Convert.ToDouble(corners[1]);
    _extent.Xmax = Convert.ToDouble(corners[2]);
    _extent.Ymax = Convert.ToDouble(corners[3]);
}
```

Once a map shows up on the screen, users can control the viewport of the map by zooming or panning. [Code 10] shows the process of adjusting the viewport of the image map according to the zooming factor. We used a source rectangle to expand or shrink the image and a destination rectangle as a canvas of the image.

#### [Code 10] Method for Zooming Image Map

```
public void ZoomRaster()
{
    // omission of what precedes
    offsetX = _bitmap.Width * ((_mapFrame.ZoomFactor-1) /
(2*_mapFrame.ZoomFactor));
    offsetY = _bitmap.Height * ((_mapFrame.ZoomFactor-1) /
(2*_mapFrame.ZoomFactor));
    srcRect.X = Convert.ToInt32(offsetX - moreX);
    srcRect.Y = Convert.ToInt32(offsetY - moreY);
    srcRect.Width = Convert.ToInt32((_bitmap.Width/_mapFrame.ZoomFactor) +
(2*moreX));
    srcRect.Height = Convert.ToInt32((_bitmap.Height/_mapFrame.ZoomFactor) +
(2*moreY));
    destRect.X = 0;
    destRect.Y = 0;
    destRect.Width = SCREEN_WIDTH;
    destRect.Height = SCREEN_HEIGHT;
    _gr.DrawImage(_bitmap, destRect, srcRect, GraphicsUnit.Pixel);
}
```

### 4.2.3. Topological Operation

The topological operations based on vector geometry are required for the process of map visualization and spatial search. We implemented MBR (Minimum Bounding Rectangle), Point-in-Polygon, and Centroid operations which are frequently used.

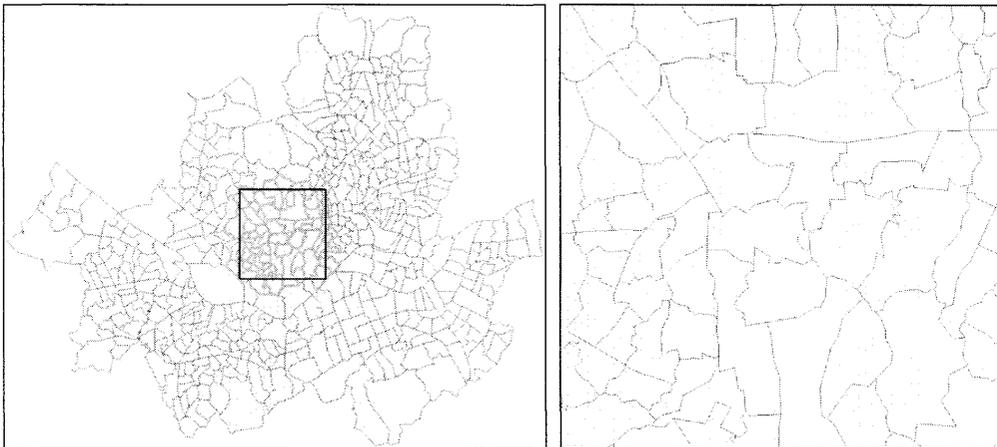
- **MBR Operation:** MBR is a rectangle oriented to the x and y axes, which bounds a geographic feature. Every feature object requires MBR as a property for the operation like “Contains” or “Intersects.” This operation is useful, for example, to decide whether a feature is included in current viewport or not.
- **Point-in-Polygon Operation:** Point-in-Polygon determines whether a point is located within a polygon or not. The most widely known version of the Point-in-Polygon is Sedgewick’s principle (Lewis, 2002). This operation

is useful, for example, to identify a polygon feature according to user's input point like mouse-click.

- Centroid Operation: Centroid is a geometric central point of a polygon, also called center of gravity<sup>94</sup>. This operation is useful, for example to label a polygon feature.

#### 4.2.3.1. MBR Operation

To decide whether a feature is included in the current viewport or not is closely related to the fast drawing of a map. In fact, fast drawing of a map is not the matter of the initial display of a map, but the matter of display with zooming or panning a map which is already displayed on the screen. It's because zooming or panning is the process of searching the appropriate part from the whole map [Figure 34].



[Figure 34] Application of MBR Operation for Viewport Determination

---

<sup>94</sup> <http://mathworld.wolfram.com/GeometricCentroid.html>

We implemented MBR-based search method named “Contains” in the “ClapRect” class for managing a rectangle such as map extent and viewport. This method determines whether one MBR contains another MBR given as a parameter, or not. The determination procedure is as follows [Code 11].

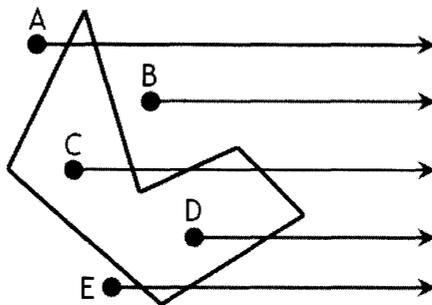
[Code 11] Method for MBR Operation

```
public bool Contains(ClapRect rect)
{
    // omission of what precedes
    if(rect.Xmin <= this.Xmax && rect.Xmax >= this.Xmax)
    {
        if(this.Ymax >= rect.Ymax && this.Ymin <= rect.Ymax)
            return true;
        if(this.Ymax >= rect.Ymin && this.Ymin <= rect.Ymin)
            return true;
    }
    if(rect.Xmin <= this.Xmin && rect.Xmax >= this.Xmin)
    {
        if(this.Ymax >= rect.Ymax && this.Ymin <= rect.Ymax)
            return true;
        if(this.Ymax >= rect.Ymin && this.Ymin <= rect.Ymin)
            return true;
    }
    if(rect.Ymin <= this.Ymax && rect.Ymax >= this.Ymax)
    {
        if(this.Xmax >= rect.Xmax && this.Xmin <= rect.Xmax)
            return true;
        if(this.Xmax >= rect.Xmin && this.Xmin <= rect.Xmin)
            return true;
    }
    if(rect.Ymin <= this.Ymin && rect.Ymax >= this.Ymin)
    {
        if(this.Xmax >= rect.Xmax && this.Xmin <= rect.Xmax)
            return true;
        if(this.Xmax >= rect.Xmin && this.Xmin <= rect.Xmin)
            return true;
    }
    // omission of what follows
}
```

#### 4.2.3.2. Point-in-Polygon Operation

Sedgewick’s principle uses a line segment from the test point to a point which is

guaranteed to be outside the polygon. According to his test, if the number of lines from the polygon that it crosses is odd, the point must be inside, otherwise outside.<sup>95</sup> In [Figure 35], point A has 2 intersections, point B has 0, point C has 3, point D has 1, and point E has 4. Therefore, we can determine point C and D are inside the polygon. Using this principle, we implemented the method “IsPointInPolygon” [Code 12] whose algorithm is borrowed from NCGIA Core Curriculum<sup>96</sup>.



[Figure 35] Sedgewick's Test for Point-in-Polygon

[Code 12] Method for Point-in-Polygon Operation

```
public bool IsPointInPolygon(ClapPoint p)
{
    int ni = 1;
    double a, b, yi;
    for(int i = 0; i < _element.Length; i++)
    {
        for(int j = 0; j < _element[i].Length-1; j++)
        {
            if(_element[i][j+1].X != _element[i][j].X)
            {
                if(((_element[i][j+1].X - p.X) * (p.X - _element[i][j].X) >= 0)
                {
                    if((_element[i][j+1].X != p.X) || (_element[i][j].X <= p.X))
                    {
                        if((_element[i][j].X != p.X) || (_element[i][j+1].X >= p.X))
                        {

```

<sup>95</sup> <http://www.informatik.uni-oldenburg.de/~trebla/graphic>

<sup>96</sup> <http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u33.html#SEC33.3>

```

        b = (_element[i][j+1].Y - _element[i][j].Y) / (_element[i][j+1].X -
_element[i][j].X);
        a = _element[i][j].Y - b * _element[i][j].X;
        yi = a + b * p.X;
        if(yi > p.Y)
            ni *= -1;
    }
}
}
}
}
}
if(ni == -1)
    return true;
return false;
}

```

### 4.2.3.3. Centroid Operation

We implemented centroid operation using the algorithm of NGCIA Core Curriculum<sup>97</sup> [Formula 6]. Getting a centroid point requires the area of the polygon, and [Code 13] and [Code 14] show the Properties for getting centroid point and polygon area, respectively.

[Formula 6] Polygon Centroid Algorithm

$$X = \frac{\sum (y_i - y_{i+1})(x_i^2 + x_i x_{i+1} + x_{i+1}^2)}{6A}$$

$$Y = \frac{\sum (x_{i+1} - x_i)(y_i^2 + y_i y_{i+1} + y_{i+1}^2)}{6A}$$

$$A = \frac{\sum (x_{i+1} - x_i)(y_{i+1} + y_i)}{2}$$

<sup>97</sup> <http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u33.html#SEC33.4>

[Code 13] Property for Centroid Operation

```
public ClapPoint Centroid
{
    get
    {
        double centX = 0, centY = 0;
        for(int i = 0; i < _element.Length; i++)
        {
            for(int j = 0; j < _element[i].Length-1; j++)
            {
                centX += (_element[i][j].Y - _element[i][j+1].Y) * (Math.Pow(_element[i][j].X, 2) +
                    (_element[i][j].X) * (_element[i][j+1].X) + Math.Pow(_element[i][j+1].X,
                    2));
                centY += (_element[i][j+1].X - _element[i][j].X) * (Math.Pow(_element[i][j].Y, 2) +
                    (_element[i][j].Y) * (_element[i][j+1].Y) + Math.Pow(_element[i][j+1].Y,
                    2));
            }
        }
        centX /= (6.0 * Area);
        centY /= (6.0 * Area);
        return new ClapPoint(centX, centY);
    }
}
```

[Code 14] Property for Getting Area of Polygon

```
public double Area
{
    get
    {
        double sum = 0;
        for(int i = 0; i < _segment.Length; i++)
            for(int j = 0; j < _segment[i].Length-1; j++)
                sum += (_segment[i][j+1].X - _segment[i][j].X) * (_segment[i][j+1].Y +
                _segment[i][j].Y);
        sum /= 2.0;
        return sum;
    }
}
```

#### 4.2.4. API Development

As a result of the functionality implementation, we provide API functions covering viewing/controlling a map, advanced rendering, and topological operation. The typical public methods and properties we implemented for

application development are presented in [Table 29] to [Table 33].

[Table 29] Typical API Methods in ClapMap Class

Method	Parameter	Return	Description
AddLayer	string filename, string filetype		Parses a map file, constructs all the necessary objects, and add a ClapMapLayer instance to this class object
RemoveLayer	int index		Removes a ClapMapLayer instance at the position of {index} from this class object
DrawMap			Draws a map with all ClapMapLayer instances added to this class with default setting
DrawMap	ClapPalette cp		Draws a map with all ClapMapLayer instances added to this class with the setting of {cp}
GetFieldNames	int index	string[]	Gets all field names of the layer of {index}
GetFieldValues	int index	string[]	Gets all values of the field of {index}
GetRecord	int index	string[]	Gets all values of the record of {index}
DrawMapLabel	int layerindex, int fieldindex		Draws map labels with the field of {fieldindex} at the layer of {layerindex}
ZoomIn	float zoomfactor		Zooms in current map with the {zoomfactor}
ZoomOut	float zoomfactor		Zooms out current map with the {zoomfactor}
ZoomToExtent			Draws a map with the initial scale
Geog2Screen	ClapPoint xy	Point	Transforms geographic coordinate {xy} to screen coordinate
Screen2Geog	Point xy	ClapPoint	Transforms screen coordinate {xy} to geographic coordinate
MoveE	int pixeldistance		Moves a map {distance} east
MoveW	int pixeldistance		Moves a map {distance} west
MoveS	int pixeldistance		Moves a map {distance} south
MoveN	int pixeldistance		Moves a map {distance} north
MoveNE	int pixeldistance		Moves a map {distance} northeast
MoveNW	int pixeldistance		Moves a map {distance} northwest
MoveSE	int pixeldistance		Moves a map {distance} southeast
MoveSW	int pixeldistance		Moves a map {distance} southwest
MoveCenterTo	Point xy		Moves a map center to the screen coordinate of {xy}

[Table 30] Typical API Methods in ClapMapRepresenter Class

Method	Parameter	Return	Description
PrepareData	double[] values, int num		Performs sorting the {values} and prepares array for classification according to number of class {num}
SetEqualInterval			Performs classification using equal interval
SetQuantile			Performs classification using quantile
SetStandardDev	float interval		Performs classification using standard deviation with the {interval}
SetColorRamp			Arranges color ramp according to classification array and start/end color
GetRandomColor		Color[]	Gets an array of Color for random color ramp
SetSymbol	Bitmap bmp		Sets symbol for point layer
SetSymbol	Bitmap bmp, int initsize		Sets resizable symbol for point or polygon layer with the initial size {initsize}
SetMarker	ClapMarker cm		Sets marker for point layer
SetMarker	ClapMarker cm, int initsize		Sets resizable marker for point or polygon layer with the initial size {initsize}

[Table 31] Typical Properties in ClapMapRepresenter Class

Property	Type	Description
NumberOfClass	int	Number of class for classification
StartColor	Color	Start color for classification
EndColor	Color	End color for classification
ColorSet	Brush[]	Array of Brush for color ramp

[Table 32] Typical API Methods in ClapRect and ClapFeature Class

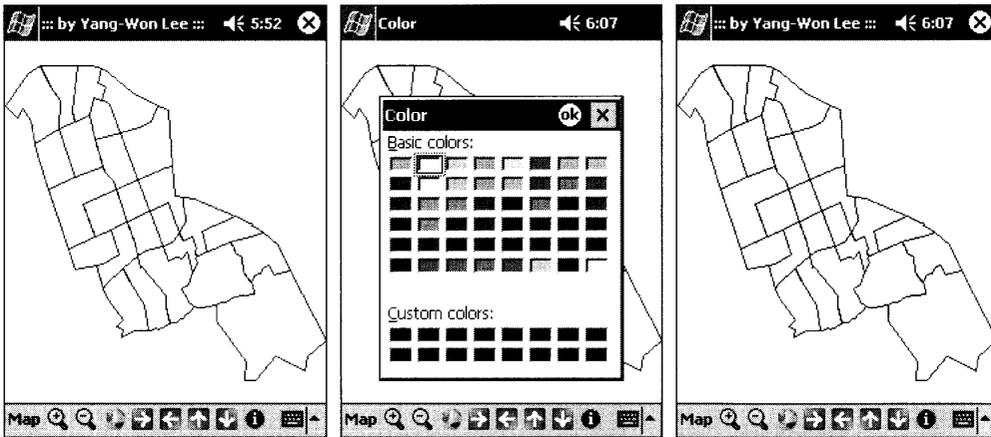
Method	Parameter	Return	Description
Contains	ClapRect cr	bool	Determines this rectangle contains the rectangle {cr} or not
IsPointInPolygon	ClapPoint cp	bool	Determines the point {cp} is in this polygon feature

[Table 33] Typical Properties in ClapFeature Class

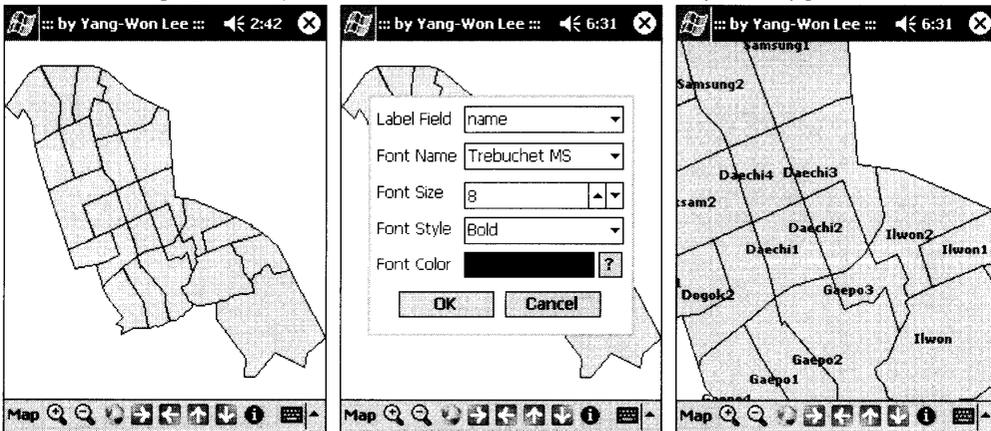
Property	Type	Description
Centroid	ClapPoint	Gets centroid point of this polygon feature
Area	double	Gets area of this polygon feature

## 4.2.5. Feasibility Test

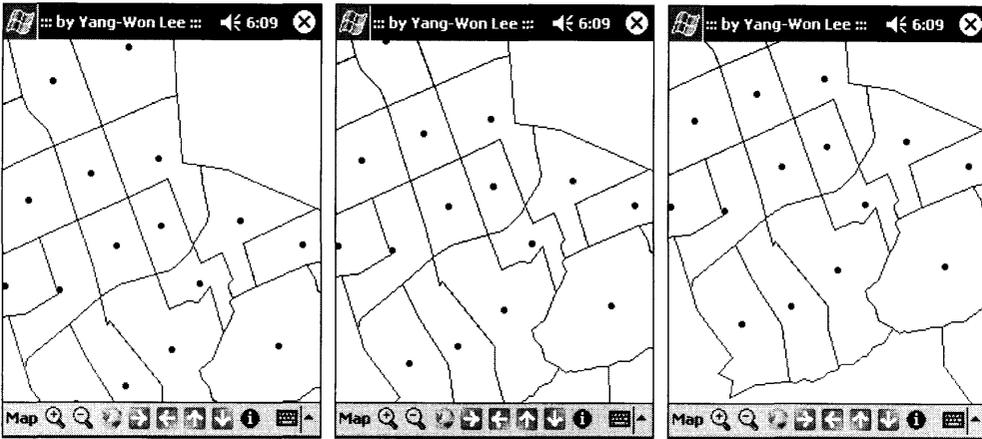
To verify our API functions, we implemented feasibility testing applications of basic visualization and advanced rendering illustrated in [Figure 36] to [Figure 44].



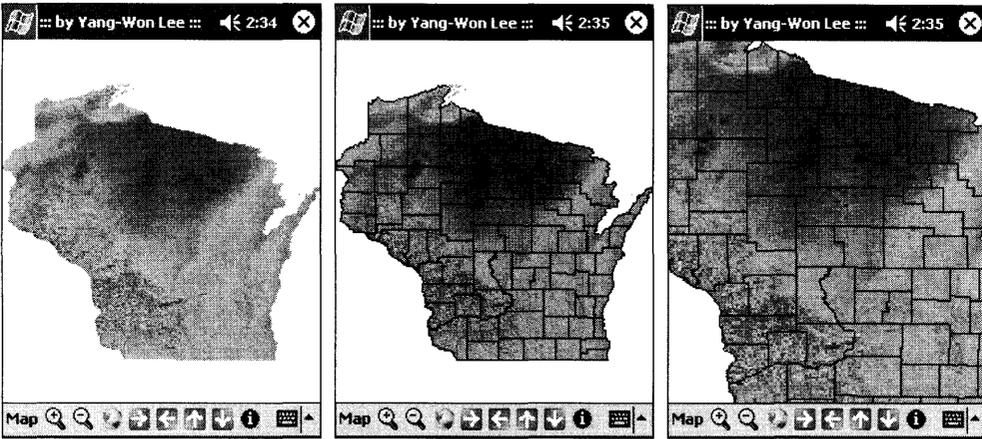
[Figure 36] Testing Screenshot of Color Setting of Polygon



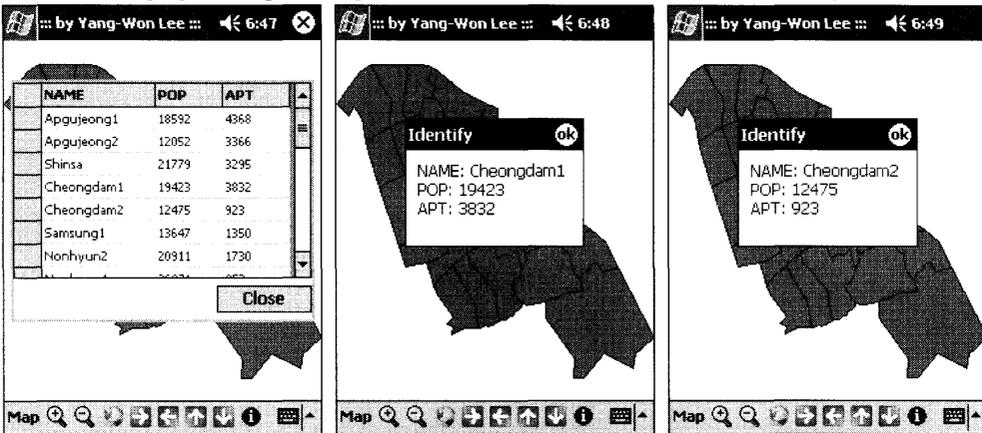
[Figure 37] Testing Screenshot of Labeling a Map



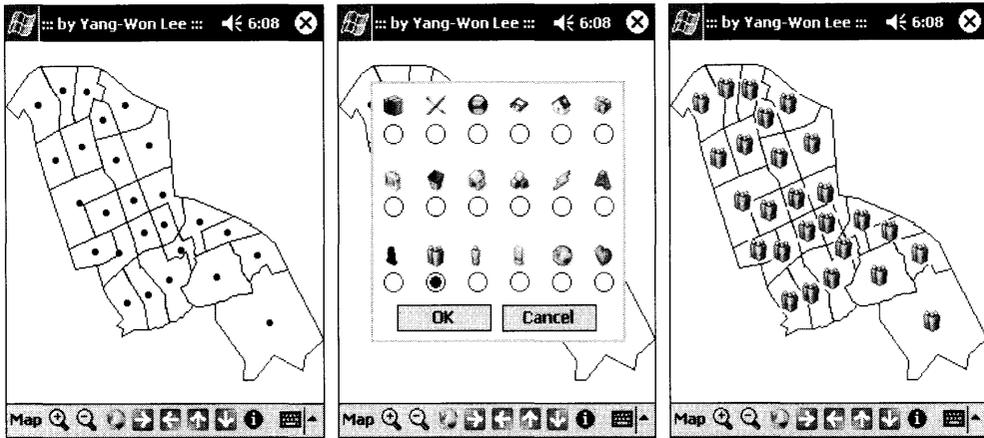
[Figure 38] Testing Screenshot of Zooming and Panning



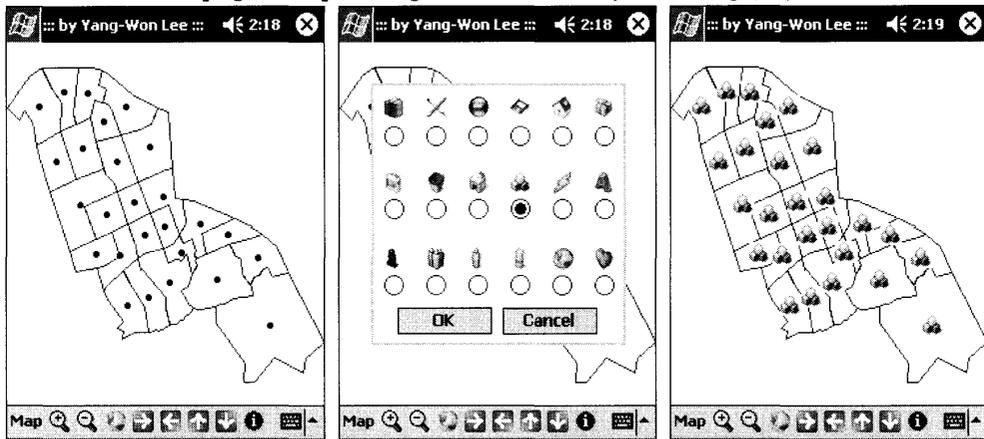
[Figure 39] Testing Screenshot of Vector-on-Raster Overlay



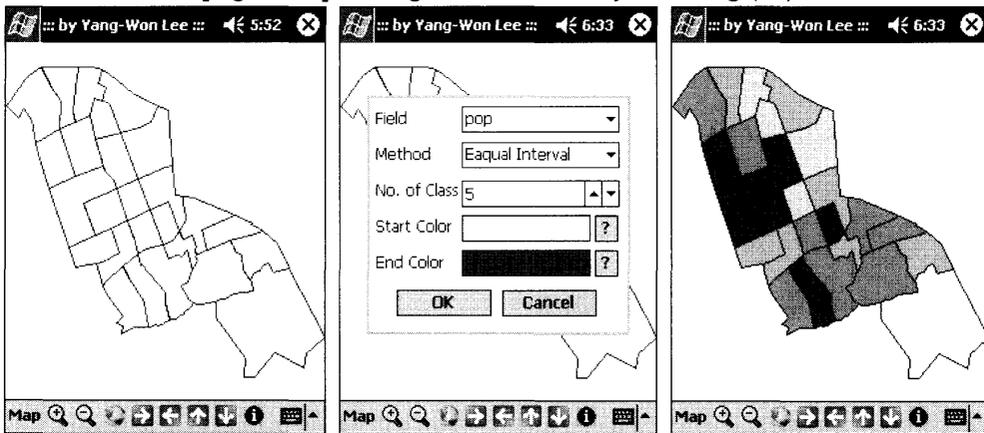
[Figure 40] Testing Screenshot of Attribute Query



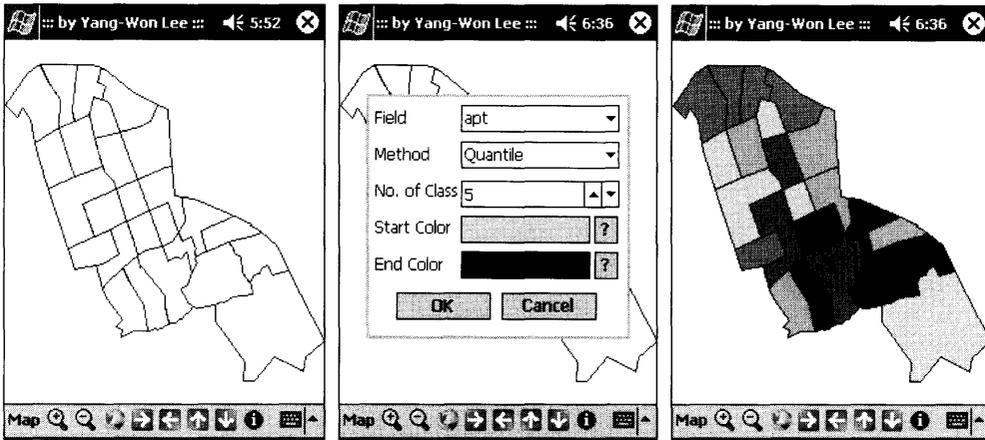
[Figure 41] Testing Screenshot of Symbolizing ( I )



[Figure 42] Testing Screenshot of Symbolizing ( II )



[Figure 43] Testing Screenshot of Choropleth Map ( I )



[Figure 44] Testing Screenshot of Choropleth Map ( II )

### 4.3. Mobile Internet Mapping

Introducing a middleware to the mobile internet mapping contributes to the distribution of loads on the network and the increase of object-orientedness. Our architecture of the mobile internet mapping is based on XML Web Services, and includes two middleware: location middleware for geocoding/reverse-geocoding, and mapping middleware for delivering a map by the communication with map server. As shown in the above benchmarking studies, XML Web Services are considered to be an efficient way to constructing a middleware for internet applications.

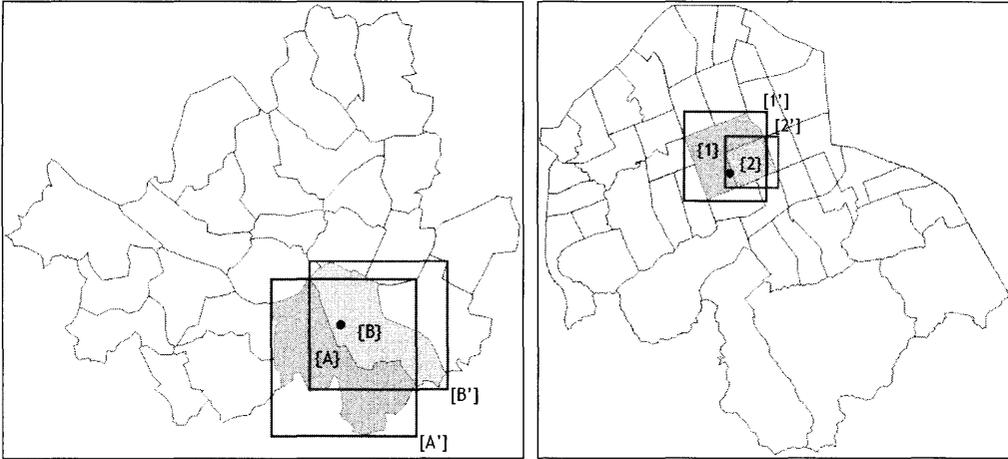
#### 4.3.1. Location Middleware

One of the most important functionalities of location middleware is geocoding/reverse-geocoding; for it interconnects the location information of a mobile user and geographic information according to his/her location.

Geocoding is finding a representative {x, y} coordinate according to a specific address, and reverse-geocoding is finding an address according to a specific {x, y} coordinate (Niedzwiadek, 2002).

In case of the geocoding, the representative point of a polygon feature according to a specific address can be easily derived by centroid operation. In case of reverse-geocoding, two-step procedure is necessary. At first, we should select one or more polygon features whose MBR contains a specific point. And then, we should select the only polygon feature which contains the point using Point-in-Polygon operation. This two-step procedure is much faster than the Point-in-Polygon operation over the whole polygon features.

Our principle for reverse-geocoding is described in [Figure 45] and [Figure 46]. The first step is MBR-based hierarchical operation for getting one or more polygon features whose MBR contains a specific point. At the higher level, we got two districts {A} and {B} whose MBR is drawn as {A'} and {B'} respectively. At the lower level, we got two sub-districts {1} and {2} whose MBR is drawn as {1'} and {2'} respectively. This hierarchical operation continues to the lowest level (level n). The second step is Point-in-Polygon operation on the lowest level (e.g., land parcel map) for getting the only polygon feature which contains a specific point drawn in red. Once the polygon feature according to the specific coordinate is acquired, any attribute value of the polygon is accessible.



[Figure 45] MBR-Based Hierarchical Operation for Reverse-Geocoding



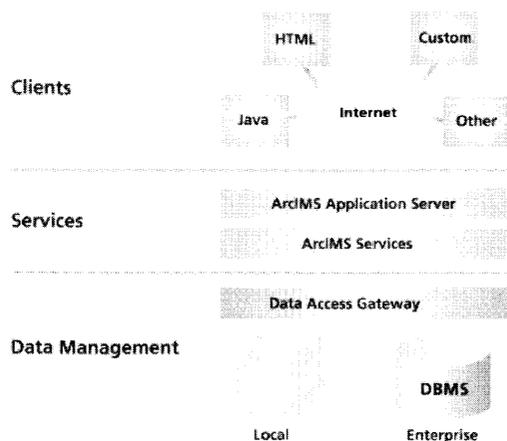
[Figure 46] Point-in-Polygon Operation for Reverse-Geocoding

## 4.3.2. Mapping Middleware

One of the most important functionalities of our Mobile Mapping Middleware is the interoperability with generic map servers. We selected ESRI ArcIMS, UMN MapServer, and BBN OpenMap as target map servers. For implementing the interoperability, we scrutinized the communication structure of each map server. After defining the syntax of the request/response of each map server, we composed web methods dealing with the request/response for the delivery of map and attribute. The “WebMethod” is a core part of XML Web Services as a bridge between map server and client.

### 4.3.2.1. Interoperability with ESRI ArcIMS

The ArcIMS<sup>98</sup> framework consists of data management, services, and clients [Figure 47]. Local files and DBMS are supported for the “Data Management”; ArcXML is the markup language for the “Map Services”; HTML, Java, and others are available for the web “Clients.”



[Figure 47] Framework of ArcIMS

<sup>98</sup> <http://esri.com/software/arcims/architecture.html>

Every request and response is transmitted through the ArcXML (an extension of XML). The example of the request is map extent, rendering color, or zooming factor; the example of the response is physical path or URL of an output image. [Code 15] shows a sample ArcXML request which is composed by an application developer, and [Code 16] shows sample ArcXML response which is transmitted from the ArcIMS to client-side.

#### [Code 15] Example of ArcXML Request

```
<?xml version="1.0"?>
<ARCXML version="1.0">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-88" miny="30" maxx="-67" maxy="50" />
        <IMAGESIZE width="500" height="350" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" >
            <SIMPLERENDERER>
              <SIMPLEPOLYGONSYPMBOL filltype="solid" fillcolor="255,255,0" />
            </SIMPLERENDERER>
          </LAYERDEF>
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

#### [Code 16] Example of ArcXML Response

```
<?xml version="1.0"?>
<ARCXML version="1.0">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-87.5" miny="30.0" maxx="-59.5" maxy="50.0" />
      <OUTPUT file="C:\ArcIMS\output\us_image_MYCOMPUTER2953026.jpg"
url="http://mycomputer.domain.com/output/us_image_MYCOMPUTER2953026.jpg" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

The most basic WebMethod for communicating with ArcIMS is described in [Code 17]. This method accepts the parameters of ArcXML sentence, server name, and map service name, and returns server's response string which includes the URL of output image map.

[Code 17] Basic WebMethod for ESRI ArcIMS

```
[WebMethod]
public string GetMap(string arcXML, string server, string mapService)
{
    string result;
    string theURL = "http://" + server +
"/servlet/com.esri.esrimap.Esrimap?ServiceName=" + mapService;
    HttpRequest HttpWReq = (HttpRequest)WebRequest.Create(theURL);
    HttpWReq.Method = "POST";
    HttpWReq.ContentLength = arcXML.Length;
    StreamWriter pWriter = new StreamWriter(HttpWReq.GetRequestStream());
    pWriter.Write(arcXML);
    pWriter.Close();
    try
    {
        HttpResponseMessage HttpWResp =
(HttpWebResponse)HttpWReq.GetResponse();
        StreamReader sr = new StreamReader(HttpWResp.GetResponseStream());
        StringReader stgr = new StringReader(sr.ReadToEnd());
        result = stgr.ReadToEnd();
    }
    catch
    {
        result = null;
    }
    return result;
}
```

#### 4.3.2.2. Interoperability with UMN MapServer

UMN MapServer<sup>99</sup> provides several CGI (Common Gateway Interface) parameters for map browsing and query<sup>100</sup>. Developers send these CGI parameter values to

<sup>99</sup> The MapServer was originally developed by UMN (University of Minnesota) ForNet project in cooperation with NASA and the Minnesota DNR (Department of Natural Resources) (<http://mapserver.gis.umn.edu>).

<sup>100</sup> <http://mapserver.gis.umn.edu/doc40/cgi-reference.html>

the server through HTTP/GET or HTTP/POST method as a request, and get the response from the server to visualize a map on the client-side. [Code 18] shows a sample configuration file dealing with the initial setting such as map extent, data storage path, initial web document, etc. In addition, [Table 34] shows an example of the request parameters transmitted to server.

[Code 18] Example of MapServer Configuration File

```
# begin of map file
NAME DEMO
#STATUS ON
SIZE 417 514
EXTENT 825697.96 482183.57 856957.96 520703.57
SHAPEPATH "data"
WEB
  TEMPLATE demo.html
END
LAYER
  NAME "DEM"
  DATA "dem.tif"
  TYPE RASTER
  #STATUS ON
END
END
# end of map file
```

[Table 34] Example of MapServer CGI Request

CGI Parameter	Parameter Value
program	/cgi-bin/mapserv.exe
map	D:\\phpdev\\www\\public\\testraster\\demo.map&layer=DEM
map_web_imagepath	D:\\phpdev\\www\\public\\tmp\\
map_web_imageurl	/public/tmp/

[Code 19] shows the most basic WebMethod for communicating with UMN MapServer. This method accepts the parameters of server, port, map, layer, image path, and image URL, and it returns the response string which includes the URL of output image map.

### [Code 19] Basic WebMethod for UMN MapServer

```
[WebMethod]
public string GetMapResponse(string server, string port, string map, string layer, string
map_web_imagepath, string map_web_imageurl)
{
    string result;
    StringBuilder sb = new StringBuilder();
    sb.Append("http://" + server + ":" + port + "/cgi-bin/mapserv.exe");
    sb.Append("?program=/cgi-bin/mapserv.exe");
    sb.Append("&map=").Append(map);
    sb.Append("&layer=").Append(layer);
    sb.Append("&map_web_imagepath=").Append(map_web_imagepath);
    sb.Append("&map_web_imageurl=").Append(map_web_imageurl);
    string url = sb.ToString();
    try
    {
        WebRequest req = WebRequest.Create(url);
        WebResponse res = req.GetResponse();
        Stream ReceiveStream = res.GetResponseStream();
        Encoding encode = Encoding.Default;
        StreamReader sr = new StreamReader(ReceiveStream, encode);
        StringReader stgr = new StringReader(sr.ReadToEnd());
        result = stgr.ReadToEnd();
    }
    catch
    {
        result = null;
    }
    return result;
}
```

#### 4.3.2.3. Interoperability with BBN OpenMap

BBN Technologies OpenMap<sup>101</sup> has a Java-based web map server using HTTP, and [Table 35] shows an example of the request transmitted through HTTP/GET or HTTP/POST.

---

<sup>101</sup> BBN Technologies OpenMap was used in a number of DARPA (Defense Advanced Research Projects Agency) and military sponsored programs in the United States (<http://openmap.bbn.com>).

[Table 35] Example of OpenMap HTTP Request

Parameter	Parameter Value
REQUEST	MAP
LAT	30
LON	-70
SCALE	250000000
PROJTYPE	Cadrg
HEIGHT	350
WIDTH	350
BGCOLOR	FFFFFF
LAYERS	shapePolitical

[Code 20] shows the most basic WebMethod for communicating with BBN OpenMap. This method accepts the parameters of server, port, latitude, longitude, scale, projection type, image height, image width, background color, and map layer, and it returns the response in the form of byte array which includes the output image map. To show a map on the client-side, developers can create a Bitmap object from the byte array via memory stream.

[Code 20] Basic WebMethod for BBN OpenMap

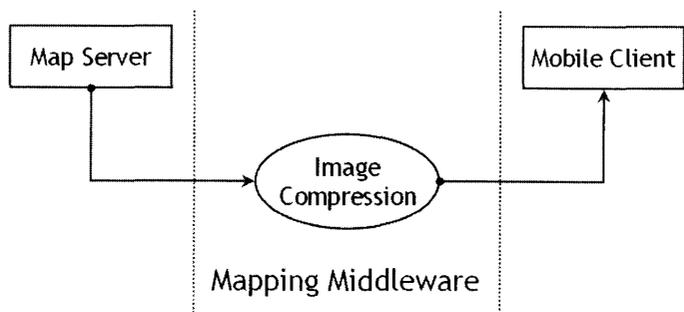
```
[WebMethod]
public byte[] GetMapStream(string server, string port, string lat, string lon, string scale, string
projtype, string height, string width, string bgcolor, string layers)
{
    byte[] result;
    string url = "http://" + server + ":" + port + "/openmap?REQUEST=MAP";
    url += "&LAT=" + lat;
    url += "&LON=" + lon;
    url += "&SCALE=" + scale;
    url += "&PROJTYPE=" + projtype;
    url += "&HEIGHT=" + height;
    url += "&WIDTH=" + width;
    url += "&BGCOLOR=" + bgcolor;
    url += "&LAYERS=" + layers;
    try
    {
        WebRequest req = WebRequest.Create(url);
        WebResponse res = req.GetResponse();
        Stream recStream = res.GetResponseStream();
        Image img = new Bitmap(recStream);
        MemoryStream memStream = new MemoryStream();
```

```
        img.Save(memStream, ImageFormat.Gif);
        result = memStream.GetBuffer();
    }
    catch
    {
        result = null;
    }
    return result;
}
```

#### 4.3.2.4. Image Compression

As the vector simplification is sometimes necessary for faster map visualization in the mobile embedded environment, image compression is sometimes necessary in the mobile internet environment because of the low bandwidth of wireless internet. Image compression is a part of image encoding method for decreasing the file size of an image with little loss of quality.

The output map created by map server is generally in the form of image file, and the client-side refers to the image URL for map display. The image compression for this output image can be performed by intercepting the image file on the way from server to client [Figure 48]. This image compression module is located in our mapping middleware because generic map servers currently do not provide the options for image compression.



[Figure 48] Workflow of Image Compression in the Mapping Middleware

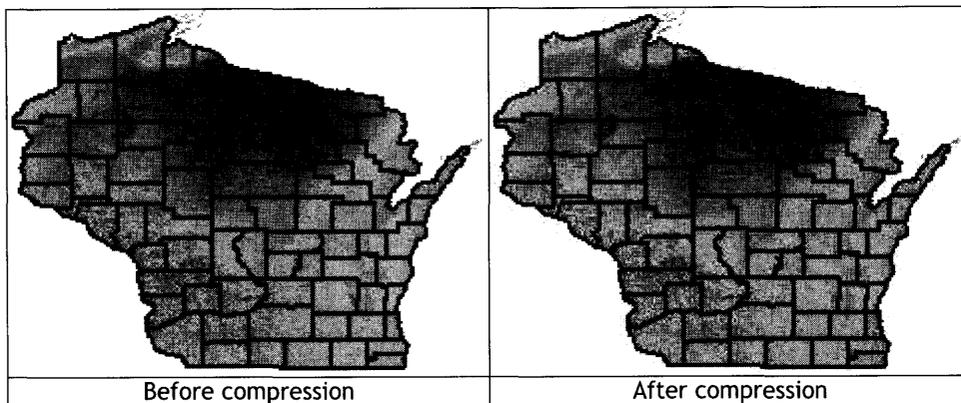
We implemented the image compression functionality using `Encoder.Quality` field in `System.Drawing.Imaging` package of the .NET Framework. This encoding field adopts the average method for image compression which uses the average value of each pixel block. The image compression module performs real-time compression of an image map, and the core part of this module is described in [Code 21]. [Code 22] is the utility method for Codec information. We provide “image/jpeg” and “image/gif” as developer’s encoding option. [Figure 49] is an example of the image map compressed by our mapping middleware connected to ArcIMS. The file size of compressed map varies (10-50% of original file) depending on the ratio option. This compression process is not mandatory but optional in the application development.

[Code 21] Method for Image Compression on the Web

```
private string GetCompressedMap(string originalPath, long ratio, string mimeType)
{
    // omission of what precedes
    theBitmap = new Bitmap(originalPath);
    theImageCodecInfo = GetEncoderInfo(mimeType);
    theEncoder = Imaging.Encoder.Quality;
    theEncoderParameter = new EncoderParameter(theEncoder, ratio);
    theEncoderParameters = new EncoderParameters(1);
    theEncoderParameters.Param[0] = theEncoderParameter;
    theBitmap.Save(replacedPath, theImageCodecInfo, theEncoderParameters);
    // omission of what follows
}
```

[Code 22] Method for Codec Information

```
private ImageCodecInfo GetEncoderInfo(string mimeType)
{
    ImageCodecInfo[] encoders;
    encoders = ImageCodecInfo.GetImageEncoders();
    for(int j = 0; j < encoders.Length; ++j)
        if(encoders[j].MimeType == mimeType)
            return encoders[j];
    return null;
}
```



[Figure 49] Result Comparison of Image Compression in the Mapping Middleware

### 4.3.3. API Development

As a result of the functionality implementation, we provide API functions in the form of WebMethod covering basic visualization and advanced rendering. Using these WebMethod, application developers can build Windows client using .NET Compact Framework and web client using the ASP.NET Mobile Controls. In case of the WebMethod for ArcIMS, every return value is the type of object array which has 6 elements [Table 36], and the typical WebMethod for ArcIMS are described in [Table 37] [Table 38].

[Table 36] Description of Object Array for ArcIMS WebMethod

Index	Type	Content
0	string	Previous ArcXML request string
1	double	Xmin of current extent
2	double	Ymin of current extent
3	double	Xmax of current extent
4	double	Ymax of current extent
5	string	URL of output image file

[Table 37] Typical WebMethod for Basic Visualization

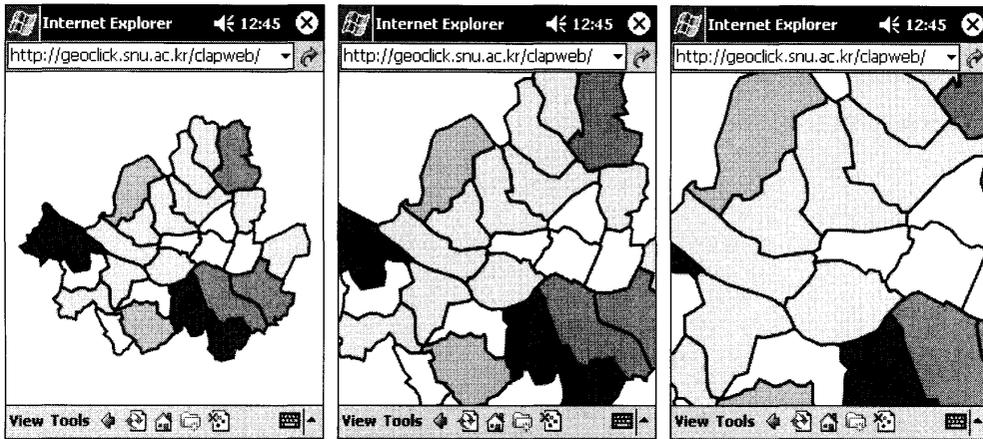
Method	Parameter	Description
GetDefaultMap	string server, string mapSvc	Gets a default map and its frame Info without image compression
GetDefaultMap2	string server, string mapSvc	Gets a default map and its frame Info with image compression
GetUserMap	string arcXML, string server, string mapSvc	Gets a user-defined map and its frame Info without image compression
GetUserMap2	string arcXML, string server, string mapSvc	Gets a user-defined map and its frame Info with image compression
GetZoomInMap	string arcXML, string server, string mapSvc	Gets a zoomed-in map and its frame Info without image compression
GetZoomInMap2	string prevAXL, string server, string mapSvc, double[] extent	Gets a zoomed-in map and its frame Info with image compression
GetZoomOutMap	string prevAXL, string server, string mapSvc, double[] extent	Gets a zoomed-out map and its frame Info without image compression
GetZoomOutMap2	string prevAXL, string server, string mapSvc, double[] extent	Gets a zoomed-out map and its frame Info with image compression
GetMoveUpMap	string prevAXL, string server, string mapSvc, double[] extent	Gets a moved-up map and its frame Info without image compression
GetMoveUpMap2	string prevAXL, string server, string mapSvc, double[] extent	Gets a moved-up map and its frame Info with image compression
GetMoveDownMap	string prevAXL, string server, string mapSvc, double[] extent	Gets a moved-down map and its frame Info without image compression
GetMoveDownMap2	string prevAXL, string server, string mapSvc, double[] extent	Gets a moved-down map and its frame Info with image compression
GetMoveLeftMap	string prevAXL, string server, string mapSvc, double[] extent	Gets a moved-left map and its frame Info without image compression
GetMoveLeftMap2	string prevAXL, string server, string mapSvc, double[] extent	Gets a moved-left map and its frame Info with image compression
GetMoveRightMap	string prevAXL, string server, string mapSvc, double[] extent	Gets a moved-right map and its frame Info without image compression
GetMoveRightMap2	string prevAXL, string server, string mapSvc, double[] extent	Gets a moved- right map and its frame Info with image compression
GetLabelMap	string server, string mapSvc, int layerID, string labelField	Gets a label map and its frame info without image compression
GetLabelMap2	string server, string mapSvc, int layerID, string labelField	Gets a label map and its frame info with image compression
GetBoundingBoxMap	string server, string mapSvc, double[] box, string labelField	Gets a map and its frame info from user-defined bounding box without image compression
GetBoundingBoxMap2	string server, string mapSvc, double[] box, string labelField	Gets a map and its frame info from user-defined bounding box with image compression

[Table 38] Typical WebMethod for Advanced Rendering

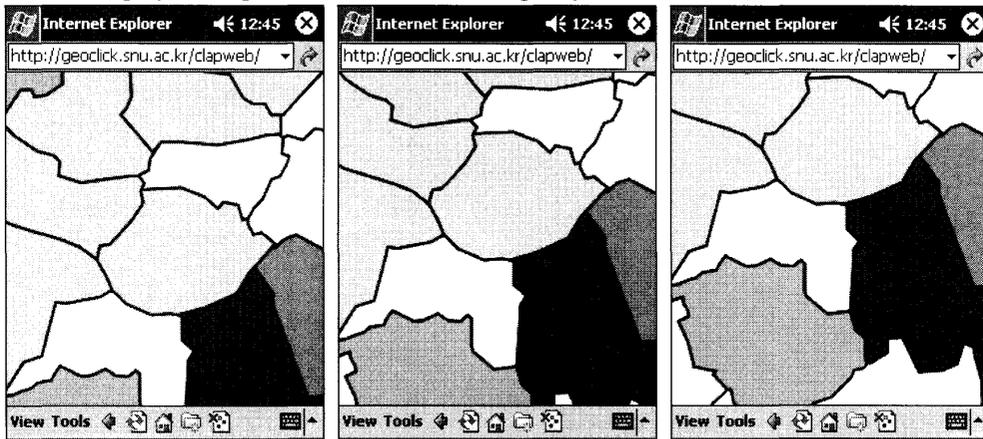
Method	Parameter	Description
GetChoroplethMap	string server, string mapSvc, string lookupField, int method, int numOfClass	Gets a choropleth map and its frame info without image compression
GetChoroplethMap2	string server, string mapSvc, string lookupField, int method, int numOfClass	Gets a choropleth map and its frame info with image compression
GetSymbolMap	string server, string mapSvc, string lookupField, int method, int numOfClass, int symbolIndex	Gets a symbol map and its frame info without image compression
GetSymbolMap2	string server, string mapSvc, string lookupField, int method, int numOfClass, int symbolIndex	Gets a symbol map and its frame info with image compression
GetChartMap	string server, string mapSvc, string lookupField1, string lookupField2, string type, int size, string color1, string color2	Gets a chart map and its frame info without image compression
GetChartMap2	string server, string mapSvc, string lookupField1, string lookupField2, string type, int size, string color1, string color2	Gets a chart map and its frame info with image compression
GetHighlightMap	string server, string mapSvc, string lookupField, string condition	Gets a highlight map and its frame info without image compression
GetHighlightMap2	string server, string mapSvc, string lookupField, string condition	Gets a highlight map and its frame info with image compression

#### 4.3.4. Feasibility Test

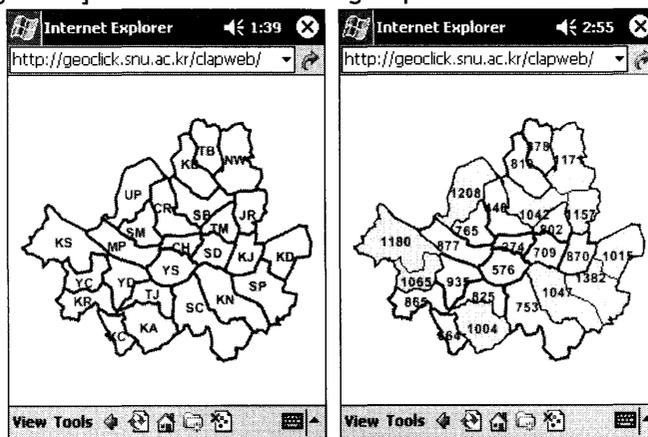
To verify our API functions, we implemented feasibility testing applications of basic visualization and advanced rendering illustrated in [Figure 50] to [Figure 55].



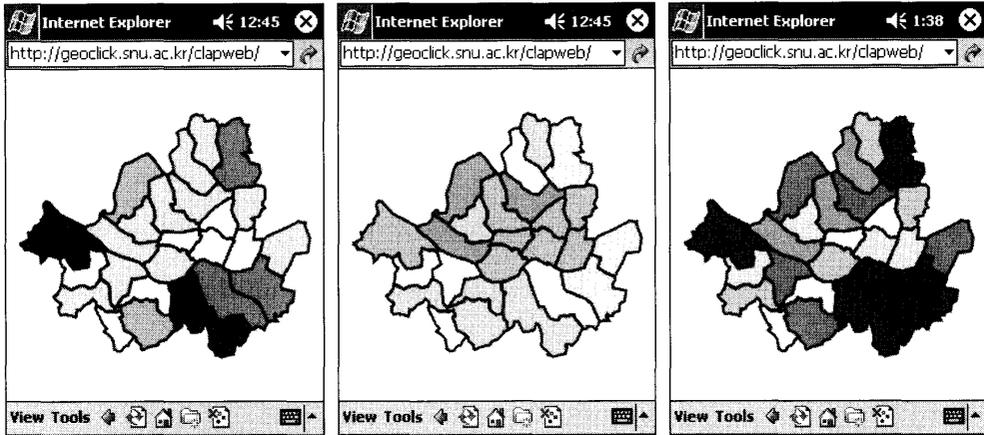
[Figure 50] Screenshot of Zooming Map on Mobile Web Browser



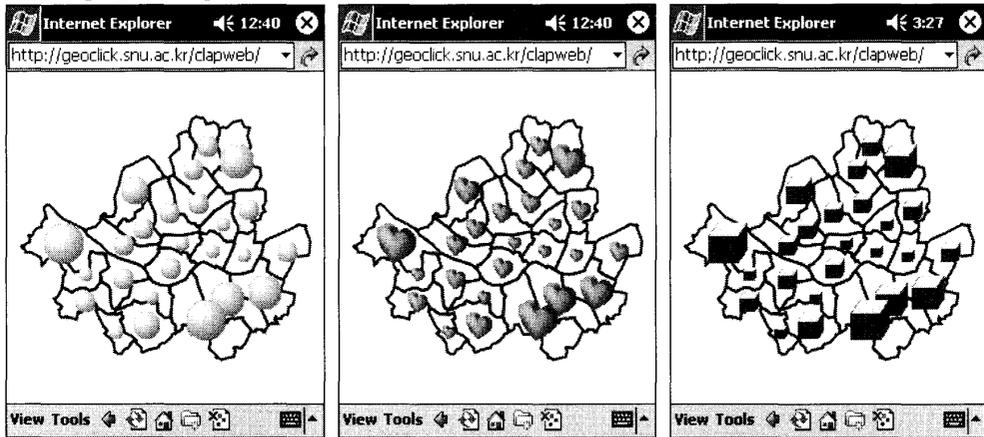
[Figure 51] Screenshot of Panning Map on Mobile Web Browser



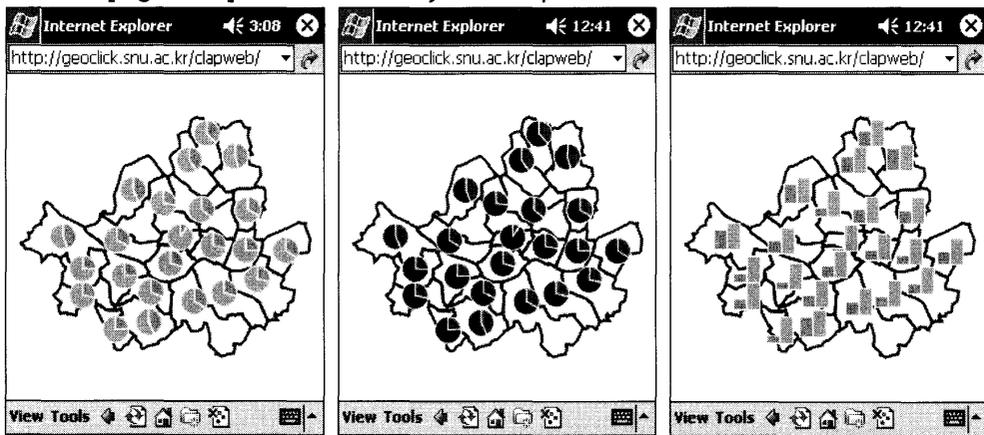
[Figure 52] Screenshot of Labeling and Highlighting Map on Mobile Web Browser



[Figure 53] Screenshot of Choropleth Map on the Mobile Web Browser



[Figure 54] Screenshot of Symbol Map on the Mobile Web Browser

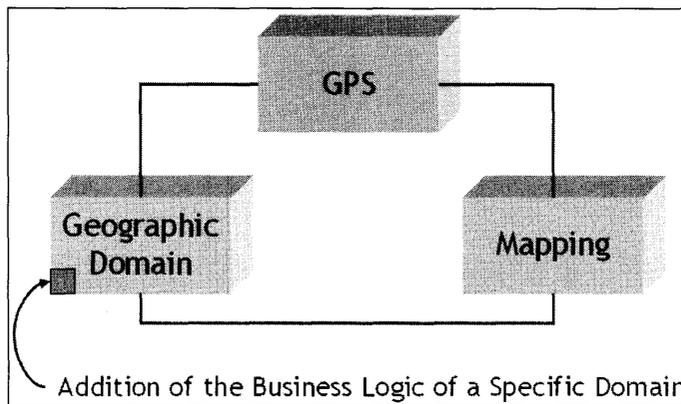


[Figure 55] Screenshot of Chart Map on the Mobile Web Browser

## 5. Application Prototyping

### 5.1. Overview

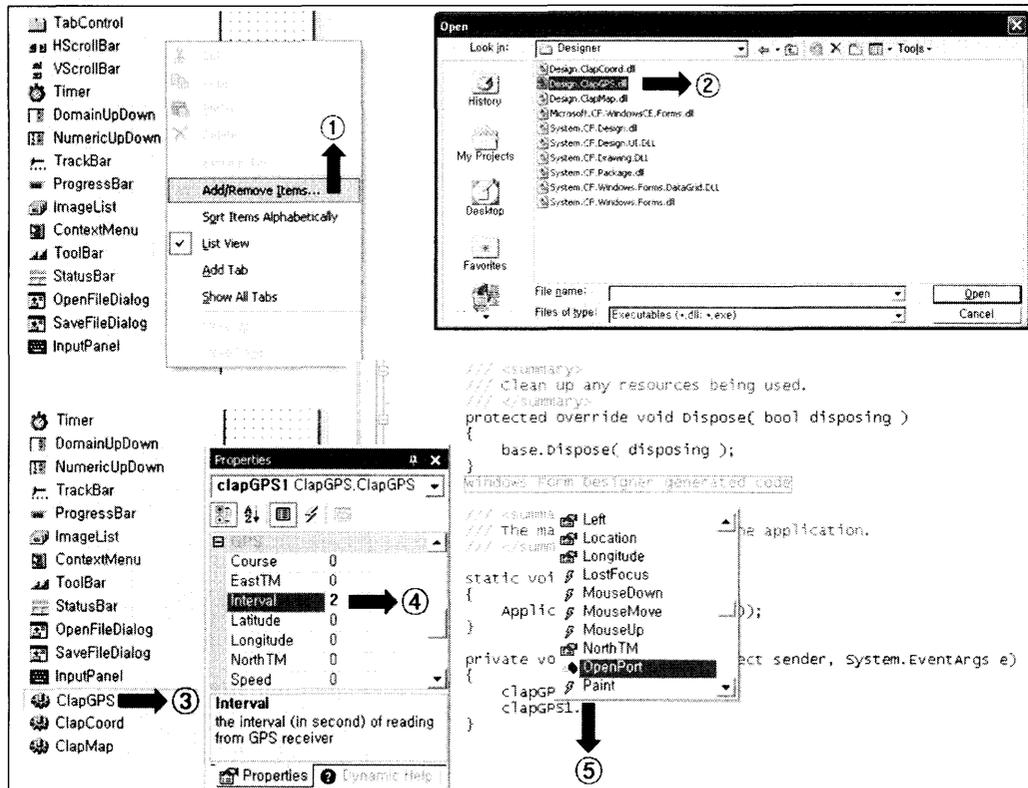
Our geocomputing platform, which is composed of the component libraries for GPS functionality, mobile embedded mapping, and mobile internet mapping built in the .NET environment, supports mobile application development in geographic domains. In the application development process, developers using our geocomputing platform may have to add necessary codes for the business logic of a specific domain for their purpose [Figure 56].



[Figure 56] Geocomputing Platform Extensibility for Geographic Applications

In case of mobile embedded mapping application with GPS support, developers can use the components of “ClapGPS,” “ClapCoord,” and “ClapMap” which manage GPS functionality, coordinate transformation, and embedded mapping, respectively. These components are imported into .NET development tool in the form of DLL. [Figure 57] illustrates the brief process of using these

components in the application of location-aware mobile mapping.



[Figure 57] Brief Process of Using Components of the Geocomputing Platform for Location-aware Mobile Mapping Application in the Embedded Environment

In the .NET development tool, application developer can add a component in the Toolbox pane (1) by designating a DLL file (e.g., ClapGPS.dll) according to the component (2). The Toolbox pane is updated with the new component (3), and application developer can set a value of the component Property (e.g., Interval), if needed (4). In the Code Editor pane, application developer can write down necessary codes with the automatic popup help for Methods and

Properties of the component<sup>102</sup> (⑤).

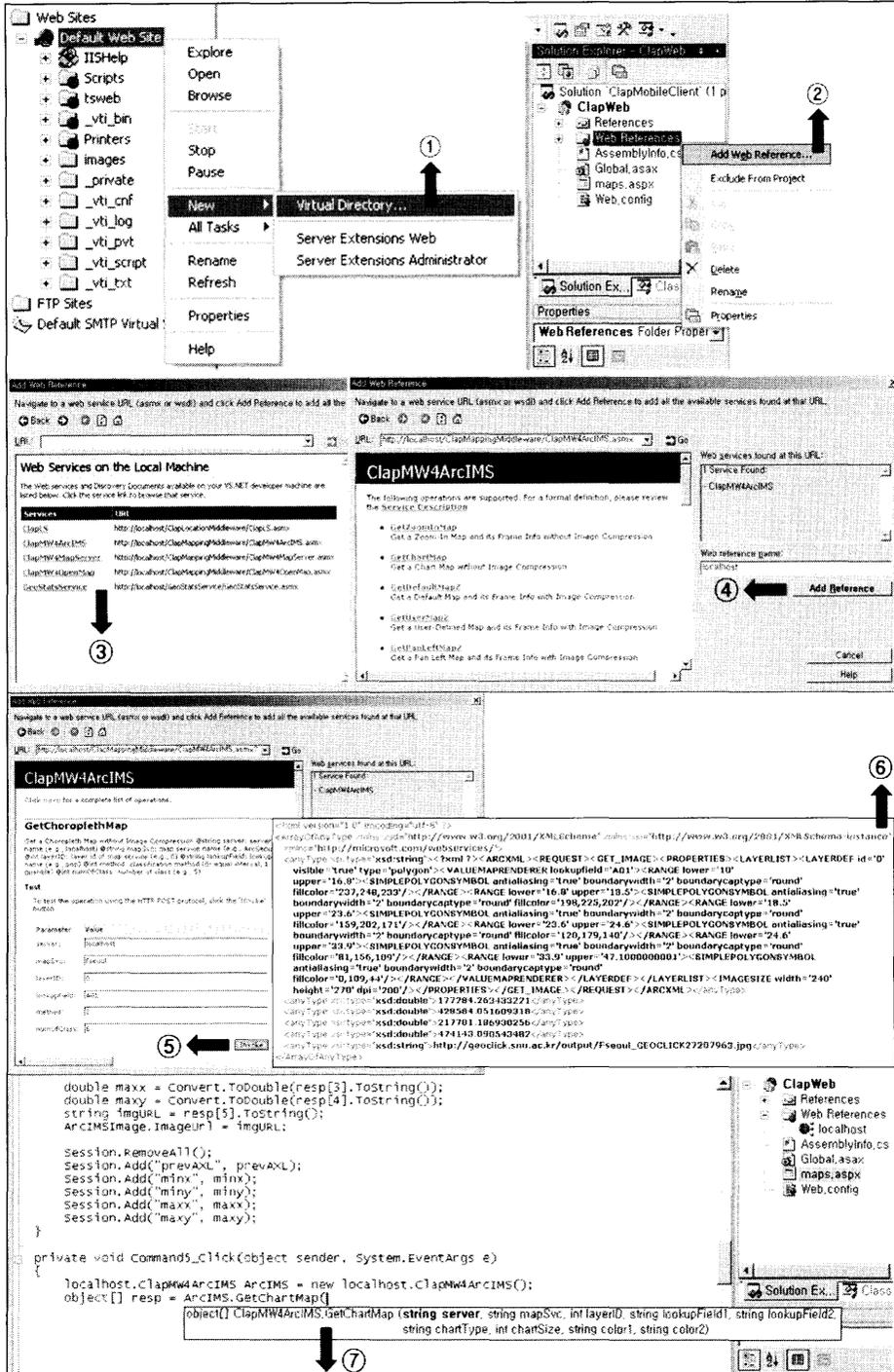
In case of mobile internet mapping application without GPS support, developers can connect to the mapping middleware through HTTP from the mobile devices such as Windows CE-based PDA/cell phone and Java-based PDA/cell phone which support XML Web Services.

In case of LBS applications with GPS support, application developers can use both the GPS functionality and the location/mapping middleware in the .NET Compact Framework environment. It allows the access to embedded functionality of GPS and internet functionality of middleware as well. The usage of the embedded functionality of GPS is the same as in the case of mobile embedded mapping; the usage of the internet functionality of middleware is made possible by using WebRequest/WebResponse class of System.Net package in the .NET Compact Framework.

The brief process of using the mapping middleware and location middleware of our geocomputing platform is described in [Figure 58]. These middleware work with any mobile environment which is compatible with HTTP, and the example usage as follows was experimented in the .NET development tool.

---

<sup>102</sup> Including the API methods and properties described in 4.1.4 and 4.2.4



[Figure 58] Brief Process of Using Components of the Geocomputing Platform for Mobile Internet Mapping and LBS Applications

The mapping middleware of our geocomputing platform is composed of the XML Web Service such as “ClapMW4ArcIMS,” “ClapMW4MapServer,” and “ClapMW4OpenMap”; the location middleware is composed of “ClapLS.” Server-side configuration for the components of mapping middleware and location middleware includes the IIS (Internet Information Server) setting of Virtual Directory creation for the XML Web Services (①). Virtual Directory allows the access to the middleware from client-side via the internet.

In the .NET development tool, application developer can add Web Reference to these middleware (②). If application developer chooses a XML Web Service middleware (e.g., “ClapMW4ArcIMS”) for internet mobile mapping (③), the WebMethod list of the XML Web Service shows up with descriptions. After checking the WebMethod list, an application developer confirms the Add Reference (④). Each hyperlink of WebMethod list guides the application developer to the WebMethod testing environment with usage instruction (⑤). The testing invocation of a WebMethod with appropriate parameter values returns the result XML data composed of the image map information, which is necessary for the mapping on mobile devices (⑥). Application developer can write down necessary codes with the automatic popup help for the WebMethod of the middleware (⑦). In addition, the usage of location middleware component is the same as this.

As seen in the process of using the components both in the embedded and internet environment, developers can compose the applications of LBS and location-aware mobile mapping with the convenience of PNP (Plug and Play): for the components save a lot of efforts the developers otherwise have to

make to compose the core logic such as GPS functionality with coordinate transformation, embedded mapping, internet mapping, and location information processing.

To verify the feasibility of our geocomputing platform, we prototyped two applications: field data collection and cadastral information service. The prototype of field data collection runs in the mobile embedded environment, being with the GPS functionality and user's own map in the mobile device; the prototype of cadastral information service runs in the mobile internet environment, being with the GPS functionality and server-provided map data.

## 5.2. Field Data Collection

The application of field data collection requires the GPS functionality with coordinate transformation and the embedded mapping functionality for monitoring current location on the map. Field data collection is a suitable example of location-aware mobile mapping to provide data source for research and enterprise GIS. Before performing field data collection, sampling method should be decided considering the characteristics of site and data; once a sampling method is chosen, mobile field work can be performed at the site, with the base map and sampling point map generated in accordance with the sampling method.

### 5.2.1. Utilization of Field Data Collection

#### 5.2.1.1. Support for Research and Enterprise GIS

Field data collection is one of the most typical geographic applications using

mobile GIS and GPS (Joyce, 2003). The goal of field data collection can be divided into two categories: one is to provide sample for researches, and the other is to gather source data for a system like enterprise GIS. For the first goal, sampling allows a researcher to draw conclusions about a whole by examining a part. It enables us to estimate characteristics of a phenomenon by observing a portion of the entire phenomenon. (Legge, 2004) For the second goal, gathering source data can construct a site-specific database with accurate location information. It is also necessary for rectifying GIS database constructed by photogrammetry and remote sensing as well as digitizing. (Pendleton, 2003)

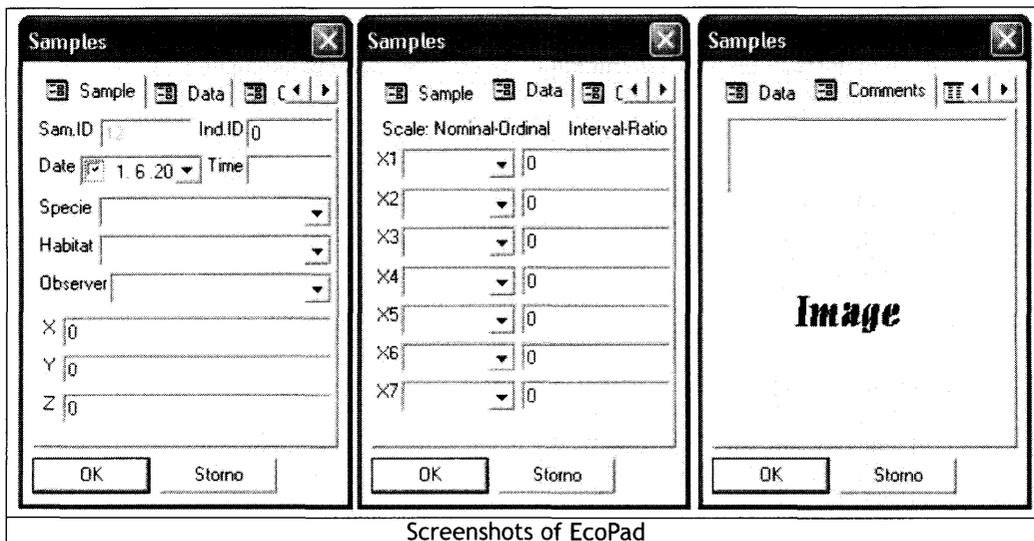
Environmental and socioeconomic researches often require field data collection (Chen & Lee, 2001). Sampling the information about land cover, soil property, plant species, or land use is the example of environmental research; statistical survey like census, economic census, or labor statistics is the example of socioeconomic investigation. (Nusser et al., 2001a)

In addition, facility inspection for pipes, valves and meter gauges of water, gas, and electricity is the example of field data collection in the enterprise GIS. Field workers are expected to update existing data and collect new data without time-consuming and error-prone conversions from paper. They anticipate saving time by receiving work orders and data files digitally to handheld devices without frequent trips to and from the office. (Harrington, 2003a) The timeliness of mobile GIS data relates to the fact that the entire information stream from the field to the enterprise is digital. The digital data stream eliminates the error or bias that is invariably introduced when information is transferred from paper to digital format by a human operator. As

a result of such a “direct-to-digital” approach, the enterprise GIS takes a major step closer to becoming a true geospatial representation of the infrastructure that exists in the field. (Harrington, 2003b)

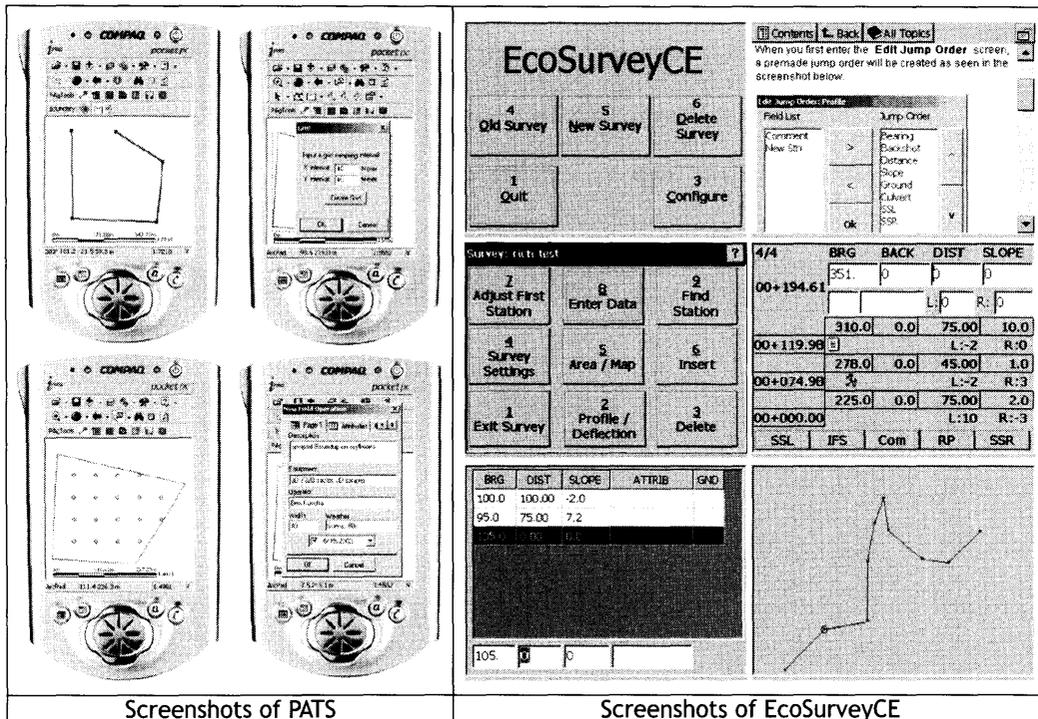
### 5.2.1.2. Existing Applications

Most of the existing applications of field data collection have been composed in VBScript in the ArcPad Application Builder environment. [Figure 59] is the screenshots of EcoPad presented in ESRI User Conference 2003, PATS (Precision Agriculture Tool Suite) presented in InfoAg<sup>103</sup> 2003, and EcoSurveyCE by D.R. Systems<sup>104</sup>.



<sup>103</sup> <http://www.farmresearch.com/infoag>

<sup>104</sup> [http://www.drsystemsinc.com/hh\\_site/data\\_collection\\_apps.htm](http://www.drsystemsinc.com/hh_site/data_collection_apps.htm)



Screenshots of PATS

Screenshots of EcoSurveyCE

[Figure 59] Screenshot of Existing Field Data Collection Applications

The EcoPad allows a mobile user to input the attribute values for ecological survey such as identification codes, date and time of an observation or capture, a name of the specie and its habitat, a name of the observer, and the location {X, Y, Z}. The {X, Y, Z} can be captured via the GPS connected to the Pocket PC, by placing the pen on the map layer or typing appropriate coordinates into the form. (Matejcek, 2003)

The PATS is an application for precision agriculture practices which need basic mapping and field-level record keeping. It has the functionalities of boundary mapping, grid sampling, and field data input with GPS support. (Koostra, 2003)

The EcoSurveyCE is a Windows CE-based data collection program designed for forest engineers. This software provides the functionality of calculation and statistical graph, in addition to the functionality of code inquiry for culvert type and soil type. It integrates with ESRI ArcPad on the Pocket PC, and provides data file to the desktop GIS packages.

### 5.2.2. Sampling Method

In the field data collection either for research or for enterprise GIS, how to collect the data is one of the key issues. Though the whole number of data can be collected according to circumstances (e.g., in case of facility management), sampling is used in most cases (particularly for research data). To acquire credible sample, sampling method is important and should be decided considering the characteristics of site and data.

Conventionally, there are two types of sampling methods: probability sampling and non-probability sampling. The difference between them is that in probability sampling, every unit has a chance of being selected, and the chance can be quantified; whereas in non-probability sampling, every item in a population does not have an equal chance of being selected, and the selection is arbitrary or subjective based on researcher's experience and judgment. The following are the most commonly used probability sampling methods. (Legge, 2004)

- Simple random sampling: In simple random sampling, each member of a population has an equal chance of being included in the sample. Also, each combination of members of the population has an equal chance of composing the sample.

- Systematic sampling: Sometimes called interval sampling, systematic sampling means that there is a gap, or interval, between each selected unit in the sample.
- Stratified sampling: Using stratified sampling, the population is divided into homogeneous, mutually exclusive groups called strata, and then independent samples are selected from each stratum. The sampling method can vary from one stratum to another.
- Cluster sampling: Cluster sampling divides the population into groups or clusters. A number of clusters are selected randomly to represent the total population, and then all units within selected clusters are included in the sample.
- Multi-stage sampling: Multi-stage sampling is like the cluster method, except that it involves picking a sample from within each chosen cluster, rather than including all units in the cluster.
- Multi-phase sampling: A multi-phase sample collects basic information from a large sample of units and then, for a sub-sample of these units, collects more detailed information.

In a conventional sampling design, the selection for a sampling unit does not depend on previous observations made during an initial survey; entire sampling units are selected before any physical sampling in the field ever takes place. Adaptive sampling was proposed as an alternative to conventional sampling methods. It refers to a sampling design in which sampling regions, defined as units, are selected based on values of the variables of interest observed during a sampling survey. (Thompson, 1992; Environmental Protection

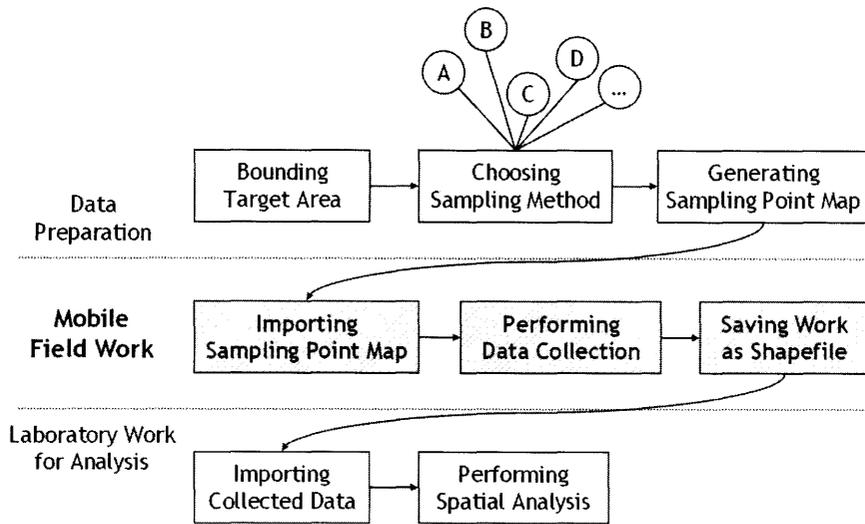
Agency, 2001) The philosophy of adaptive sampling is that spatial sampling will be most efficient if we make decisions about how and where to sample on the basis of all available information. Making the best what we know is the procedure of assessing field variability, identifying key variables, and proceeding to local estimation. (Lark, 2004)

Though the optimal sampling method varies according to the goal of sampling and the characteristics of site, it requires a sampling decision rule to optimize an objective function, which is a utility function for the outcome evaluation criteria. The sampling decision rule is based on the following steps. (Cox, 1999)

- Bounding the area of the greatest interest: by statistical analysis of spatial sample data
- Restricting the class of the decision rules: identifying a set of high-performance decision rules that can be described by a small number of decision parameters
- Simulation/optimization of the decision parameters: estimating the expected performance of each decision rule by simulating its application starting from the current data set

### 5.2.3. Application Prototype

If any sampling method among above-mentioned ones is chosen according to the sampling goal and characteristics of site and data, the base map and sampling point map of the site should be prepared. The generation of a sampling point map requires several statistical techniques, and this work corresponds to the data preparation stage in the workflow of field data collection [Figure 60].



[Figure 60] Workflow of Field Data Collection Application

The workflow is composed of three stages: data preparation, mobile field work, and laboratory work for analysis. Following the data preparation stage, mobile field work is performed at the site, with the base map and sampling point map stored in mobile device like PDA. A mobile user inputs necessary information according to each sampling point, and saves the data as a Shapefile. In the laboratory work stage, the user imports the collected data into desktop GIS package and performs spatial analysis he/she wants. The compatibility with the GIS back-end applications such as ArcView, ArcGIS, and MapInfo is important for the later analysis in laboratory (Pundt, 2002).

Our application prototype was developed for the mobile field work stage, and a scenario of the field data collection is as follows.

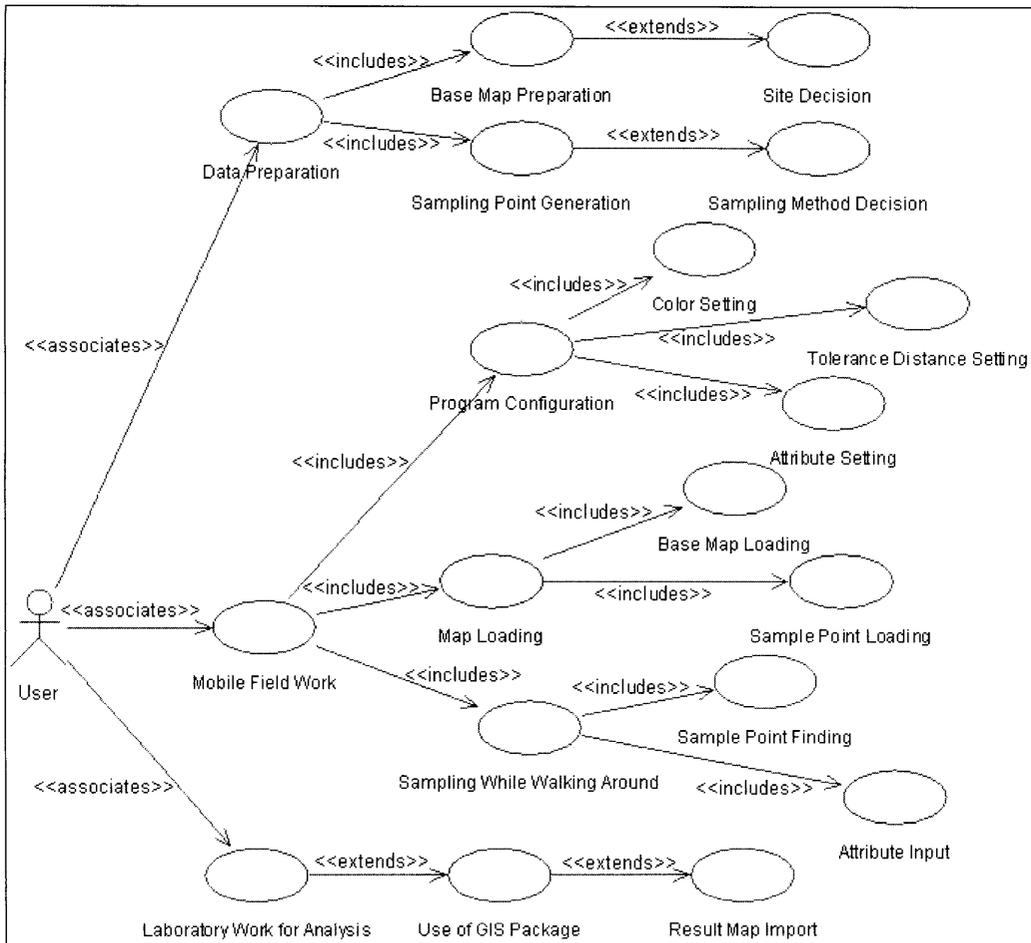
- Open the base map and sampling point map of the site.
- Assign the colors for the current location, the nearest point from the

current location, and already sampled point, respectively.

- Set the tolerance distance between the current location and the nearest target point.
- Input the name of attributes to investigate.
- Walking around the field, application user can see the current location and the nearest target point are painted in each color assigned.
- If the distance between the current location and the nearest target point becomes smaller than the tolerance distance, the attribute input panel shows up automatically.
- Input the attribute values for the sampling point.
- Walking around again, application user can see the already sampled points are painted in the color assigned.

To clarify this scenario, we composed a UML use case diagram [Figure 61].

This diagram shows the conceivable use cases of the field data collection application which was built by using the components of our geocomputing platform.



[Figure 61] Use Case Diagram for Field Data Collection Application

Like the existing applications of field data collection reviewed in the section 5.2.1.2, this application prototype also has the functionality of GPS support and user-defined data input. However, differently from them, the functionality of automatic notification of data input is added to this application prototype. This automatic notification is activated when the distance between the current location and the nearest target point becomes smaller than the

tolerance distance. To Search the nearest point based on distance operation is a core part for the automatic notification of data input. [Code 23] is “DistanceFrom” method of “ClapPoint” class, and [Code 24] is “SearchNearest” method of “ClapPointCollection” class. Both classes in the component of our geocomputing platform were used for “AutoNotify” method in this application [Code 25]. In addition, typical screenshots of this application are described in [Figure 62].

[Code 23] DistanceForm Method for Distance Calculation between Two Points

```
public double DistanceFrom(ClapPoint pnt)
{
    double dX = Math.Pow((X - pnt.X), 2);
    double dY = Math.Pow((Y - pnt.Y), 2);
    return Math.Sqrt(dX + dY);
}
```

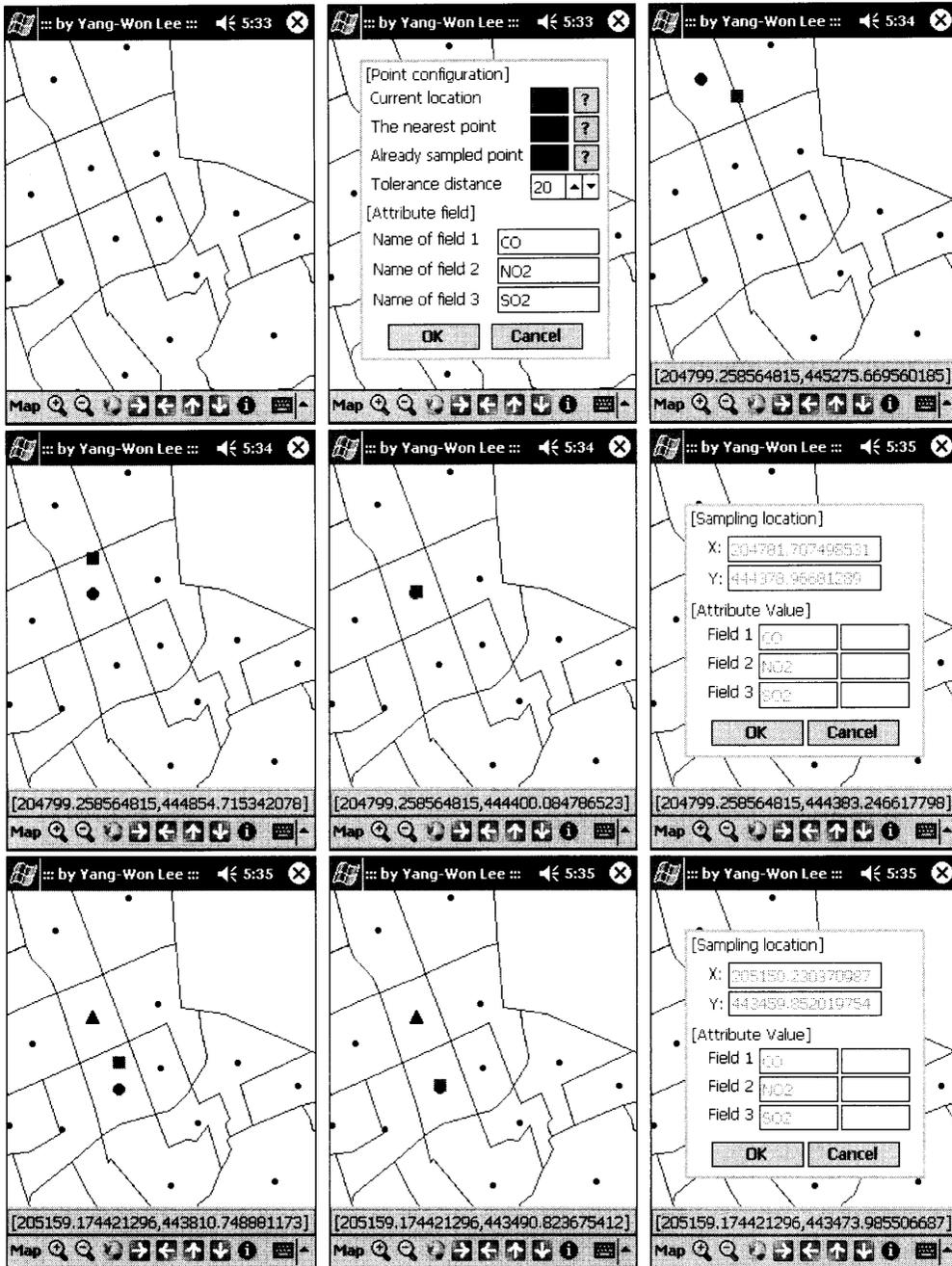
[Code 24] SearchNearest Method for Searching the Nearest Point from a Point

```
public ClapPoint SearchNearest(ClapPoint loc)
{
    double dist, min = 999999999;
    ClapPoint nearest = new ClapPoint();
    foreach(ClapPoint pnt in List)
    {
        dist = loc.DistanceFrom(pnt);
        if(dist < min)
        {
            min = dist;
            nearest = pnt;
        }
    }
    return nearest;
}
```

[Code 25] AutoNotify Method for Automatic Notification of Data Input

```
private void AutoNotify(object sender, System.EventArgs e)
{
    ClapPoint currentLoc = new ClapPoint(_gps.EastTM, _gps.NorthTM);
    ClapPoint nearestPnt = _pntCollection.SearchNearest(currentLoc);
    SetColors(currentLoc, nearestPnt);
}
```

```
double distance = currentLoc.DistanceFrom(nearestPnt);
if(distance <= _config.Tolerance)
{
    SetTimerStop();
    bool OK = InputAttrValues();
    if(OK)
    {
        _sampledPntCollection.Add(nearestPnt);
        SetColors(_sampledPntCollection);
    }
    SetTimerGo();
}
}
```



[Figure 62] Screenshot of Field Data Collection Application

## 5.3. Cadastral Information Service

The application of cadastral information service requires the GPS functionality with coordinate transformation and the internet mapping functionality for monitoring current location on the map. Contrary to the application of field data collection which deals with already chosen area, cadastral information service should be able to cover any area in a city that a user may request. Therefore, a large amount of land parcel map is necessary for this service, and that is why the internet mapping connected to geodata server is required. Though the cadastral information has become available for everyday life, the mobile service for cadastral information has not been developed until now in Korea.

### 5.3.1. LIS and Cadastral Information

Cadastral information is managed by local governments using a geographic information system called LIS (Land Information System) (ESRI, 2001). Also known as Land Records Information System (LRIS), LIS means a system designed to acquire, manage, retrieve, analyze, and display land records. The primary component of LIS is cadastre, which is a public record of the dimensions and value of land parcels<sup>105</sup>, used to record ownership and calculate taxes.

Cadastre, as a legal term, is parcel-based description of interests or rights in real property; typically supported by titles<sup>106</sup> or deeds<sup>107</sup>, and registry.

---

<sup>105</sup> Parcel is a tract or plot of land.

<sup>106</sup> Title is a description of a land parcel.

<sup>107</sup> registration of land transaction with public authority (register of deeds)

Functions of a legal cadastre are as follows<sup>108</sup>:

- Defining property rights (often in conjunction with written and case law)
- describing the extent (spatial, sometimes temporal) of property rights
- supporting land transfer
- providing evidence of ownership (e.g., using land as collateral)
- program administration (e.g., enforcement of laws, targeting of incentives)
- public land management

Cadastre, as a fiscal term, is property valuation and land taxation.

Functions of fiscal cadastre are as follows (Dale & McLaughlin, 1988):

- information base for property taxation
- distribution of funds from public programs
- monitoring and supporting land markets
- information for growth management and land use planning

The cadastral information is closely related to everyday life, and cadastral information services on the internet (wired internet) are now being compiled by metropolitan governments. For example, the Seoul Land Information Service<sup>109</sup> provides cadastral information by parcel [Figure 63]. The information covers the description of urban planning, military facility, agricultural land, forest, nature park, water service, river, cultural assets, electricity development, land transaction, development program, etc., whereas the information about ownership is omitted for the purpose of privacy protection.

---

<sup>108</sup> <http://www.ncgia.ucsb.edu/giscc/units/u164/u164.html>

<sup>109</sup> <http://210.90.46.33:8080/servlet/intro>

시도 서울특별시 구명  구청명  등명  법정동  구분 일반번지  지번 741 12 검색

• 확인내용

1 도시관리 계획	용도지역	도시지역,제2종일반주거지역		
	용도지구			
	용도구역			
	도시계획시설			
	지구단위계획구역			
	기 타	개발밀도관리구역() 기반시설부담구역() 개발행위허가제한지역() 도시개발구역() 재개발구역() 도시계획입안사할()		
2 군사시설	군사시설보호구역() 해군기지구역() 해당없음(+) 민용항공기지구역[비행안전구역() 기지보호구역()]	7 하천	하천구역() 하천예정지() 연안구역() 담건설예정지역() 해당없음(+)	
3 농지	농업진흥구역() 농업보호구역() 해당없음(+)	8 문화재	문화재() 문화재보호구역() 해당없음(+)	
4 산림	보전임지[생산()·공익()] 해당없음(+)	9 전원개발	전원개발사업구역[발전소()·변전소()] 전원개발사업예정구역() 해당없음(+)	
5 자연공원	공원구역() 공원보호구역() 해당없음(+)	10토지거래	허가구역() 해당없음(+)	
6 수도	상수원보호구역() 수질보전특별대책지역() 수변구역() 해당없음(+)	11개발사업	택지개발예정지구() 해당없음(+) 산업단지[국가()·지방()·농공()]	
12 기타				

741-12 매

741-11 매, 741-13 매, 741-14 매, 740-5 매, 740-23 매, 740-6 매, 735-22 매, 735-37 매, 803-11 도, 803-12 도, 740-1 매

발령  
(연속주제)국토계획/도시지  
일반상업지역  
제2종일반주거지역  
제3종일반주거지역

[Figure 63] Web Interface of Seoul Land Information Service

### 5.3.2. Application Prototype

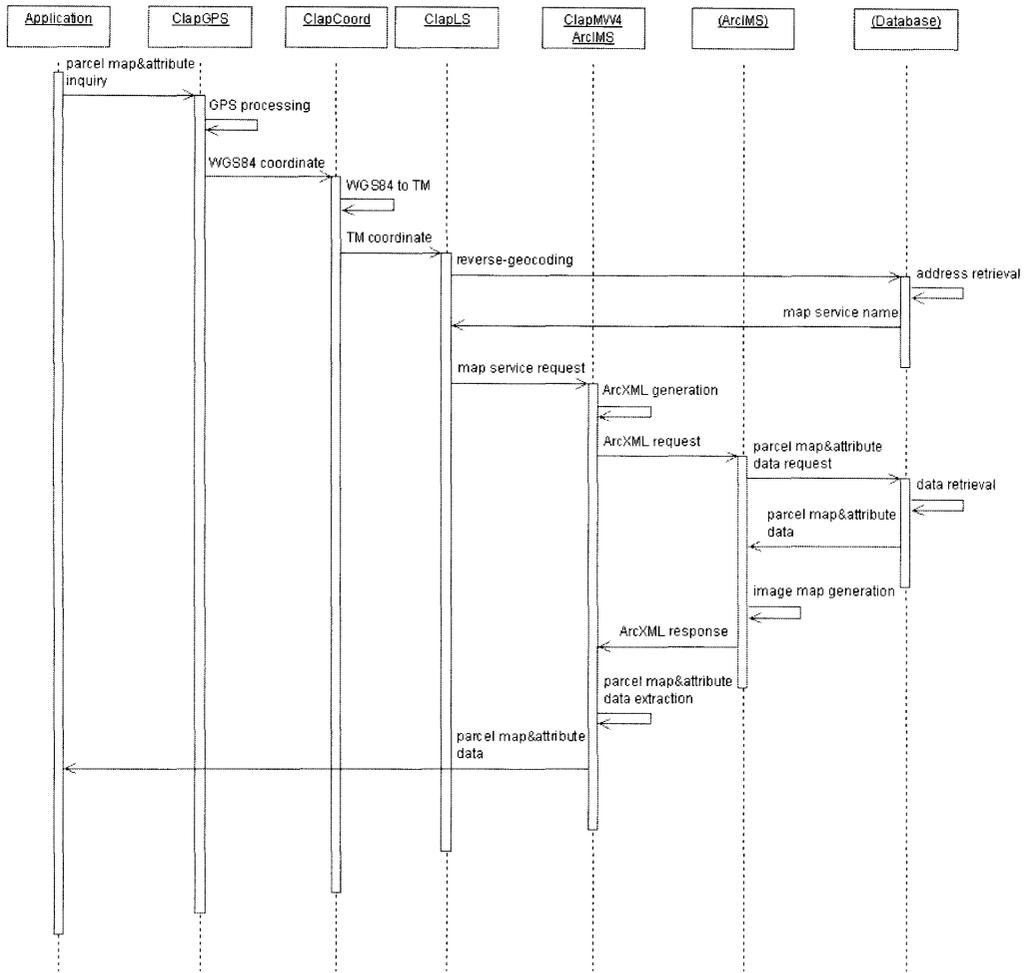
Our application prototype of cadastral information service was composed in the form of Windows program on PDA, which communicates with the location middleware and mapping middleware through the wireless internet. In case of

mobile web browser, the image map transmitted via the mapping middleware can be displayed by URL hyperlink; in case of Windows program on PDA, the image map can be displayed by creating a Bitmap object from URL [Code 26].

[Code 26] GetBitmapFromURL Method for Image Map Display

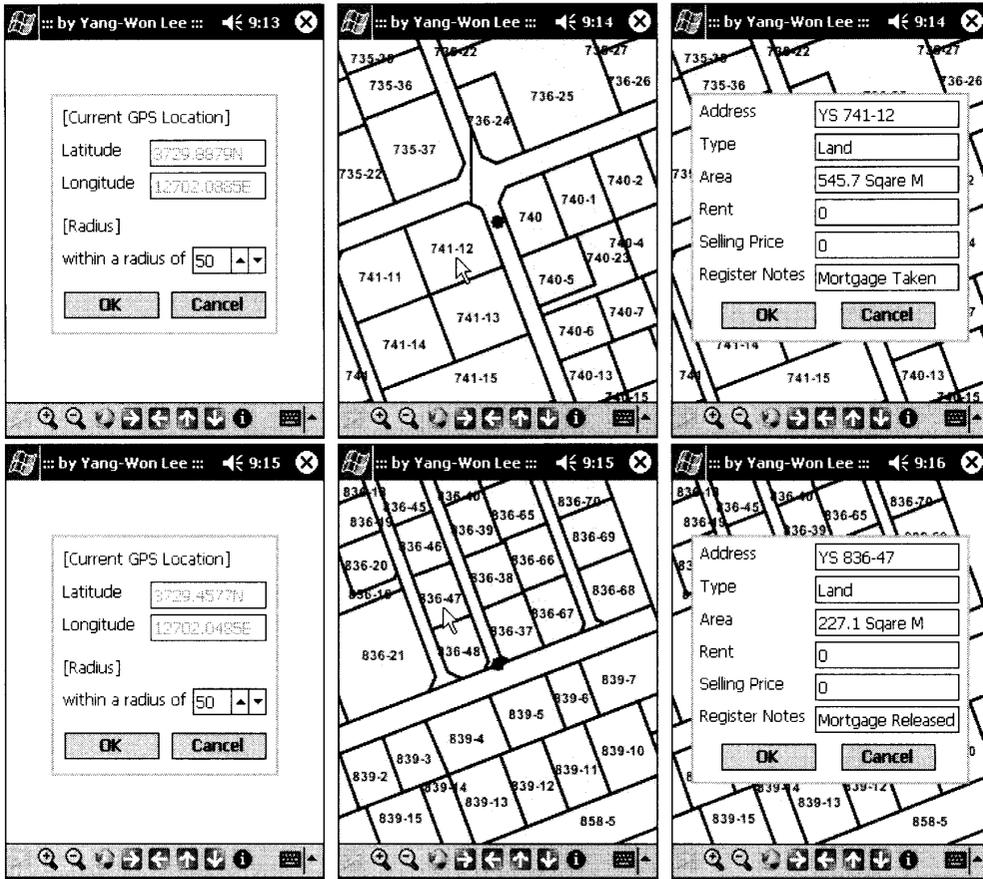
```
private Bitmap GetBitmapFromURL(string imgURL)
{
    System.Net.WebRequest req = System.Net.WebRequest.Create(imgURL);
    System.Net.WebResponse res = req.GetResponse();
    System.IO.Stream stream = res.GetResponseStream();
    bmp = new Bitmap(stream);
    return bmp;
}
```

The cadastral information service was implemented by using the components of “ClapGPS” for GPS data management; “ClapCoord” for coordinate transformation; “ClapLS” for location processing; “ClapMW4ArcIMS” for the brokerage between mobile client and map server. In addition, ArcIMS was used as a map server operating together with a database system which manages a large amount of map data. [Figure 64] is a UML sequence diagram used for programming the application.



[Figure 64] Sequence Diagram for Parcel Information Service

In this application, a mobile user can see the current location is pinpointed on the land parcel map according to the GPS coordinate and radius extent assigned. If the user clicks on a specific parcel, the information of the parcel such as parcel type, area, rent, selling price, and register notes show up [Figure 65].



[Figure 65] Screenshot of Cadastral Information Service

## 6. Conclusion

### 6.1. Research Summary

So far, this thesis has discussed the design and implementation of a geocomputing platform for LBS and modular mobile mapping. This geocomputing platform targets LBS application development, and focuses on modular mobile mapping as a core element of LBS. Modular mobile mapping which support LBS consists of the functionalities of location information processing, mobile embedded mapping, and mobile internet mapping which play a unique role in the LBS application development. Modular mobile mapping becomes not only a core element of LBS, but also a useful application for mobile GIS.

Specifically, the following research objectives were set out at the beginning of this thesis:

- To build a system framework of the geocomputing platform based on the examination of major mobile GIS products and LBS server products in addition to the generic requirements of the application development environment of LBS and modular mobile mapping. Meeting the generic requirements will cover basic functionalities for LBS and modular mobile mapping. Analyzing the key features of current major products will identify technical niches to be filled for the application development environment of LBS and modular mobile mapping.
- To develop components of the geocomputing platform for LBS and modular mobile mapping suitable for the system framework. The

component development will particularly focus on the technical niches that major mobile GIS products and LBS server products do not cover: the need of .NET-based technology and the need of the interoperability with generic map servers.

- To integrate the components into the geocomputing platform and to provide the application development environment of LBS and modular mobile mapping through the integrated geocomputing platform. The application development can be performed by extending necessary components of the geocomputing platform and composing additional codes of the business logic for a specific domain.

To achieve these objectives, we performed literature and technology review, framework design of the geocomputing platform, implementation of the geocomputing platform, and application prototyping. What have been discussed in this thesis can be summarized as follows.

Chapter 2 explored literatures and technologies related to location-aware computing, mobile GIS, wireless map service, LBS, etc. The review of literatures and technologies has become the foundation of the framework design of our geocomputing platform.

The purpose of mobile GIS is to introduce the functionalities of desktop GIS and/or internet GIS to a mobile device, whereas that of LBS is to provide information service based on the location of a mobile user. In case of mobile GIS, mapping is indispensable, but location-awareness is not; in case of LBS, location-awareness is indispensable, but mapping is not. However, the recent trends of mobile GIS and LBS show that mobile GIS is adopting GPS functionality

for the acquisition of accurate location information, and LBS is adopting mapping functionality for the efficient delivery of geographic information.

The investigation of the key features of major mobile GIS products presents the need of .NET-based programmable component for modular mobile mapping. The evaluation criteria of mobile GIS products include two criteria of functionality and programmability. Though the ArcPad provides the richest functionality for composing an application, every application on the client-side requires ArcPad installation because these applications run only under the ArcPad. On the contrary, MapX Mobile has smaller functionality than ArcPad, but the programmability provided by MapX Mobile component allows an independently executable application on Windows. In addition, introducing .NET-based technology for mobile mapping component means not only adopting a new technology, but also applying the new technology to LBS and GIS domain for more efficient and diverse application development environment.

The investigation of the key features of major LBS server products shows the need of the interoperability with generic map servers. The major LBS server products such as Microsoft MapPoint Location Server, Autodesk LocationLogic, and Intergraph IntelliWhere LocationServer are only compatible with their own map servers, not supporting generic map servers. The interoperability with generic map servers is important because the interoperability means the compatibility between components of a system and the extensibility of application development. This interoperability allows application developers' option for map server according to the goal of the application.

In chapter 3, we built the framework of our geocomputing platform for LBS and modular mobile mapping. In addition to the generic requirements of the application development of LBS and modular mobile mapping, the niches major mobile GIS products and LBS server products do not fill have become the foundation for building the framework design of the geocomputing platform.

The two guiding principles of the framework design are “supportability of LBS” and “modularity of mobile mapping.” The functionalities necessary for LBS application development such as location-awareness, mobile embedded mapping, and mobile internet mapping were modularized in the geocomputing platform. This modularization allows convenient access to a specific functional component in the application development process and diverse compatibility with external entities such as map server and database system.

Chapter 4 described the implementation process of our geocomputing platform in the .NET environment. This geocomputing platform is composed of the components of location-awareness, mobile embedded mapping, and mobile internet mapping. The implementation of location-awareness was performed by serial communication, GPS data handling, and coordinate transformation. For the mobile embedded mapping, the functionalities such as map data pre-processing for lightweight map, map visualization including advanced rendering, and topological operation for map visualization and spatial search were implemented. Mobile internet mapping is based on the middleware brokerage for the communication between mobile client and map server. We implemented location middleware for geocoding/reverse geocoding and mapping middleware which interoperates with generic map servers.

The components for location-awareness and mobile embedded mapping were built in the form of DLL (Dynamic Link Library) which is one of the most common deployment formats for Windows application development. The components for mobile internet mapping were built in the form of XML Web Services which provide high performance for internet application development.

In chapter 5, we performed application prototyping to verify the feasibility of our geocomputing platform. For the mobile embedded applications, the component of “ClapGPS.dll,” “ClapCoord.dll,” and “ClapMap.dll” can be imported in the .NET development tool for GPS functionality, for coordinate transformation, and for embedded mapping, respectively. For the mobile internet applications, the component of “ClapGPS.dll” and “ClapCoord.dll” can be used in the same manner. In addition, the XML Web Services of “ClapMW4ArcIMS,” “ClapMW4MapServer,” and “ClapMW4OpenMap” for internet mapping, and “ClapLS” for location information processing can be referenced. These components and XML Web Services provide API methods and properties which cover necessary functionalities for application development, and developers can write additional codes for the business logic of a specific domain.

The application of field data collection can assist mobile field work through the real-time notification of current location, on-site input of attribute value, and the compatibility with desktop GIS packages. The application of cadastral information service is useful for the purchasers/renters of house or land through the real-time notification of current location and on-site inquiry of cadastral information.

## 6.2. Research Implications

The geocomputing platform we developed is based on the principles of “supportability of LBS” and “modularity of mobile mapping.” For this geocomputing platform, we focused on filling the technical niches, which are necessary for mobile GIS and LBS, but not covered by current major products. The components of our geocomputing platform can be practically used for the application development of LBS and modular mobile mapping. During the process of analysis, design, and implementation of the geocomputing platform, the implications can be derived as below.

First, design and implementation of our geocomputing platform was performed on the basis of “supportability of LBS” and “modularity of mobile mapping,” and these principles can play the role of a basic strategy for the development process of a software platform for mobile geocomputing.

The functionalities such as location-awareness, mobile embedded mapping, and mobile internet mapping are necessary for LBS application development, and these functionalities were modularized in our geocomputing platform for the convenient access to a specific functional component and the diverse compatibility with external entities in the process of application development. Functional modularity (or decomposability) is important for arranging complex functionalities for LBS application development in the form of component, and compatible modularity (or composability) is important for the interoperability with the external entities necessary for LBS application development such as map server and database system.

Secondly, design and implementation of our geocomputing platform was performed through the analysis of the major technologies of mobile GIS and LBS, and it provided a clue to the future direction of the development of software platform for mobile geocomputing: “the need of .NET-based programmable component for modular mobile mapping” and “the need of the interoperability with generic map servers.”

The need of .NET-based programmable component arises from the need of complementing the current mobile GIS technology. A close investigation into the characteristics of major mobile GIS products indicates that a programmable component, which allows application developers to build an independently executable program, is not sufficient, and .NET-based technology for modular mobile mapping has not been developed yet. Therefore, it is necessary to consider the component-based programming interface based on .NET environment for the development of mobile geocomputing platform. It provides the convenience and high performance of .NET mobile computing, as a result of introducing and applying a new technology to GIS domain.

The need of the interoperability with generic map servers for LBS platform is rooted in the compatibility between components of a system and the extensibility of application development. Although major LBS server products may not provide an option for choosing map server for commercial reasons, the interoperability with generic map servers should be taken into consideration for the development of mobile geocomputing platform. The interoperability with generic map servers provides flexible choice of mapping engine to achieve the specific goal of LBS application, thereby allowing the extension of various map

data in the region the application requires.

Thirdly, the components of our geocomputing platform can be practically used for the application development of LBS and modular mobile mapping. Those built in the form of DLL and those built in the form of XML Web Services can be referenced in the .NET development environment. These components provide API methods and properties which cover necessary functionalities for application development, and developers can write additional codes for the business logic of a specific domain for their purpose. With the convenience of PNP (Plug and Play), the components save a lot of efforts the developers otherwise have to make to compose the core logic such as GPS functionality with coordinate transformation, embedded mapping, internet mapping, and location information processing.

### 6.3. Future Work

In spite of the achievements and implications of this study, several questions for more advanced mobile geocomputing remain to be answered. In particular, these missions are related to the emerging technologies and standards such as GML, WFS, SVG, and WIPI.

GML suggested by OGC is important as a standard format for transport and storage of geographic information; WFS provides GML-based feature data service over the web. The efforts to utilize GML for mobile computing environment have started to be made these days by individuals, companies, and organizations. SVG is an XML-based specification for describing vector graphic

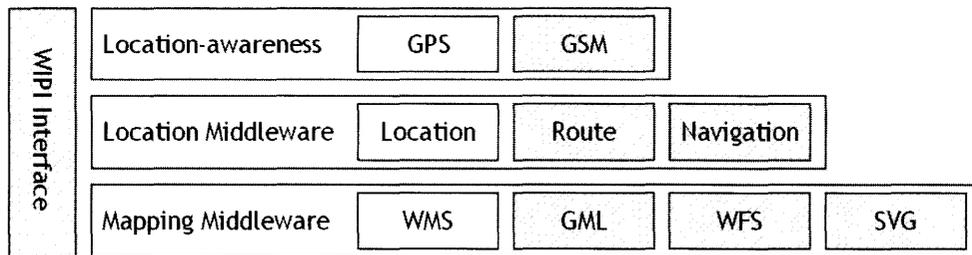
elements, and the advantages of SVG are the fancy visualization and small data size. The World Wide Web Consortium (W3C) has established two mobile profiles for SVG: the first profile, SVG Tiny, is defined for cell phone, and the second profile, SVG Basic, is suitable for PDA.

In Korea, where the location services focus on cell phone, WIPI (Wireless Internet Platform for Interoperability) becomes a new topic for LBS. Until now, three telecommunication service providers, SK Telecom, KTF, and LG Telecom have adopted different VM (Virtual Machine)<sup>110</sup>: SKT adopted GVM/SKVM, KFT adopted BREW/MAP, and LGT adopted Java as their virtual machine of cell phone. The coexistence of four different types of cell phone has brought about a problem in the construction of applications and contents, because each type of virtual machine of cell phone requires different way of programming. To overcome this problem, Korea Wireless Internet Standardization Forum has established WIPI standard (version 1.0 in 2002; version 2.0 in 2004).

Though the technologies and standards such as GML, WFS, SVG, and WIPI are not included in our geocomputing platform, the methods for integrating them into our geocomputing platform need to be studied further to meet the new technological trends. [Figure 66] illustrates the future image of our geocomputing platform.

---

<sup>110</sup> VM is a software that acts as an interface between program code and hardware platform that actually performs the program's instructions (<http://whatis.com>).



[Figure 66] Extended Architecture of the Geocomputing Platform

The Mapping Middleware component may be able to include the managers for GML, WFS, and SVG in addition to the current WMS; the Location Middleware component may be able to include route and navigation service in addition to the current location service; the Location-awareness component may be able to include GSM (Global System for Mobile Communications) method for cellular network in addition to the current GPS method. These components of mapping middleware, location middleware, and location-awareness should be wrapped in WIPI interface for the standardized communication between mobile client and server.

# Bibliography

- Baldwin, C. Y.; & Clark, K. B. (1997) Managing in an Age of Modularity, Harvard Business Review, September-October 1997, pp. 84-93.
- Ball, M. (2003) Field Mapping Capabilities Provide Platform for Change, GEO World, July 2003.
- Banerjee, S.; Agarwal, S.; Kamel, K.; Kochut, A.; Kommareddy, C.; Nadeem, T.; Thakkar, P.; Trinh, B.; Youssef, A.; Youssef, M.; Larsen, R.; Shankar, A. U.; & Agrawala, A. (2001) Rover Technology: Enabling Scalable Location-Aware Computing, available at [http://www.cs.umd.edu/~moustafa/papers/cs\\_tr\\_4312.pdf](http://www.cs.umd.edu/~moustafa/papers/cs_tr_4312.pdf).
- Beinat, E. (2001) Location-based Services: Market and Business Drivers, Geoinformatics, April 2001.
- Booch, G. (1990) Object Oriented Design with Applications, Benjamin-Cummings.
- Brown, A. W.; & Wallnau, K. C. (1996) Engineering of Component-Based Systems, Component-Based Software Engineering: Selected Papers from the Software Engineering Institute, pp. 7-15.
- Carnegie Mellon Software Engineering Institute (1997) Component-Based Software Development / COTS Integration, available at <http://www.sei.cmu.edu/str/descriptions/cbsd.html>.
- Carnegie Mellon Software Engineering Institute (1997) Object-Oriented Analysis, available at <http://www.sei.cmu.edu/str/descriptions/ooanalysis.html>.
- Chappell, D. (2002) Comparing .NET and Java, Proceedings of the Microsoft Tech-Ed 2002 Europe, Track DEV345.
- Chen, Y.; & Lee Y., eds. (2001) Geographical Data Acquisition, Springer.

- Choi, Haeock (2002) LBS: State of the Art in Korea, Proceedings of the 3<sup>rd</sup> Meeting of APT IMT-2000 Forum.
- Choi, Haeock (2003) LBS: Location-Based Services, TTA Journal, no. 86, pp. 59-69.
- Coad, P.; & Yourdon, E. (1991) Object-Oriented Analysis, Prentice-Hall.
- Computer Science and Telecommunications Board (2003) IT Roadmap to a Geospatial Future, The National Academics Press.
- Cornelius, B. (2002) Comparing .NET with Java, available at <http://www.dur.ac.uk/barry.cornelius/papers/comparing.dotnet.with.java>.
- Cox D. D.; Cox, L. H.; & Ensor, K. B. (1999) Spatial Sampling and the Environment, Technical Report of National Institute of Statistical Sciences.
- Cox, L. A. (1999) Adaptive Spatial Sampling of Contaminated Soil, Risk Analysis, vol. 19, no. 6, pp. 1059-1069.
- D'Roza, T.; & Bilchev, G. (2003) An Overview of Location-Based Services, BT Technology Journal, vol. 21, no. 1, pp. 20-27.
- Dale, R. F.; & McLaughlin, J. D. (1988) Land Information Management: An Introduction with Special Reference to Cadastral Problems in Third World Countries, Oxford.
- De Vita, E.; Piras, A.; & Sanna, S. (2003) Using Compact GML to Deploy Interactive Maps on Mobile Devices, Proceedings of the 20<sup>th</sup> International World Wide Web Conference.
- Dey, A. K.; & Abowd, G. D. (1999) Towards a Better Understanding of Context and Context-Awareness, GVU Technical Report GITGVU-99-22, College of Computing, Georgia Institute of Technology.

- Drane, C.; Macnaughtan, M.; & Scott, C. (1998) Positioning GSM Telephones, IEEE Communications Magazine, vol. 36, no. 4, pp. 46-59.
- Drew, W. (2003) Wireless Networks: New Meaning to Ubiquitous Computing, Journal of Academic Librarianship, vol. 29, no. 2, pp. 102-106.
- Edwards, W. (2002) J2EE and .NET Application Server and Web Services Performance Comparison, available at <http://www.middlewareresearch.com/endeavors/021028J2EEDOTNET/endeavor.jsp>.
- Electronics and Telecommunications Research Institute (2003) High-Speed Wireless Internet Technology of 2.3 GHz Frequency Range, Proceedings of the 2<sup>nd</sup> TTA Seminar on the Core Standard Technology.
- Electronics and Telecommunications Research Institute (2004) Spatial Data Server for Mobile Environment, available at <http://www.edbt04.gr/demos/lee.ppt>.
- Emmerich, W. (2002) Distributed Component Technologies and Their Software Engineering Implications, Proceedings of the 24<sup>th</sup> International Conference on Software engineering, pp. 537-546.
- Environmental Protection Agency (2001) Adaptive Sampling and Analysis Programs, available at [http://www.epa.gov/tio/download/char/asaps\\_itsr\\_doe-em-0592.pdf](http://www.epa.gov/tio/download/char/asaps_itsr_doe-em-0592.pdf).
- ESRI (1998) ESRI Shapefile Technical Description, ESRI.
- ESRI (2001) The ESRI Press Dictionary of GIS Terminology, ESRI.
- ESRI (2002) ArcPad: Mobile Mapping and GIS, White Paper of ESRI.
- Evans, A. S.; & Welling, A. J. (1999) UML and the Formal Development of Safety-Critical Real-Time Systems, Proceedings of the IEE Colloquium on Applicable Modeling, Verification and Analysis Techniques for Real-Time

Systems, pp. 2/1-2/4.

Filev, A.; Loton, T.; McNeish, K.; Schoellmann, B.; Slater, J.; & Wu, C. G. (2003) UML with Visual Studio .NET, Wrox.

Goode, C.; Kauffman, J.; Miller, C. L.; Raybould, N.; Sivakumar, S.; Sussman, D.; Cornes, O.; Birdwell, R.; Butler, M.; Johnson, G.; Krishnamoorthy, A.; Llibre, J. T.; & Ullman, C. (2002) ASP.NET with C#, Wrox

Goulão, M.; & Abreu, F. B. (2002) The Quest for Software Components Quality, Proceedings of the 26<sup>th</sup> Annual International Conference on Computer Software and Applications (COMPSAC 2002), pp. 313-318.

Han, Ki-Joon (2003) A Review of LBS Standards and Trends, Informationalization Policy, vol. 10, no. 4, pp. 3-17.

Hanaro Telecom (2003) Technology Trends in 2.3 GHz High-Speed Wireless Internet, Proceedings of the 2<sup>nd</sup> TTA Seminar on the Core Standard Technology.

Harrington, A. (2003) Field Experiences Highlight Mobile GIS Benefits, GEO World, June 2003.

Harrington, A. (2003) Increase Productivity with Timely Data, GEO World, Feb 2003.

Hecht, L. (2000) Location Services: Remarks, Considerations, Challenges, Proceedings of the 1<sup>st</sup> Location Interoperability Forum Meeting of Open GIS Consortium.

Heineman, G.T. (1998) A Model for Designing Adaptable Software Components, Proceedings of the 22<sup>nd</sup> Annual International Conference on Computer Software and Applications (COMPSAC'98), pp. 121-127.

Hind, J. (2002) Use P/Invoke to Develop a .NET Base Class Library for Serial Device Communications, MSDN Magazine, vol. 17, no. 10.

- Hjelm, J. (2003) *Creating Location Services for the Wireless Web*, John Wiley & Sons.
- Hofmann, H.; Volker, M.; & Jeanne, S. (1999) *Mechanisms of Component-Oriented Software Development*, *Internet Research: Electronic Networking Applications and Policy*, vol. 9, no. 1, pp. 66-75.
- Hughes, B.; & Cahill, V. (2003) *Towards Real-time Event-Based Communication in Mobile Ad Hoc Wireless Networks*, Technical Report of Department of Computer Science, Trinity College, Dublin.
- Hwang, Chul-Sue (1998) *A Design and Implementation of Distributed Object-Oriented Digital Maps*, PhD Dissertation of Seoul National University.
- Jacobson, I. (1992) *Object-Oriented Software Engineering: A Use-Case Driven Approach*, Addison-Wesley.
- Jin, Heui-Chae (2003) *Technological Review for Location Based Services*, *Journal of the Korea Open GIS Association*, vol. 5, no. 1, pp. 13-25.
- Jin, Heui-Chae; & Hong, Kilpyo (2003) *Model Analysis and Technology Research for Location Based Services*, *Proceedings of the 2003 GIS/RS Joint Spring Meeting*, pp. 438-450.
- Joyce, R. (2003) *Using GPS for Soil Sampling*, *R & D Highlands Ranch*, vol. 45, no. 10, p. 29.
- Kennedy, M. (2002) *The Global Positioning System and GIS*, Taylor & Francis.
- Kim, Eun-Hyoung; & Park, Jun-Gu (2004) *Study on Service Type of Wireless GIS in Public Sector*, *Proceedings of the 2004 GIS/RS Joint Spring Meeting*, pp. 147-166.
- Kim, Hyung-Jun (2003) *A Study on the Development of Base Technique and Main Functions for the Mobile GIS*, Master Thesis of Chonbuk National University.

- Kim, Jung-Ho (1999) A Study on Transformation of Coordinates by GPS, PhD Dissertation of Kyonggi University.
- Kim, Seogyun; & Lee, Joonseok (2003) Analysis on the Industry Trends Related to LBS Market, Trend Analysis on Informationalization Technology, vol. 9, no. 4, pp. 20-26.
- Koepfel, I. (2000) What Are Location Services? - From a GIS Perspective, Technical Report of Sun Microsystems.
- Korea Geospatial Information & Communication Company (2003) Introduction to Location Based Services, Korea Geospatial Information & Communication Company.
- Korea Information Strategy Development Institute (2001) A Study on the Law System for the Promotion of the Use of Spatial Information, Korea Information Strategy Development Institute.
- Korea Network Information Center (2003) 2003 Survey on the Wireless Internet Use, Korea Network Information Center.
- Korea Wireless Internet Standardization Forum (2004) Map Service for Portable Device: Phase 1 - Scopes and Requirements, Korea Wireless Internet Standardization Forum.
- Korea Wireless Internet Standardization Forum (2004) Wireless Internet Platform for Interoperability 2.0, Korea Wireless Internet Standardization Forum.
- Krasner, G. E.; & Pope, S. T. (1988) A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80, Journal of Object-Oriented Programming, vol. 1, no. 3, pp. 26-49.
- Ku, Cha-Yong (2002) The Scale Characteristics of Satellite Imagery with Spatial Resolution, PhD Dissertation of Seoul National University.

- Koostra, B. (2003) Development of a Precision Agriculture Tool Suite Using ArcPad, available at <http://www.bae.uky.edu/bkoostra/papers/InfoAg2003.pdf>.
- Korea LBS Standardization Forum (2003) Standard for Functional Interface for Location Based Service - Stage 1, Korea LBS Standardization Forum.
- Korea LBS Standardization Forum (2004) LBS Service Platform Stage 2 KLP (Korea Location Protocol) Version 1.0.0, Korea LBS Standardization Forum.
- Langley, W. (2003) ArcPad and Field Data Collection at the City of Garland, available at <http://gis.esri.com/library/userconf/proc03/p0198.pdf>.
- Lark, R. M. (2004) Strategies for Cost-Effective Soil Sampling, Proceedings of the HGCA Conference 2004, pp. 5.1-5.14.
- Lee, Jong-Hun (2002) LBS Related Fields, Electronics and Telecommunications Research Institute.
- Lee, S.; Lin, J.; & Jiau, H. C. (2000) Constructing COTS-Based Software Components with Reusing COTS Products, Proceedings of the 2<sup>nd</sup> International Symposium on Multimedia Software Engineering, pp. 289-296.
- Lee, Sang-Yun; Kim, Seonja; & Kim, Heungnam (2004) The Current and Future Trend of the Standardization of WIPI in Korea, Communications of the Korea Information Science Society, vol. 22, no. 1, pp. 16-23.
- Legge, J. A. (2004) Statistics: Power from Data, Statistics Canada.
- Lewis, J. L. (2002) A Reliable Test for Inclusion of a Point in a Polygon, ACM SIGCSE Bulletin, vol. 34, no. 4, pp. 81-84.
- Lyytinen, K.; & Yoo, Y. (2002) Issues and Challenges in Ubiquitous Computing, Communications of the ACM, vol. 45, no. 12, pp. 62-65.

MapInfo Corporation (2000) Mobile Location Services, White Paper of MapInfo Corporation.

MapInfo Corporation (2002) MapX Mobile Developer's Guide, MapInfo Corporation.

Matejicek, L. (2003) Development of Software Tools for Ecological Field Studies Using ArcPad, Proceedings of the 23<sup>rd</sup> Annual ESRI International User Conference.

McDonald, K. (2002) The Modernization of GPS: Plans, New Capabilities and the Future Relationship to Galileo, Journal of Global Positioning Systems, vol. 1, no. 1, pp. 1-17.

Mei, H.; Zhang, L.; & Yang, F. (2001) A Software Configuration Management Model for Supporting Component-Based Software Development, ACM SIGSOFT Software Engineering Notes Archive, vol. 26, no. 2, pp. 53-58.

Meier, R. (2002) Communication Paradigms for Mobile Computing, Mobile Computing and Communications Review, vol. 6, no. 4, pp. 56-58.

Microsoft Corporation (2001) Global XML Web Services Architecture, White Paper of Microsoft Corporation.

Mikkola, J. H. (2003) Modularity, Component Outsourcing, and Inter-firm Learning, R&D Management, vol. 33, no. 4, pp. 439-454.

Milroy, S (2003) .NET Compact Framework Overview, available at <http://www.winnetmag.com/Article/ArticleID/38314/38314.html>

Ministry of Information and Communication (2003) LBS Industry Support Plan, Ministry of Information and Communication, Korea.

Moran, T. P.; & Dourish, P. (2001) Introduction to This Special Issue on Context-Aware Computing, Human-Computer Interaction, vol. 16, no. 2, 3 & 4, pp. 87-95.

Morisio, M.; Seaman, C. B.; Basili, V. R.; Parra, A. T.; Kraft, S. E.; & Condon, S. E. (2002) COTS-Based Software Development: Processes and Open Issues, *Journal of Systems and Software*, vol. 61, no. 3, pp. 189-199.

National Computerization Agency (2002) A Study on Standardization and Technologies for GIS Services Based on Location Information, National Computerization Agency.

National Geographic Information Institute (2002) A Study on the Construction of National Statistics Digital Map, National Geographic Information Institute.

National Geographic Information Institute (2003) A Study on the Coordinate Transformation of Digital Map II, National Geographic Information Institute.

National Marine Electronics Association (2002) NMEA-0183 Standard Input/Output Messages Protocol, National Marine Electronics Association.

Niedzwiadek, H. (2002) OGC OpenLS Platform, available at [http://www.jlocationservices.com/Newsletter/Dec.02/AboutOpenLS\\_Platform\\_Dec2002.pdf](http://www.jlocationservices.com/Newsletter/Dec.02/AboutOpenLS_Platform_Dec2002.pdf)

Nim, Young-Bin (1997) A Study on the Coordinate Transformation Properties for the Practice Use of GPS, PhD Dissertation of Dong-A University.

No, Kyung-Chan (2001) Design and Implementation of Geographical Information System on Cellular Phone, Master Thesis of Hanyang University.

Nusser, S.; Miller, L.; Goodchild, M.; & Clarke, K. (2001) Collecting and Using Geospatial Data in the Field, Proceedings of the 2001 NSF Workshop on Challenges in Statistical Digital Government.

Nusser, S.; Miller, L.; Clarke, K.; & Goodchild, M. (2001) Future Views of Field Data Collection in Statistical Surveys, Proceedings of dg.o 2001 National Conference on Digital Government Research.

- Object Management Group (2003) *OMG Unified Modeling Language Specification*, Object Management Group.
- Object Management Group (2004) *Introduction to OMG's UML*, available at [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm).
- Open GIS Consortium (1999) *OpenGIS Simple Features Specification for OLE/COM Revision 1.1*, Open GIS Consortium.
- Open GIS Consortium (2002) *Web Feature Service Implementation Specification*, Open GIS Consortium.
- Open GIS Consortium (2004) *OpenGIS Geography Markup Language (GML) Implementation Specification*, Open GIS Consortium.
- Open GIS Consortium (2004) *OpenGIS Location Services (OpenLS): Core Services*, Open GIS Consortium.
- Open GIS Consortium (2004) *OpenGIS Web Map Service Interface*, Open GIS Consortium.
- Park, Key-Ho; Jung, Jae-Gon; & Hwang, Myoung-Hwa (2003) *A Study on Mobile SFA System Prototyping Using P2P LBS Service*, *The Journal of Geographic Information System Association of Korea*, vol. 11, no. 1, pp. 61-72.
- Parkinson, B. W. (1996) *Global Positioning system: Theory and Applications*, vol. I / II, Lawrenceville.
- Patterson, C. A.; Muntz, R. R.; & Pancake, C.M. (2003) *Challenges in Location-Aware Computing*, *Pervasive Computing*, vol. 2, no. 2, pp. 80-89.
- Pendleton, G. (2003) *GIS Data Collection vs. Mobile GIS*, *GEO World*, May 2003.
- Pocket Systems (2004) *PocketGIS User Manual*, Pocket Systems.
- Pundt, H. (2002) *Field Data Collection with Mobile GIS: Dependencies between Semantics and Data Quality*, *Geoinformatica*, vol. 6, no. 4, pp. 363-380.

- Pyeon, Mu-Wook (1991) A Study on Development of the Automatic Vehicle Location System Using Global Positioning System, Master Thesis of Seoul National University.
- Roof, L.; & Fergus, D. (2003) The Definitive Guide to the .NET Compact Framework, APress.
- Seoul Development Institute (2002) A Study on the Mobile GIS Application, Seoul Development Institute.
- Seoul Development Institute (2003) A Study on Using Mobile Technologies for Seoul Metropolitan Government, Seoul Development Institute.
- Si, Jong-ik (2003) Setting to Work for Mobile GIS Standard, Digital Times, 8/21/03.
- Stanski, P.; & Wiharto, M. (2003) Applications on Handhelds, available at <http://kajol.csse.monash.edu.au/cse3211/lectures/CSE3211-cfdotnet95.ppt>.
- Stockus, A.; Bouju, A.; Bertrand, F.; & Boursier, P. (1999) Integrating GPS Data within Embedded Internet GIS, Proceedings of the 7<sup>th</sup> ACM International Symposium on Geographic Information Systems.
- Taylor, D. A. (1990) Object-Oriented Technology: A Manager's Guide, Addison-Wesley.
- The Middleware Company (2003) J2EE and .NET, available at <http://gotdotnet.com/team/compare/Middleware30.pdf>.
- Thompson, S. K. (1992) Sampling, John Wiley & Sons.
- Thrunet Corporation (2003) Application Plan for 2.3 GHz PWLAN, Proceedings of the 2<sup>nd</sup> TTA Seminar on the Core Standard Technology.
- Tu, S.; Xu, L.; Abdelguerfi, M.; & Ratcliff, J. J. (2002) Achieving Interoperability

for Integration of Heterogeneous COTS Geographic Information Systems, Proceedings of the 10<sup>th</sup> ACM International Symposium on Advances in Geographic Information Systems.

Um, Jung-Sup; & Jung, Ji-Hoon (2003) Validity Evaluation of Real Time Mobile GIS Combined with PDA in University Building Facility Management, The Journal of Geographic Information System Association of Korea, vol. 11, no. 1, pp. 41-60.

Virrantaus, K.; Tirri, H.; Veijalainen, J.; Markkula, J.; Katanosov, A.; Garmash, A.; & Terziyan, V. (2001) Developing GIS-Supported Location-Based Services, Proceedings of the 2<sup>nd</sup> International Conference on Web Information Systems Engineering, pp. 66-75.

Wang, J. A. (2000) Towards Component-Based Software Engineering, The Journal of Computing in Small Colleges, vol. 16, no. 1, 182-194.

Wheeler, S. (1996) Object-Oriented Programming with X-Designer, available at <http://rd13doc.cern.ch/Atlas/Notes/004/Note004-1.html>.

Wigley, A. (2003) Microsoft .Net Compact Framework, Microsoft Press.

Wolter, R. (2001) XML Web Services Basics, available at <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebsrv/html/websrvbasics.asp>.

World Wide Web Consortium, Mobile SVG Profiles: SVG Tiny and SVG Basic, available at <http://www.w3.org/TR/2002/PR-SVGMobile-20021115>.

You, Jinsoo; Oh, Byoungwoo; & Ha, Suwook (2004) Towards Developing a Standardized GML Profile for Mobile Application Services, Proceedings of the GML Days Conference 2004.

Yuan, M. J. (2002) A comparison of the Philosophies, Approaches, and Features of J2ME and the Upcoming .Net CF, available at

[http://www.javaworld.com/javaworld/jw-02-2003/jw-0221-wireless\\_p.html](http://www.javaworld.com/javaworld/jw-02-2003/jw-0221-wireless_p.html).

Yuan, M. J. (2003) J2ME and .Net Compact Framework in Action, available at [http://www.javaworld.com/javaworld/jw-05-2003/jw-0516-wireless\\_p.html](http://www.javaworld.com/javaworld/jw-05-2003/jw-0516-wireless_p.html).

Yun, Jae-Kwan (2003) A Real-Time Mobile GIS Based on the HBR-Tree for Location Based Services, PhD Dissertation of Konkuk University.

Zeiler, M. (1992) Integrating GIS and GPS for Mapping and Analysis of Electric Distribution Circuits, Proceedings of the 36<sup>th</sup> Annual Conference on Rural Electric Power, pp. D2/1-D2/10.

Zingler, M.; Fischer, P.; & Lichtenegger, J. (1999) Wireless Field Data Collection and EO-GIS-GPS Integration, Computers, Environment and Urban Systems, vol. 23, no. 4, pp. 305-313.

# 국 문 초 록

## LBS와 기능성 모바일매핑을 위한 지오컴퓨팅 플랫폼의 설계와 구현

서울대학교 대학원 지리학과

이 양 원

정보기술의 진보가 가져온 컴퓨팅환경과 일상생활의 변화 중 하나는 모빌리티의 증가라고 할 수 있다. 최근 들어, 무선인터넷의 보급이 확대되고 있으며, 휴대폰이나 PDA (Personal Digital Assistant)와 같은 모바일기기용 애플리케이션 개발과 콘텐츠 구축이 활발해지고 있다. 이 연구는 모바일기기 사용자의 실시간 위치를 통해 다양한 정보를 제공하는 위치기반서비스 (Location Based Services: LBS)와 그 핵심요소로서의 모바일매핑 (Mobile Mapping)을 지원하는 지오컴퓨팅 플랫폼 (Geocomputing Platform)의 새로운 모형을 제시하는 데 그 목적을 둔다.

이 연구에서는 모바일 GIS와 LBS에 관한 기술문헌의 검토와 함께 기존의 메이저급 모바일 GIS 제품들과 LBS 서버 제품들의 기술특성을 분석함으로써 모바일 환경에서의 지오컴퓨팅 플랫폼을 위한 시스템 구조를 도출하고, 이를 바탕으로 ‘반드시 필요하지만 기존기술이 커버하지 않고 있는’ 틈새 (Niche)를 채울 수 있는 지오컴퓨팅 플랫폼을 개발한다. 기존기술에 대한 보완으로서 이 연구에서 중점을 둔 것은 닷넷 기반의 모바일매핑 컴포넌트 (.NET-Based Mobile Mapping Component)의 개발과 일반 지도서버들과의 상호운용 (Interoperability with Generic Map Servers)의 구현이다.

컴포넌트 기술은 애플리케이션 개발에 필요한 모듈화된 (Modularized) 맵핑 기능들을 취사선택 및 조합하여 적재적소에 사용 가능하게 하고, 닷넷은 새로이 부상하는 진보된 기술로서 모바일 GIS 영역에서도 적극적인 활용이 기대되고 있다. 이 연구에서는 이러한 요구에 부응하여 LBS와 모바일매핑을 위한 닷넷 기반의 컴포넌트를 개발하고, 모바일 임베디드 (Mobile Embedded) 환경과 모바일 인터넷 (Mobile Internet) 환경 모두에 적용 가능한 통합된 형태의 지오컴퓨팅 플랫폼을 구성하였다.

이 연구에서 개발한 지오컴퓨팅 플랫폼의 특징으로서 일반 지도서버들과의 상호운용이 중요한 이유는, 애플리케이션 구축에 있어서 지도서버 선택의 제약 없이 필요한 기능을 가진 지도서버를 필요한 곳에 배치함으로써 애플리케이션 개발자에게 지도서버 선택권을 제공하는 데 있다. 이는 애플리케이션의 목적에 따라 행정구역도, 도로지도, 지형도, 지번도 뿐만 아니라 3차원지도나 멀티미디어지도와 같이 다양한 지도를 플러그인 (Plugin) 가능하게 함으로써 지리정보를 적극적으로 활용하는 LBS 애플리케이션 구축을 지원하는 것이다.

이 연구에서 개발한 지오컴퓨팅 플랫폼은 GML (Geography Markup Language), WFS (Web Feature Service), SVG (Scalar Vector Graphics) 등 최근 GIS 분야에서 각광받고 있는 기술들과, 우리나라 무선인터넷의 새로운 표준인 WIPI (Wireless Internet Platform for Interoperability)의 지원이 가능하도록 확장하는 것을 향후의 과제로 삼고 있다.

주요어: 위치기반서비스, 모바일 지리정보시스템, 맵핑 미들웨어, 닷넷 기술  
학 번: 2001-30097

地理學論叢 別호 59

2004년 8월 31일

발행처 : 서울대학교 국토문제연구소

발행인 : 유우익

편집인 : 박수진

인쇄소 : 신일문화사