博士論文

Development of learning assistance tool on geospatial information technology

（地理空間情報技術学習支援ツールの開発）

太田守重

謝辞 － Acknowledgements

この論文は、1995 年以来、国際標準化機構 (ISO)の専門委員会 211 (ISO/TC 211) における地理情報標準検討に参加させていただいた経験などを踏まえて行った、地理空間情報技術 (GIT) の学習支援ツールの研究開発をまとめたものです。この間、多くの皆様からご指導ご鞭撻を賜りました。

この論文の執筆にあたり、まず、浅見泰司教授（東京大学）に心より感謝の意を捧げます。先生が地理情報システム学会 (GISA) の会長をお務めの節、私は学会が運営している GIS 資格認定協会(GISCA)の幹事長を務めておりましたが、GISCA の代表を快くお引きうけいただきました。また、先生が GIS 教育に関する科研費研究の代表者をお務めの節には、私も研究協力者として参加させていただきましたが、これらのことがご縁と思い、本研究についてもご指導いただくことを、たってお願い申し上げました。これまで長きにわたり忍耐強くご指導くださいましたこと、心より感謝申し上げます。

今井浩教授（東京大学）におかれましては、ISO/TC 211 の設立以来、標準化プロジェクトチームのリーダをお務めの後、ワーキンググループの代表 (convener) にご就任になり、さらに TC 211 国内委員会の委員として、貴重なご示唆を頂いて参りました。本研究におきましてもご指導、ご鞭撻を賜りましたこと、心より感謝申し上げます。

久保田光一教授（中央大学）におかれましては、様々な地理情報標準化活動の場で貴重なご意見をいただくのみならず、科研費研究の GIT 班の班長として長きにわたりご指導いただくとともに、中央大学において、GIT の教育実践の機会をお与えくださいました。心から感謝申し上げます。

高見淳史准教授（東京大学）及び廣井悠准教授（東京大学）におかれましては、本研究における教育の有効性検証などについて貴重なご指摘を賜りましたことにつき、心より感謝申し上げます。

ISO/TC 211 における標準化活動の体験は、GIT の体系を整理する上で、大変役に立ちました。標準化の過程で知り合いになった方々との意義深い交流がなければ、この研究は始まりませんでした。TC の設立当初から長く議長を務められた Olaf Østensen 氏をはじめ、Morten Borrebæk 氏、Steinar Høseggen 氏（ノルウェー）、 Charles Roswell 氏、John Herring 氏、David Danko 氏、Robin Fegeas 氏（アメリカ）、Rob Walker 氏（イギリス）、Clemens Portele 氏（ドイツ）、Ron Lake 氏、Herman Varma 氏（カナダ）、Ken

Bullock 氏、Andrew Jones 氏（オーストラリア）、Antony Cooper 氏、Serena Coetzee 氏（南アフリカ）など、数多くの方々を思い出します。Serena Coetzee 準教授 (University of Pretoria)については、2015 年に開催された国際地図学会リオデジャネイロ大会において、先生などが主催した事前会議で口頭発表したことがきっかけとなり、International Journal of Cartography への論文投稿を試み、その掲載が実現しました。編集委員長をお務めの元国際地図学会会長 William Cartwright 教授（RMIT University）をはじめ、関係者の皆様にも心から感謝致します。また、EU の GIS 教育関係者が開催している EU GIS Education Seminar (EUGISES)では何回か発表させていただいてきましたが、アイルランドの University College of Cork で 2014 年に開催された折には、当方の口頭発表に対して、GIS 教育の権威で、以前 UC Santa Barbara におられ、GIS 教育の科研費研究の行事で来日されたこともある Karen Kemp 先生から、貴重なコメントをいただいたことも思い出されます。

国内における標準化活動においては、発足以来長く TC 211 国内委員会の委員長として貢献された故伊理正夫先生には、長きにわたり様々な場でご指導を賜って参りました。先生のお人柄を偲ぶと共に、心からの謝意を捧げます。そして現在国内委員長をお務めで、私が最初に参加させていただいた、GIS 教育に関する科研費プロジェクトの代表者でもあった東京大学名誉教授岡部篤之教授（青山学院大学）、GISA において GISCA の設立を主導され、TC 211 では住所関係の標準化にご尽力された碓井照子名誉教授（奈良大学）、TC 211 や Open Geospatial Consortium (OGC) において標準化プロジェクトのリーダとして貢献され、長きにわたりご助言を賜っている柴崎亮介教授（東京大学）、IT の視点でご助言を賜っている瀬崎薫教授（東京大学）、委員各位、ならびに幹事会幹事各位、そして、元国土地理院長稲葉和雄氏（現一般財団法人日本デジタル道路地図協会専務理事）をはじめとして、標準化活動を実質的に牽引しておられる国土地理院の皆様、国内審議団体である公益財団法人日本測量調査技術協会 (APA) の皆様、さらには、APA と共に地理情報分野の標準化活動を推進しておられる一般財団法人日本情報経済社会推進協会（JIPDEC）の坂下哲也常務理事、郡司哲也氏、Reese Plews 氏をはじめとする皆様に、心から感謝致します。はからずも 2007 年には、経済産業省産業技術環境局長より国際標準化貢献者表彰を授けていただき、2013 年には国土交通省国土地理院長より測量事業関係功労者表彰を授けていただきましたが、これらの皆様のお陰があったからこそであり、改めて感謝申し上げます。また、Plews 氏には、学術誌 Cartography and Geographic Information Science に掲載され

た論文の共著者としてもご協力を賜りました。

ソフトウェアの開発については、GIS 教育に関する科研費研究の中に設けられている GIT 班の皆様のご支援が大きな力になりました。TC 211 国内委員会の委員でもあり、GISA の空間 IT-SIG の主査をお務めになった有川正俊教授（前東京大学、現秋田大学）、藤田秀之准教授（電気通信大学）、古橋大地教授（青山学院大学）、黒川史子氏（国際航業株式会社）に心から感謝致します。なお、本研究で開発したソフトウェア gittok は、2014 年度に国土地理院から電子国土賞（PC 部門）、また同年 GISA から GIS 学会賞（ソフトウェア・データ部門）を受賞しました。関係者の皆様に、改めて心から謝意を表します。

効果的な地理情報表現には地図学が大きな役割を果たしますが、この分野について長きに渡りご助言を賜っている森田喬名誉教授（法政大学）に心から感謝致します。森田先生との交流を通じて、アメリカにおける地理情報標準検討委員会の委員長として Spatial Data Transfer Standard (SDTS) の制定に尽力され、さらに、長きにわたり国際地図学会で標準化委員会の委員長を務められた、 Harold Moellering 教授 (The Ohio State University) とお近づきになることができました。Moellering 先生が専門とする解析地図学の知見は、本研究開発の基盤の一つになっています。また、地図学会誌『地図』に投稿した論文が 2017 年度に論文賞をいただきましたが、関係者の皆様に、改めて心から感謝致します。

本研究にあたっては、開発しているプロトタイプを実際に教育の場で使用し、改良を加えてまいりました。中央大学で経験させていただいた半期ずつ2カ年（2014-2015）の実践においては、久保田光一教授はもとより、今井桂子教授、田口東教授に数多くのご指摘をいただき、教材の改良を行うことができました。また、2016 年に開始した社会人を対象としたセミナー (GITEC) については、機会を与えてくださっている GISCA の現幹事長大伴真吾氏（朝日航洋株式会社）及び関係者各位に感謝致します。その運営にあたっては、武蔵野美術大学の新宿サテライト教室を3回、合計8日間にわたりお借りすることができましたが、快く許可して下さった大学当局及び、清水恒平准教授に感謝致します。それ以前、東京大学工学部において、2008 年から半期ずつ3カ年にわたって担当させていただいた、地理情報標準の概論を主題とする授業において、学習支援ツールの必要性を痛感したことが、ソフトウェア開発の動機になりました。非常勤講師のお声がけを下さった貞広幸雄教授（東京大学）に心から感謝致します。

ところで、GISA に設けられていた空間 IT-SIG と東京大学空間情報科学研究センターは共同で、2003年から空間情報規格スタジオと銘打ち、地理情報標準の社会人教育を行い、2006 年度に、GIS 学会賞（教育部門）を受賞致しました。そのまとめとして、代表をお務めだった有川正俊先生と幹事長だった私が監修・執筆にあたり、2007 年に『GIS のためのモデリング入門』という解説書を上梓致しました。この体験も本研究の礎となっています。執筆に加わってくださった石井邦宙氏、岡田泰征氏、落合修氏、黒川史子氏、郡司哲也氏、高橋亜依氏、田端謙一氏、政木英一氏、溝淵真弓氏、山本尉太氏、そして当時すでに株式会社ジャスミンソフトの代表取締役としてご活躍の贄良則氏に、改めて心から謝意を表します。

本研究には、私が所属している国際航業株式会社の皆様のご理解とご協力が不可欠でした。代表取締役社長土方聡氏、前社長である中原修氏、大島洋志氏、竹本孝氏、中島円氏をはじめ、数多くの皆様に感謝致します。そして、親会社である日本アジアグループ株式会社代表取締会長兼社長の山下哲生氏、同社取締役で国際航業会長もお務めの呉文繍氏に心から感謝申し上げます。

さらに、長年私を支えてくれた家族に感謝します。特に、2011 年 2 月に他界した妻恭子は、結婚以来30 年間にわたって、息子や娘とともに、明るい家庭を築き上げてくれました。そして、恭子の生前に結婚した息子夫婦には、その後、孫が授かりました。恭子に感謝の気持ちを直接伝えることができないことが、残念でなりません。

これまでの人生において、感謝すべき方々は、まだまだたくさんおられます。お世話になった全ての皆様に心から感謝の意を捧げます。そして今後とも、よろしくご指導ご鞭撻のほど、お願い申し上げます。

2019 年3月

Contents

Abstract

This thesis aims to propose a learning assistance tool on geospatial information technology (GIT). For that purpose, we will propose the new conceptual structure of GIT, introduce the assistance tool learning about that structure, report the teaching practices of introductory courses using that software, and validate the result of those practices. Additionally, we will propose the schemata for three applications in order to confirm that the modeling technology in GIT is useful for developing applications regarding urban engineering.

Currently, in many countries, the use of geospatial data has become easier than before, as the development of spatial data infrastructure (SDI), promotion of open data policy, and standardization of geospatial data exchanges are progressing. In addition, lower price of commercial software and easy access to open freeware became possible, and the development of geospatial applications is rather easier than before. Meanwhile, object-oriented modeling technology adopted in the international geographic information (GI) standards is used in order to respond to requests on information sharing based on international cooperation, for example the International Hydrographic Organization (IHO), the World Meteorological Organization (WMO), and the United Nations Food and Agriculture Organization (UNFAO). In Japan, Geospatial Information Authority (GSI) published the Japan Profile of GI Standards (JPGIS), and GSI is promoting its use. These facts indicate that GI standards will be used more in the near future in the field of GIT. On the other hand, the number of engineers that can make geospatial objects are not enough, and specialists who can develop applications in accordance with GI standards are also insufficient. One of the reasons is that opportunities to learn data acquisition and application development based on GI standards are quite few at least in Japan. Moreover, related teaching materials are also not sufficient.

For the development of teaching material, we need to have a metamodel to illustrate comprehensive architecture. Most of the Geospatial Information Science and Technology Body of Knowledge (GIS&T BoK) is superior in terms of it covers knowledge in detail, however it has not been provided as a metamodel for application development. Therefore, based on previous studies that the author has done (Ota 2014; Ota and Plews 2015; Ota 2016, 2017), we first will try to schematize the conceptual structure of GIT, which is a metamodel for the learning assistance software tool. For that purpose, we refer to four different models in combination. They are 1) the four-level metamodel hierarchy used as

a reference model in the ISO GI standards (ISO/TC 211 2014a), 2) the deep structure and the surface structure of cartographic data, proposed from the viewpoint of analytical cartography (Moellering 1980; Nyerges 1991), 3) the management structure for geospatial objects with metadata, and 4) a group of consistent GI standards provided by ISO/TC 211. This makes possible that knowledge system of GIT is expressed as a group of implementable schemata. In addition, we construct the consistent conceptual model by solving problems including inconsistency between knowledge areas that were discovered during the process of software development. These facts will be advantages of our conceptual model.

Next, GIT's learning assistance tool called gittok, which refers to the conceptual model of GIT is introduced. Currently, gittok consists of 40 software components. The purposes of development are for assisting learning about GIT, and to confirm that the GI application implementation is possible based on the conceptual model. We verified that gittok satisfies the capability required in the introductory course of GIT by using three test items of functional suitability specified in ISO/IEC 25010 (ISO/IEC 2011). Incidentally, in 2014, gittok received the GISA award (software and data division) from the GIS Association of Japan, and the electronic home land award (PC division) from Geospatial Information Authority of Japan.

Then, we introduce the GIT's introductory education practices using gittok. Two terms from fiscal 2014 to 2015 at Chuo University, and three times from fiscal 2016 as introductory course of Geospatial Information Technology Education Course (GITEC) sponsored by the GIS Certification Association (GISCA) have been practiced. Prior to these experiences, the author had conducted semester courses mainly on the introduction to GI standards at The University of Tokyo for three terms from fiscal 2008 through 2010. Gittok did not exist during this period yet. However, that experience encouraged the author to develop a GIT learning assistance software tool.

Furthermore, Kirkpatrick's four-level evaluation model that has been widely used (Kirkpatrick 1979; Rouse 2011) was applied to validate the effectivity of GITEC introductory course. We validated the effect of education up to 3 levels, they are reaction, learning, and behavior. The fourth level (results) is the stage of examining how much profit the organization to which the student belongs is obtained by using the acquired knowledge. However, the purpose of GIT education is to introduce basic knowledge and encourage students to learn GIT deeper. Therefore, we did not evaluate 'results', as a direct influence on the organization's interests is out of scope for education. Against the anonymous questionnaire immediately after the course, 63% of the students responded “GITEC is interesting, acquired knowledge is useful, and I want to learn GIT deeper”. And 29% of respondents answered “GITEC is interesting and useful.” [Reaction]. Next, according to the tests conducted before and after the course, it was confirmed by the t-test that there was a significant gain of knowledge (20 points increase in average value, 16.6 points or more for 95% confidence interval) [Learning]. In addition, we conducted the follow-up bearer questionnaires in April 2018 for all students who completed the course since January 2016. The collection rate was 66%, and the respondents responded that the acquired knowledge is useful (64%) or somewhat useful (32%) [Behavior]. According to these results of validation surveys, we may conclude that introductory course using gittok have certain effectivity.

Meanwhile, application schema illustrated by Unified Modeling Language (UML) can help to understand laws and rules at a glance. However, it seems that there have been few trials to represent specifications by application schema in the field of urban engineering. Therefore, in order to show that the modeling technology in GIT is useful for reviewing of application specification, we introduce an application schema for the confirmation of frontage requirements, and an application schema to find the buildable surface restricted by the slant line regulation. Then we propose the requirements for improving the Guidance for Urban Planning GIS published by the Ministry of Land, Infrastructure and Transport of Japan in 2005.

Finally, we conclude this research. We first proposed the conceptual structure of GIT, which is consistent enough for implementing GI applications. Next, we developed the learning assistance tool on GIT, which is called gittok. It also can be seen as an example of GI application in accordance with the conceptual structure of GIT. Then, gittok was examined by the education practices, and it was confirmed that introductory courses using gittok have certain effectivity. In addition, we introduced three application schemata in order to confirm that the modeling technology is useful for specification review for the field of urban engineering. However, gittok is still under development. Efforts to improve the performance of gittok through education practices are required.

Abbreviations

APA	Association of Precise survey and Applied technology
BSA	Building Standards Act
BSAEO	BSA Enforcement Order
BSAER	BSA Enforcement Regulation
BoK	Body of Knowledge
CPA	City Planning Act
CRS	Coordinate Reference System
CSMF	Conceptual Schema Modelling Facilities
CSS	Cascading Style Sheet
DBMS	Data Base Management System
DGIWG	Defence Geospatial Information Working Group
DoLETA	U.S. Department of Labor, Employment and Training Administration
EU	European Union
GFM	General Feature Model
GI	Geospatial Information
GIM	General Instance Model
GIS	Geographic Information System
GIS&T	Geospatial Information Science & Technology
GISA	Geographic Information Systems Association of Japan
GISCA	GIS Certification Association
GIT	Geospatial Information Technology
GITA	Geospatial Information & Technology Association
GITEC	GIT Education Course
gittok	Geospatial Information Technology TOol Kit
GLM	General List Model
GML	Geography Markup Language
GMM	General Metadata Model
GNSS	Global Navigation Satellite System
GPM	General Portrayal Model
GSI	GeoSpatial Information authority of Japan

GTCM	Geospatial Technology Competency Model
GUI	Graphical User Interface
HMMG	Harmonized Model Maintenance Group
HTML	Hyper-Text Markup Language
ICA	International Cartographers Association
IHO	International Hydrographic Organization
INSPIRE	Infrastructure for Spatial Information in the European Community
ISO	International Organization for Standardization
IT	Information Technology
ITS	Intelligent Transportation System
IUGG	International Union of Geodesy and Geophysics
JGD	Japan Geodetic Datum
JIS	Japanese Industrial Standards
JPGIS	Japan Profile of GI Standards
JSON	JavaScript Object Notation
JSPS	Japan Society for the Promotion of Science
KML	Keyhole Markup Language
LINKVIT	Leveraging INspire Knowledge into Vocational Innovative Training
LSD	Label Style Dictionary
MDE	Model-Driven Engineering
MIC	Maximum Inscribed Circle
MMS	Mobile Mapping System
MOOC	Massive Online Open Course
NoSQL	Non SQL
NRC	National Research Council
NSDI	National Spatial Data Infrastructure
OER	Open Educational Resources
OGC	Open Geospatial Consortium
OMG	Object Management Group
OSM	Open Street Map
PI	Place Identifier
RGB	Red Green Blue
RISE	Reference Information Specification for Europe

RM	Real Map
RMSE	Root Mean Square Error
SDI	Spatial Data Infrastructure
SG	Spatial Geometry
SQL	Structured Query Language
SQualRE	Systems and software Quality Requirements and Evaluation
SSD	Symbol Style Dictionary
SVG	Scalable Vector Graphic
UAV	Unmanned Aerial Vehicle
UCGIS	University Consortium for Geographic Information Science
UML	Unified Modeling Language
UP-GIS	Urban Planning GIS
URL	Uniform Resource Locator
US	United States
UTC	Coordinated Universal Time
UTM	Universal Transvers Mercator
UX	User Experience
VGI	Volunteered Geographic Information
VMT	Virtual Map Type
VR	Virtual Reality
W3C	World Wide Web Consortium
WMO	World Meteorological Organization
XML	Extensible Markup Language

1. Preface

1.1. Introduction

In this thesis, we introduce a software tool assisting the learning about GIT that has been rare so far. Its effectiveness was validated by the practices of introductory education using this tool. Then, we will introduce three application schemata regarding urban engineering to show that GIT is useful for this research domain. In this chapter, as a preface of this thesis, the background and motivation of this research will be described, then the primary contributions of this research will be introduced.

1.2. Background and motivation

Today, acquisition, sharing and reuse of geospatial data are becoming easier than before, mainly because of the reduced costs and availability of software and hardware such as the Global Navigation Satellite System (GNSS), the Unmanned Aerial Vehicles (UAV), sensors such as the laser scanner and the multi-band digital camera, and mobile devices such as the Mobile Mapping System (MMS) and smart phones. In addition, open data policies are promoted by governments, and the use of free GIS packages as open-source software are becoming very widespread. These facts are evidences in the enhancements to ubiquitous information services using maps, air-photos, satellite images, three-dimensional images, Volunteered Geographic Information (VGI) (Goodchild 2007) and the Spatial Data Infrastructure (SDI).

Meanwhile, in short, GIT is a technology for modeling phenomena occurring and disappearing in the real world, analyzing the model, and visualizing the results for decision making. It is an interdisciplinary field formed mainly by information technology (IT) and GI science. IT contains object oriented technology, database, telecommunication, computer graphics, and so on. GI science contains geodesy, cartography, hydrography, surveying, photogrammetry, remote and close-range sensing, and so on. Moreover, unique research and development is being carried out in applied fields of GI science such as geography, geomorphology, geology, meteorology, oceanography, urban engineering, civil engineering, environmental studies, sociology, economics, disaster prevention, crime prevention, lifeline management, and so on. In fact, the scope of GIT and related knowledge areas are vast and more and more expanding because of the rapid progress of research and development.

Therefore, we need to reconsider the structure of knowledge about GIT. This was the reason that Geographic Information Science and Technology (GIS&T) Body of Knowledge (BoK) were proposed by experts from the USA (DiBiase et al. 2006), Europe (Painho et al. 2007), South Africa (Du Plessis and Van Niekerk 2014) and so on. In addition, we should keep in mind that the NCGIA Core Curriculum in GIS (Goodchild and Kemp 1990) was widely adopted in universities and colleges in the USA (Heywood, Cornelius, and Carver 2011).

In Japan, the Core Curriculum was translated in Japanese by Kubo et al. (Kubo 1993), and it has given significant influence to the Japanese education on GI science. The initiative on Japanese GIS&T BoK was undertaken by the research project "Development of geographic information science curricula and sustainable web library systems for serving their contents" [Chair: Atsuyuki Okabe, The University of Tokyo] from 2005 up to 2007. The Core Curriculum, the Strawman Report, and other materials such as the GI standards provided by International Organization for Standardization Technical Committee 211 (ISO/TC 211) were referred for this research. Also, the research project "Geographic Information Sciences Education and Spatial Thinking" [Chair: Yasushi Asami, The University of Tokyo] was performed from 2009 to 2013. International conference on spatial thinking and geographical information sciences was held from September 14 to 16, 2011 (Asami 2011). Moreover, the research project "Development of open-access e-learning material for GIS education based on the existing core curriculum and body of knowledge" [Chair: Takashi Oguchi, The University of Tokyo] is running from 2015 to 2019. The Japan Society for the Promotion of Science (JSPS) has been funding all projects. The author has been a member of these three projects.

Meanwhile, geographic information standards have been investigated by ISO/TC 211 – Geographic information/Geomatics since 1994, and Japan has actively contributed as a participant member with voting rights. The author had contributed as an expert of ISO/TC 211 since 1995 till 2005, and has been a member of Japanese domestic committee to ISO/TC 211 since 2006. More than 70 standards have been enacted by 2018. By many international organizations and countries, the standards are being used as metamodels to create domain specific standards and specifications. In fact, there are many examples such as the standard of geographic data files (GDF) for Intelligent Transportation System (ITS) by ISO / TC 204 (ISO/TC 211 2011a), the universal hydrographic data model (S-100) by IHO (IHO 2017), CityGML for the three-dimensional modeling of a city provided by Open Geospatial Consortium (OGC) (Gröger et al. 2012), and IndoorGML for indoor navigations also provided by OGC (Lee et al. 2018.), and so on.

In Japan as well, the international standards provided by ISO/TC 211 have adopted as the Japanese Industrial Standards (JIS). The Japan profile of geographic information standards (JPGIS) as a subset of these standards (http://www.gsi.go.jp/ENGLISH/page_e30210.html accessed 2018-04-20) was released from the Geospatial Information Authority of Japan, and the data product specifications conforming to JPGIS for application domains are widely used.

Furthermore, European Union (EU) published the Directive 2007/2/EC for establishing an Infrastructure for Spatial Information in the European Community (INSPIRE) (EU 2007). Japanese government enacted "Basic Act on the Advancement of Utilizing Geospatial Information" in the same year (Murakami 2008). These laws encourage to use GI standards for multi-purpose utilization of geospatial data.

Meanwhile, according to Geospatial Information & Technology Association (GITA),

> *"Approximately 70 to 80 percent of the information managed by business is somehow connected to a specific location—an address, street, intersection, or "xy" coordinate. Therefore, geospatial technology is finding its way into every corner of the business world. And, because the technology's uses are so widespread and diverse, the geospatial market is growing at an annual rate of almost 35 percent. The commercial subsection of this market is expanding at a phenomenal rate of 100 percent each year."*
>
> (http://www.gita.org.au/geospatial.php, accessed 2018-01-10)

In addition, in accordance with Geospatial Technology Competency Model (GTCM) provided by United States Department of Labor (DiBiase et al. 2010), workers in the industry sector are expected to have competency of software and application development.

These facts indicate that the education in terms of the number of persons who have knowledge of GIT is a pressing problem for effective developments of GI applications.

Several worthy efforts to provide education on the GI standards can be identified. There is the report on 'Teaching GI standards in higher education' (Reinhardt 2005). The advisory group on outreach in ISO/TC 211 published the Standards Guide (ISO/TC 211 2009). However, the academic environment for learning GIT and GI standards seems to be not yet matured enough. In the case of Japan, several developments of GI systems and educations, for example (Tani 2009; Hatayama et al. 1999), have been conducted and each has proved their achievements. However, they do not take into account GI standards deeply. In the case of EU, LINKVIT project, for example, developed the comprehensive education course as a vocational training course for professional engineers and postgraduates mainly focused on the topics such as SDI, INSPIRE, geoportals, data harmonization and open data (Vancauwenberghe and Vandenbroucke 2015). LINKVIT is an abbreviation of 'Leveraging INspire Knowledge into Vocational Innovative Training'. This education is worthwhile for the dissemination of the knowledge required to realize the SDI in compliance with INSPIRE directive. However, it seems to be provided for professionals who already have knowledge of GI applications, and not for beginners, for example, students in higher education institution.

The basic forms of education are lectures and exercises. Especially hands-on exercises using learning assistance tools are effective to get the skill. Therefore, in order to learn GIT, the assistance software tool is essential. However, at least, based on the Internet research, the author could not find a software tool aimed specifically at introducing the knowledge of GIT based on GI standards (Ota and Plews 2015).

Therefore, this thesis aims to provide the all-in-one software for beginners to assist for learning about GIT based on GI standards. The software tool developed for learning about GIT is called 'gittok' (Geospatial Information Technology TOol Kit). 'Gi' in gittok is pronounced 'ji' such as giraffe, ginger, Gibraltar, Gina and Giovanni.

1.3. Contributions

The primary contributions of this research are 1) to propose the conceptual structure of GIT, 2) to introduce the learning assistance tool of GIT based on its conceptual structure. This tool was validated its effectivity by the four-level evaluation model. In addition, 3) we will propose three application schemas regarding the urban engineering in order to confirm that GIT is useful for developing GI applications.

1.3.1. Schematization of GIT

The first major contribution of this research is to propose the new and original conceptual schema of GIT which is useful as a basis of GI applications. Schema means a formal description of a model (ISO/TC 211 2014a) described in the conceptual schema language. We use Unified Modeling Language (UML) for schematization in this thesis, as ISO/TC 211 uses UML for the formal description of GI standards. There have been few previous researches to investigate the conceptual schema of GIT that is used for designing a GI application architecture. In this thesis, the conceptual schema of GIT is called the concept map. It is modeled in accordance with 1) the four-level metamodel hierarchy, 2) the deep structure and the surface structure of cartographic data, 3) the management structure for geospatial objects with metadata, and 4) GI standards provided by ISO/TC 211. The resulting knowledge system is specified in detail by UML class diagrams and unique function diagrams, so that the knowledge contained in GIT and their relationships can be systematically understood. Most of the GIS&T BoK are well structured descriptions of knowledge areas, however it does not specify the metamodel for application developments. Therefore, the concept map of GIT is a breakthrough idea for using GIT BoK as a design pattern for GI applications.

In addition, we make three unique contributions to reinforce the organization of GIT structure. The first is to clarify 'modeling' as the knowledge area in GIT. The term 'model' is defined as an abstraction of some aspects of reality (ISO/TC 211 2015b). Modeling in this thesis is a function to define geospatial features and their relationships as a conceptual model. The second is that we introduce a feature without geometric attributes that is called 'virtual feature'. It enables to extend the capability of GI applications, for example, by one dimensional map called a list. The third is to propose the consistent conceptual structures for designing of application components by introducing new metamodels, for example, the general portrayal model (GPM), which were not clearly described in the GI standards.

The merits of a conceptual structure of GIT proposed in this thesis are shown as follows.

- It is useful to understand the whole image of GIT.
- The paradigm of GIT is expanded by the adoption of an object-oriented technology.
- Consistent application development and data acquisition will be possible, because the entire structure of GIT introduced in this thesis is modeled under the consideration of the consistency between different knowledge areas.

The originalities of a conceptual structure of GIT proposed in this thesis are shown as follows.

- Virtual feature without spatial attributes, instance models for gazetteer and list are introduced to realize the consistency of the structure.
- One dimensional geospatial representation by a list is introduced to expand the paradigm of GIT.
- Metamodels such as the general metadata model, general instance model, general portrayal model

and general list model are defined to add the completeness of the structure of GIT.

- Difference between cartographic objects and cartographic products are clarified in the structure of GIT. Cartographic product is a final output of GI application systems.

There have been few approaches referring to GI standards critically for the consideration of the architecture of GI applications. The reason seems to be that the standards have only been understood as the rules for geographic data interchange. However, according to the scope of ISO/TC 211, the standards may specify, for geographic information, methods, tools and services for data management (including definition and description), acquiring, processing, analyzing, accessing, presenting and transferring such data in digital/electronic form between different users, systems and locations (https://www.iso.org/committee/54904.html accessed 2018-07-03). Therefore, we may consider the architecture of gittok by referring to GI standards.

1.3.2. Software tool for learning about GIT

The second major contribution is to provide GIT learning assistance software called gittok as an application system based on the conceptual map of GIT. The current status of introductory education on GIS is typically ‘learning with general-purpose GIS’, however gittok is an all-in-one software specialized for ‘learning about GIT’ (Ota and Plews 2015). Because, there are few applications specialized for the introductory course of GIT. As already mentioned, the society needs the experts who have skill to develop and maintain the GI applications including general-purpose GIS, SDI and software for specialized purpose referring to the GI standards.

1.3.2.1. Expected merits from learning about GIT

We can expect merits from learning about GIT at least for five types of persons. At first, as gittok is a tool aiming at grasping the overall picture of GIT, it will provide merits for <u>students</u> who want to understand this field. In this research, it was confirmed that students can obtain certain knowledge by educational experience at Chuo University (Ota and Plews 2015). The detail report is found at the section 9.3.

On the other hand, this education also brings merits that <u>engineers</u> can improve their knowledge by confirming the overall picture of GIT that is based on GI standards. For example, the GI standard specifies that the application schema included in the data product specification shall be described with

UML as the conceptual schema language (ISO/TC 211 2007b). Learning the modeling knowledge in the introductory course will provide basic skill to attendees. In fact, most of the people who attended the education course organized by GISCA gained their knowledge. It was confirmed by the comparison between pre-test and post-test (Ota 2017). The verification of the education for developers is described in detail at Chapter 9 and Chapter 10.

In addition, it will be beneficial for researchers who are interested in "Learning about GIT" to reconfirm their own knowledge, and to use gittok as one of the teaching materials. However, we should seek additional opportunities to evaluate this learning, and we should improve not only gittok but also slides and texts for lectures and exercises.

In the meantime, collaboration of experts from multiple fields is required in order to solve today's social demands that are becoming complicated. The family of GI standards are published to be the common language for experts to consider the specifications and dataset acquisition for NSDI and GI applications. Members in the collaboration work should know the common language by learning about GIT. The learning material such as gittok will be able to contribute to lead easier understanding of such a skill.

Finally, decision-makers need to measure the investment effect of the project. However, in order to do the prediction and evaluation, they are required to have literacy on GIT. It will be preferable to enlighten them by using an appropriate learning assistance tool in order to get the literacy.

1.3.2.2. Gittok

As already mentioned, the purpose to develop gittok is to provide an assistance tool for the introductory education for learning about GIT. It is difficult to learn a whole image of GIT without exercises using a learning assistance software. Because, most of the knowledges on GIT are ‘procedural’. However, it is not possible to obtain sufficient knowledge by taking an introductory course only, but there is no doubt that this is the first step to gain knowledge deeper. Therefore, gittok aims that student are able to get the desirable experiences and they will have minds that they want to obtain further in-depth knowledge, after taking the course.

As unique characteristics of gittok, all knowledge areas in GIT are separated in components under the main component (title page), and they are divided into sub-components those represent knowledge units. Currently, the total number of sub-components is 40. This hierarchical structure helps learners grasp where they are learning. Moreover, since the consistency of data exchanged between components is ensured, there is no need to take time for converting the format of data, and no missing of information occurs. At least by the Internet research, we could not find similar learning assistance software.

Actually, "learning about GIT" will be possible by the computer science/information-related departments will collaborate more actively with geography-related departments (Kawabata et al. 2010). Gittok aims that student are able to get the desirable experiences and they will have minds that they wish to obtain further in-depth knowledge related to IT and geography. Therefore, gittok can be expected to be a breakthrough toward the comprehensive GIT education realized by multidisciplinary collaboration between geography related departments and IT-related departments.

1.3.2.3. Validation of education effectivity

The third major contribution is that the education method using gittok was validated its effectiveness. The education practicing with gittok has been confirmed through the semester-long courses organized by Chuo University (2014 - 2015) and the open seminars held by GIS Certification Association (GISCA) in Japan (2016 -). The author took charge of these education classes as the lecturer. The introductory courses by GISCA called Geospatial Information Technology Education Course (GITEC) were held three times from January 2016 to May 2017. The length of each course was 3 days (21 hours). Almost all students are beginners of GIT and engineers working at the GIS related industry.

We used four-level evaluation model (Kirkpatrick 1979) for the evaluation of GITEC introductory course. This evaluation model is one of the well-known learning evaluation approaches. According to this model,

i. Reaction (how learner feel about instruction),
ii. Learning (learner's performance on in-class test),
iii. Behavior (acquired knowledge are used or not after graduation), and
iv. Results (organizational benefit such as improvement of working efficiency)

are measured.

However, it is hard to measure the results such as cost reduction only by this education, because the purpose of the introductory course is that students will get the image of GIT as a preparation for learning the deeper knowledge. Therefore, we validate the education effectivity by reaction, learning and behavior.

1) Reaction

Questionnaire has been conducted to evaluate educational effects at the end of each course. 63% of students responded that the education is interesting, useful and they want to learn GIT deeper. And 29% of respondents answered that the education is interesting and useful. In fact, 55% of completed graduates participated in the intermediate level courses held after.

2) Learning

The pretest and the posttest are performed to measure learning gain. By the t-test, it was confirmed that there was a significant gain in the average of the posttest by comparing with the average of the pretest (p-value = 0, the gain of mean value is 20, 95% confidential interval is (16.6, ∞)).

3) Behavior

In April 2018, anonymous follow-up questionnaire was carried out targeting all completed graduates (38 persons). 44% of the first completed graduates responded the questionnaire, despite that the first education course was held two years and three months before. While, the total response rate was 66%. 96% of respondents (63% of all completed graduates) responded the knowledge gained by the course are useful or somewhat useful for their activity.

1.3.3. Applications of urban engineering

Modeling technology in GIT is expected to be useful to describe phenomena occurring in the urban area. We proposed an application schema to confirm the frontage requirements, and an application schema to find a buildable surface restricted by the slant line regulation. Then we proposed the requirements that should be considered when improving the Guidance for Urban Planning GIS published by the Ministry of Land, Infrastructure and Transport of Japan in 2005. These examples and proposal will be expected to be useful for improvement of applications on urban engineering.

We constructed the new conceptual structure of GIT, and based on it, the software for learning about GIT called gittok was developed, then the introductory educations were practiced by using gittok since 2014. And those courses were validated the effectivity by the three levels of evaluations: reaction, learning and behavior. Moreover, we introduced three application schemas regarding the urban engineering in order to show that GIT will be useful for developing GI applications. Through these research activities, we may conclude that gittok will bring breakthrough method for learning about GIT.

2. Schematization of GIT

2.1. Introduction

We discuss and propose a framework for conceptual schema of GIT in this chapter. We try to confirm the definition of the 'geospatial information technology (GIT)' and the 'geographic information (GI) standards' at first, because they are important keywords used in this thesis. Then, we define a matrix to show the conceptual schema of GIT, which consists of four rows by referring the four level metamodel hierarchy (meta-metamodel, metamodel, model, and object) (ISO/TC 211 2014a) and three columns by referring the structures of geospatial object in the deep structure, the structures of cartographic object in the surface structure (Moellering 1980; Nyerges 1991), and the management structure using metadata as information about a resource (ISO/TC 211 2014b). This matrix is called the 'concept map of GIT'.

In addition, each element in the concept map associates with functions exchanging objects between other elements or outside of the application domain. Functions represent knowledge areas in GIT, they are modeling, acquisition, management, analysis, exchange and representation.

Most of the BoKs are well structured description of knowledge areas, however it does not specify the metamodel for application developments. Advantage of the concept map of GIT is that it can be used as the metamodel for the specifications of GI applications and geospatial objects. Therefore, the concept map of GIT enables GIT BoK for use as a design pattern for GI applications.

By the way, we should use the terms 'data' and 'object' carefully. Generally, data is a set of values stored in the computer device. While, according to ISO 19107 (ISO/TC 211 2003a), object is,

> *"Entity with a well-defined boundary and identity that encapsulates state and behaviour [UML Semantics [17]].*
>
> *NOTE This term was first used in this way in the general theory of object oriented programming, and later adopted for use in this same sense in UML. An object is an instance of a class. Attributes and relationships represent state. Operations, methods, and state machines represent behavior."*

In the context of object-oriented technology, object is an instance of a class. Object consists of states and behaviors. Classes are extensible templates for creating objects, providing initial values for instance variables and the bodies for methods (Bruce 2002). In this thesis, state is reworded as attribute, association or inheritance. A method in a feature is reworded as an operation in accordance with UML. Operations are algorithms for various purposes, for example, spatial analysis, data format conversion, and attribute reliability evaluation. Meanings of 'data' and 'object' are sometimes vague in existing documents. Therefore, to understand the meaning of them clearly, we need to care the context around these terms.

In the following sections, we will try to confirm the definition of GIT and GI standards. Then, we will define the concept map of GIT.

2.2. GI Technology (GIT)

As far as technology is a collection of knowledge, we should consider the meaning of knowledge at first for the construction of a conceptual structure of GIT. According to the Oxford English Dictionary, knowledge is defined as "facts, information, and skills acquired through experience or education." Facts and information are declarative knowledge, while skill is procedural knowledge. Declarative knowledge represents factual information. And, procedural knowledge, on the other hand, represents 'know-how' (Runco and Chand 1995). In principle, declarative knowledge of a feature is described as attributes, while procedural knowledge is described as methods, in other words, operations.

According to the Oxford English Dictionary, technology is defined as "the application of scientific knowledge for practical purposes, especially in industry". We can replace "scientific knowledge" by "both declarative knowledge and procedural knowledge on science".

Geospatial Information Technology (GIT) is defined by DiBiase, et al. in Geographic Information Science and Technology Body of Knowledge (GIS&T BoK) as follows.

> *"The specialized set of information technologies that support data acquisition, data storage and manipulation, data analysis, and visualization of geo-referenced data."* (DiBiase et al. 2007)

DiBiase, et al. adopted the definition of GI science proposed by Goodchild as follows.

> *"The multidisciplinary research enterprise that addresses the nature of geographic information and the application of geospatial technologies to basic scientific questions."* (Goodchild 1992)

Earlier, Okabe (2006) had proposed the definition of GI science as follows.

"The study area to research general methods and methodologies to acquire, manage, analyze, integrate and communicate geographic data systematically, and simultaneously to develop general methods and methodologies to apply them in order to make them widely available." (Okabe 2006)

These definitions recognize there is a complementary relationship between GI science and GIT. Since, technology is knowledge applied by science and by application for practical purposes, the structure of GIT should be consistent to ensure its reliability. In order for technology to be useful for people, the knowledge contained in GIT should be harmonized each other as much as possible. In other words, when transforming scientific knowledge to technological knowledge, appropriate selection and tuning are required to keep consistency of the technological knowledge structure.

However, the definition of GIT introduced in GIS&T BoK do not actually clarify 'modeling' of real world phenomena. For example, the definition of GI system proposed by Burrough was as follows (Burrough 1982).

"A powerful set of tools for collecting, storing, retrieving at will, transforming, and displaying spatial data from the real world for a particular set of purposes."

There is also no clear description about 'model' in this definition, even if 'a particular set of purposes' indirectly indicates the universe of discourse that is defined as a view of the real or hypothetical world that includes everything of interest (ISO/TC 211 2014a). We cannot ignore the knowledge area for modeling, because reliable data acquisition and analysis cannot be performed without linguistic and formalized expressions such as a data specification or an application schema.

By the way, there are a few definitions of GIT including modeling. For example, Worboys proposed the definition of GI system as follows.

"a computer-based information system that enables capture, modelling, manipulation, retrieval, analysis and presentation of geographically referenced data."(Worboys 1995)

Moreover, since GI standards adopt the four-layer metamodel hierarchy based on the model driven architecture as the reference model (ISO/TC 211 2014a), 'modeling' cannot be neglected as the knowledge area of GIT. Therefore, we should define GIT as information technologies for *modeling*, acquisition, management, analysis, exchange and representation of geospatial information. We will discuss modeling in detail at the section 5.3.

This definition of GIT was brought from the viewpoint of 'function flow', because technological elements of above definition are all functions to transform from real world phenomena into geospatial objects, from geospatial objects to cartographic objects, and from cartographic objects to information representation for users. Each function is connected with others through various objects. Function is realized based on the procedural knowledge, while objects are specified based on the declarative knowledge.

In the meantime, the term 'GI system' seems to be an alternative term of GIT. However, the GI system is a computer system developed as an application of GIT. Today, GIT is applied not only in conventional GI systems such as cadaster management, facility management, urban planning, environmental protection, disaster prevention and so on, but also in many other systems such as intelligent transportation systems, restaurant guides, smart phone games, services to moving individuals at outdoor and indoor, and so on. In fact, many of developers apply knowledge of GIT in a part of their system, but so far, they do not care about that their system is GI system or not. It might mean that GIT is melt in IT. However, since GI is unique by comparing with other types of information, GIT will be positioned as a unique domain in the field of IT. We will discuss the structure of GIT more in detail at the section 2.4.

2.3. GI standards

GIT should conform to GI standards as long as it is possible. According to ISO, an International Standard provides rules, guidelines or characteristics for activities or for their results, aimed at achieving the optimum degree of order in a given context (https://www.iso.org/deliverables-all.html#IS accessed 2018-06-06). By conforming products to standards, ensuring of product compatibility and interface consistency, improvement of production efficiency, ensuring of product quality, and promotion of accurate information transmission will be possible.

ISO/TC211, OGC and other related organizations provide GI standards and specifications to realize the geospatial data interoperability. According to the scope of GI standardization in ISO/TC 211,

> *"This work aims to establish a structured set of standards for information concerning objects or phenomena that are directly or indirectly associated with a location relative to the Earth.*

These standards may specify, for geographic information, methods, tools and services for data management (including definition and description), acquiring, processing, analyzing, accessing, presenting and transferring such data in digital / electronic form between different users, systems and locations.”
(https://www.iso.org/committee/54904.html accessed 2018-01-10).

Since 1994, ISO/TC 211 has been working to provide GI standards for modeling, acquisition, analysis, manage, exchange and representation of GI. In fact, the scope of ISO/TC 211 is similar to the definition of GIT. Most of the GI standards provided by ISO/TC 211 are collectively harmonized with others, because the Harmonized Model Maintenance Group (HMMG) established in TC 211 verifies the relationships between standards. In addition, this group maintains the harmonization among schemata described as UML class diagrams included in the standards. Moreover, voting by the member countries strengthens the hypothesis that the standard is a common global rule. Consequently, we can expect that standards are the harmonized system of commonly used rules.

Standards are revised when inconsistencies or misconceptions are found, information technology useful for standard is evolved, scientific knowledge useful for the standard is found, and the social demands change. This is the reason ISO keeps the technical committee for the improvement of standards. Therefore, the GI standards are valuable in order to strengthen the structure of GIT, despite that it is necessary to see standards critically. This is the reason that the GIT should conform to GI standards.

Meanwhile, GI standards do not aim to prompt the systematization of GIT knowledge. As far as the reference model of GI standards provided by ISO/TC 211 is concerned,
“It is an underpinning for the interoperability in the area of geographic information, which aims to support the development of SDI” (ISO/TC 211 2014a).

Actually, ‘interoperability in the area of geographic information’ is one of the main issue of GIT, however it is not the whole scope of GIT. Therefore, we may consider the conceptual structure of GIT independently from GI standards, but simultaneously, we should refer to GI standards as much as possible, as they are reliable and consistent set of geospatial knowledge in principle.

2.4. Concept map of GIT

2.4.1. Introduction

In this section, we will define the 'concept map' of GIT that defines the framework of the software architecture for GI applications.

There have been few researches about the conceptual structure of GIT for designing GI applications. According to Dibiase et al., GIS&T BoK was the first attempt to provide the comprehensive BoK about GIS&T as described below.

> *"Unlike other information technology fields, including computer science and information science, there has, until now, been no collective effort by a community of researchers and educators to specify a comprehensive body of knowledge that defines the GIS&T domain. The Body of Knowledge is an attempt to fill that void."* (DiBiase et al. 2006)

In fact, it is very important to study knowledge areas and units on GIS&T. However, it is not a formalized model to clarify the conceptual structure of GIT.

Next, distinguished literatures introducing GI systems have been published, for example (Aronoff 1989; Worboys 1995; Chrisman 2002; Duckham, Goodchild, and Worboys 2003; Longlay et al. 2015). Although these books are effective as textbooks for GI systems and sciences, they do not care a comprehensive description for conceptual design of GI applications. In order to design the GI application, the generic reference model should be used, and it should designate requirements for functions and data in GI applications.

Reference model is a framework for understanding significant relationships among the entities of some environment, and for the development of consistent standards or specifications supporting that environment (ISO/TC 211 2014a). The four-layer metamodel hierarchy is a general-purpose reference model, and it can be applied as a norm to consider the conceptual structure of GIT. We will discuss the four-layer metamodel hierarchy in the section 2.4.2.

Furthermore, to consider the design of GI applications, we can apply the deep structure and the surface structure of GI, those were proposed from the viewpoint of analytical cartography (Nyerges 1991;

Moellering 1980). The combination of these structures is a model originating from Chomsky's proposal (Chomsky 1968) as concepts used in linguistics. For example, the sentences "He loves you" and "You are loved by him" mean almost same thing and use similar words. According to Chomsky, these two sentences are distinct surface forms that derive from a common deep structure. Cartographic products in the surface structure such as maps and charts are direct representation for users' understanding, while geospatial data in the deep structure are used to produce different representations. Indeed, 'Make once, use many times' is the principal reason to establish the SDI and to promote the open data policy by many countries. However, Nyerges and Moellering proposed to use these structures before 1990s. Therefore, it is reasonable to redefine them in consideration of the technical environment today. We will discuss the deep structure and the surface structure in the section 2.4.3.

Management structure announced as Executive order 12906 in the United States in 1994 was the concept of SDI to improve the efficiency of GI management. It has been accepted in many countries including Japan as the National Spatial Data Infrastructure (NSDI), geo-catalogue, and distributed geo-library, since the middle of 1990s. Therefore, the management structure should be considered as a part of GIT. We will briefly introduce the management structure in the section 2.4.4. Then, it will be discussed in detail at Chapter 7.

In the following sections, we will discuss the reference model, the surface structure and the deep structure, and the management structure as axes of the conceptual structure of GIT. Then, we will propose the concept map of GIT as a table of which columns are defined by the four-layer metamodel hierarchy, and rows are defined by the management structure, the deep structure and the surface structure. We will discuss the concept map of GIT in the section 2.4.5.

2.4.2. Reference model

Fortunately, we can apply the four-layer metamodel hierarchy included in the Conceptual Schema Modelling Facilities (CSMF) (OMG 2003) as the methodology of reusable model designing. According to ISO 19101-1:2014 – Geographic information – Reference model – Part 1: Fundamentals (ISO/TC 211 2014a), four-level decomposition approach, in other words the four-layer metamodel hierarchy is adopted as the reference model for ISO/TC 211 family of standards. The four-layer means meta-metamodel, metamodel, model and object.

For modeling the geospatial information domain, we interpret this hierarchy as shown in Figure 2.4-1. Domain is an application extent of this reference model. The management structure, the deep structure, and the surface structure are domains for GIT. The top of this hierarchy is the run-time objects as instances of model elements. A model is an instance of a metamodel to define languages that describe application domains. A metamodel is an instance of a meta-metamodel to define a language for specifying models. Finally, a meta-metamodel is a universal language for specifying different metamodels.

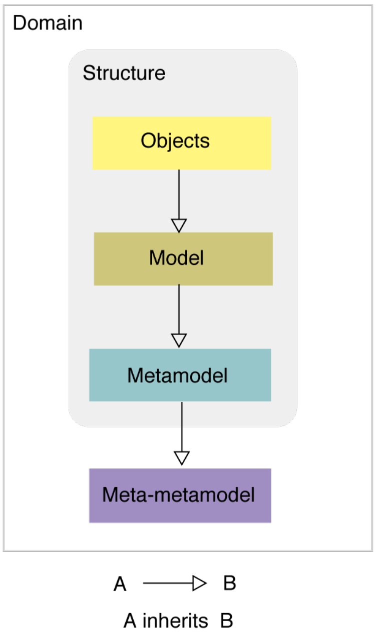


Figure 2.4-1. Reference model for conceptual structures of GIT

WMO, for example, recognizes importance of the four-layer metamodel hierarchy for the development of standards (WMO 2008.). And it explains the four-layer metamodel hierarchy as follows.

> *"Conceptual modelling is critically important to the definition of the ISO 191xx series. The approach to conceptual modelling in the ISO 191xx series is based on the principles described in the Conceptual Schema Modelling Facilities (CSMF). The ISO CSMF includes four model levels: Meta-metamodel, Metamodel, Application model and data. The Unified Modelling Language (UML) is the basis for the conceptual schema language (ISO 19103) used within the ISO 191xx series."*

As a consequence, we decided to use the four-layer metamodel hierarchy as a reference model of the conceptual modeling for GIT.

2.4.3. Surface structure and deep structure

The four-layer metamodel hierarchy can be applied as a norm to consider the structure of GIT as it is a general-purpose model. We can apply the deep structure and the surface structure of GI proposed from the viewpoint of analytical cartography. It is reasonable to redefine these structures in the consideration of technical environment today and clarify as part of GIT as already mentioned.

Nyerges (Nyerges 1991) inspired by Chomsky (Chomsky 1968), Tobler (Tobler 1979) and Moellering (Moellering 1980) proposed the surface structure and the deep structure for geospatial and cartographic objects as follows.

Surface structure is the organization of simple information chunks immediately perceptible to a user of a map. The simple information chunks are the individual graphical symbols on a map display, as well as database elements stored in a data base. Deep structure is the organization of the simple chunks that often provides the meaning of the information chunks in surface structure. The deep structure contains the conceptual information, as the basis of meaning, as well as the various geographical relationships among chunks that form the basis of abstractions of reality.

Schemata for chunks of surface structure on maps, are rules that organize symbols as they appear individually on the map display and as they are stored individually in a data base. For deep-structure chunks, the schemata organize the conceptual information associated with the surface structure in terms of the meanings of database/knowledgebase elements and their geographical relationships.

Cartographic products such as maps and charts are direct representation for users' understanding. While geographic data included in the deep structure associate with meanings of the real-world phenomena. And different cartographic products may be reproduced by the same geographic data.

The surface structure for mapping was the rules for symbols and labels according to the proposal by Nyerges. He picked up the surface structure for 'map data' only. However today, not only maps but also atlas, geographic name dictionary, gazetteer, virtual reality, augmented reality, walk-through video, audio guidance, and so on, are the products derived from different kinds of data in the surface structure. Therefore, in this thesis, we use the term 'cartographic objects' instead of 'map data'. And we use 'cartographic product' instead of 'map'. Moreover, we call the rules for describing the cartographic objects as a portrayal schema by referring to GI standard.

Before Nyerges (1991), Moellering proposed a four-class concept of Real and Virtual Map Types (Moellering 1980). This concept is useful for the harmonization between functions those transform geospatial objects to cartographic objects. Moellering defined Real and Virtual Map as follows.

Real Map - Any cartographic product, which has a directly viewable cartographic image and has a permanent tangible reality (hard copy). There is no difference as to whether that real map was produced by mechanical, electric, or manual means.

Virtual Map, Type 1 - Has a directly viewable cartographic image but only a transient reality similar to a map image on a display device.

Virtual Map, Type 2 - Has a permanent tangible reality but cannot be directly viewed as a cartographic image. These are all hard copy media, but in all cases these products must be further processed to be made viewable.

Virtual Map, Type 3 - Has neither of the characteristics of the earlier types but can be converted into a real map as readily as the other two types of virtual maps. Computer-based information in this form is usually very easily manipulated.

When interpreted these definitions by today's technology and discussions above, the definitions can be reworded as follows.

- Real Map (RM) - Cartographic product processed from cartographic objects for visualization as, for example, paper map, atlas, geographic name dictionary, gazetteer, virtual reality, walk-through video, audio guidance, interactive map on display device such as a screen of PC, smart phone, VR headset and so on.
- Virtual Map Type 1 (VMT1) – Cartographic objects transformed from geospatial objects, for visualizing cartographic product.
- Virtual Map Type 2 (VMT2) – Standardized cartographic objects transformed from original cartographic objects (VMT1) in compliance with the GI standards for exchange objects between different systems.
- Virtual Map Type 3 (VMT3) - Geospatial objects captured by observation from the real world, or obtained by existing materials which reflect the real world.

However, as a matter of fact, to provide standardized geospatial objects for exchange is one of the main reasons to establish the GI standards. Therefore, standardized geospatial objects for exchange as Virtual Map Type 4 (VMT4) (Ota and Plews 2015) should be added as follows.

- Virtual Map Type 4 (VMT4) – Standardized geospatial objects with the system independent format in compliance with GI standards, which is produced by transformation from the original geospatial objects.

Nyerges (1991) also mentioned that RM and VMT1 are associated with the surface structure and VMT3 is associated with the deep structure. However, he did not mention the position of VMT2.

While, Moellering stated "*in all cases these products must be further processed to be made viewable*". Therefore, it should be placed between different systems, because VMT2 is obtained by transformation from VMT1 to be made it viewable in the different system environment.

A schema for chunks of deep structure is equivalent for application schema in the context of GI standards, because an application schema is a formal description of the universe of discourse. While a schema for chunks in the surface structure is similar to a portrayal schema, because these schemata are formal description of requirements for representation.

According to the definition of GIT, it is seen as a collection of functions from the real world to information such as standardized geospatial objects, standardized cartographic objects, and cartographic product. Cartographic product is an outcome of geospatial information processing appearing in the real world. As well as the real world is not in the deep structure, cartographic product which is tangible in the real world is not included in the conceptual structures of GIT.

2.4.4. Management structure

Metadata is described to express information about the geospatial object, and they are described in compliance with the metadata schema. The metadata schema is a domain specific rule to give information about geospatial objects. Therefore, its definition should be harmonized with the application schema. Today, many types of metadata are used in the different fields such as catalogues of digital libraries and museums, digital photographs, documents produced by word processors and so on. Geospatial metadata is used for retrieving and evaluating geospatial objects stored in the repository. A set of metadata associating with the data files is managed in the management structure. Metadata is created in accordance with the metadata schema. However, there is no specific definition of a metamodel for metadata schema in ISO 19115-1 (ISO/TC 211 2014b). To avoid ambiguity in the domain specific metadata schema, the metamodel for metadata schema is quite useful. Therefore, we will propose the general metadata model (GMM) in the section 7.2. And the geospatial metadata schema used in gittok is introduced in the section 7.3.

The management structure describes a geo-library. It is a system to acquire, process, store, distribute, and improve utilization of geospatial data (U.S.Government 1994). A geo-library is also called a geo-catalogue or a geospatial clearinghouse. In this thesis, we use 'geo-library'. Functions of geo-library are discussed in the section 7.4.

Meanwhile, data base management systems (DBMS) have been introduced as a data management structure in almost all GIS literatures. Simple feature model for the description of geometry was adopted in “ISO 13249-3, Information technology - Database languages – SQL Multimedia and Application Packages - Part 3: Spatial” was a remarkable contribution so as to integrate geometry and non-geometric attributes in a geographic feature. However, the selection of DBMS and its utilization are the issues of GI application implementation. Moreover today, the situation around data management system is evolving rapidly, for example, as progresses of NoSQL, XML database, data files following GeoJSON and semantic Web. Therefore, GIT in this thesis includes only the conceptual structure for geo-library and excludes the DBMS. Instead, data exchange with XML document will be introduced at the section 5.6. Cartographic products represented by hyper-text markup language (HTML) file with SVG tag and GeoJSON file will be introduced in the section 8.8.2.

2.4.5. Concept map of GIT

As a result of above discussions, GIT can be described in accordance with the deep structure and the surface structure, and the management structure. Moreover, the four-layer metamodel hierarchy is applied to describe these structures.

In the case of deep structure, geospatial objects are described in compliance with the application schema and so on, as models. Application schema as a model specifies the universe of discourse defined as an instance of the general feature model (GFM) as a metamodel. GFM is described in detail at the section 3.2. The meta-metamodel is a notation rules of UML. UML is used to describe not only a metamodel (e.g., GFM) but also a model (e.g., application schema).

In the case of surface structure, cartographic objects are described in compliance with the portrayal schema as a model. As an instance of General Portrayal Model (GPM), portrayal schema specifies the rules for graphic representation such as interactive maps and lists. GPM is a rule for the description of portrayal schema as a metamodel. It is described in detail at the section 4.2. The meta-metamodel is a notation rules of UML. UML is used to describe the metamodel (e.g., GPM) and the model (e.g., portrayal model).

In the case of management structure, metadata are described in compliance with a metadata schema as a model. GMM is a generic rule for the description for metadata schema in any application domain as an instance following the UML notation rule. The meta-metamodel is a notation rules of UML. UML is used to describe the metamodel (e.g., GMM) and the model (e.g., geospatial metadata model).

Consequently, the conceptual structure of GIT can be described in accordance with the four-layer metamodel hierarchy, the deep structure and the surface structure, and the management structure. Therefore, we can define a matrix of which rows are defined by the four-layer metamodel hierarchy and its columns are the management structure, the deep structure and the surface structure. In this thesis, the matrix as shown in Figure 2.4-2 is called *the concept map of GIT*. GI standards are basis to ensure the consistency between the elements of the concept map. The elements in this map relates with adjacent elements, excluding the relationship between general metadata model and general geospatial models. The reason is that the general metadata model is independent from application domains.

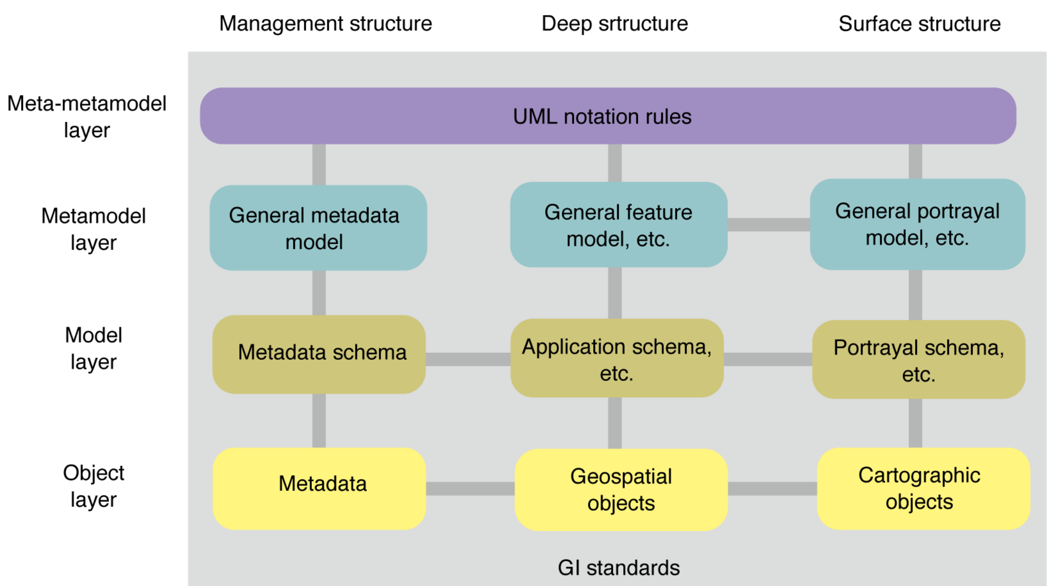


Figure 2.4-2. Concept map of GIT.

As already defined in 2.2, GIT refers to a set of information technologies for modeling, acquisition, management, analysis, exchange and representation of geospatial information. Modeling is a function to produce application models by abstraction of the real-world phenomena, and to produce portrayal models for designing of cartographic products. Acquisition is a function to transform real world phenomena to geospatial objects. Management is a function to accept the request and return the geospatial objects from the repository, and to store the geospatial objects with metadata in the repository. Analysis is a function to transform the geospatial objects to information as different geospatial objects. We do not discuss the mathematical or physical models for spatial analyses in this chapter, because most of them can be considered as operations defined in the feature types in the application schema. Exchange is a function to encode and decode between original geospatial/cartographic objects and standardized geospatial/cartographic objects. Finally, representation is a transformation from cartographic objects to cartographic product used for display of the products on the graphic media, such as paper, graphic display, VR head set, and so on. Usually, cartographic objects are reversible to geospatial objects, however cartographic products are not reversible to cartographic objects, as they are fixed to show on the display device.

As a consequence, the conceptual structure of GIT is defined as the concept map shown in Figure 2.4-2, and functions and objects in GIT are distributed in the cells of the concept map. Therefore, we will discuss the conceptual structure of GIT from the viewpoint of objects and from the viewpoint of functions. However, before these discussions, we will resolve the inconsistency between the deep structure and the surface structure that the author found during the development of gittok.

2.5. Inconsistency between deep structure and surface structure

2.5.1. Introduction

In this section, inconsistency between the deep structure and the surface structure is pointed out, and the way to resolve the inconsistency between two structures will be proposed. However, before the discussion, we introduce physical feature, virtual feature and proxy attribute as they will be useful means to resolve the inconsistency.

According to 4.1.11 in ISO 19101-1 (ISO/TC 211 2014a) , feature is defined as follows.

"abstraction of real world phenomena.

A feature can occur as a type or an instance. Feature type or feature instance should be used when only one is meant."

Feature is visible or invisible. Visible feature has geometric shape represented by coordinates on the earth. While invisible feature does not have a shape. In this thesis, 'physical feature' is a feature having geometric attributes in principle. It is an abstraction of tangible and visible phenomena in the real world. Meanwhile, we cannot see the international border line on the ocean surface, we cannot see the artificial structure yet to be constructed, and we cannot see the ancient city already destroyed. All of them are invisible in the real world. However, even they are invisible, if the consensus about that they are/were existing in the real world are established by parties involved in, they will be assumed as physical features.

In this thesis, 'virtual feature' is a feature that has no geometric attributes in principle. It is intangible and invisible in the real world. For example, we know the ministry of foreign affairs is an existence in the real world. But to think of the shape of it is nonsense. Such a feature is thought as virtual. On the other hand, sometimes we abstract visible entity as a virtual feature. For example, to represent a ranking list of the city brand in the world, we can ignore geometric attributes of the cities. Meanwhile, we may imagine that the feature type 'Company' may have a position of the head office as a geometric attribute. However, the head office should be defined as an independent feature type associating with Company type. Because, even if a point is displayed on the map, it is a position of the head office, not the company, and because a company is primarily a group of people such as the ministry of foreign affairs.

In this thesis, considering the flexibility of application, we introduce 'proxy attribute'. Proxy attribute is an attribute to switch the feature physical or virtual. We think a feature as a virtual feature as long as the proxy attribute is non-geometric.

2.5.2. Inconsistency

We can view GIT from the hypothesis "*Geospatial data are represented on maps.*" On the view point of the surface structure, the purpose of the technology on the surface structure has been considered as mapping. As well, we can view GIT from the hypothesis "*Geospatial data are independent from their representations*". This is a view point of the deep structure as it concerns how to abstract real world phenomena, not its representation (Ota 2017).

First hypothesis

Most of currently used GI systems are designed under the first hypothesis as 'geometry-centric system'. A feature must have geometric attribute that can be displayed on a map. For example, according to

Ormsby et. al. (Ormsby et al. 2010), *"Features have shape and size."* QGIS adopts the vector layer model (Graser 2013). The fundamental data model of ArcGIS is the geometry-centric layer-based model, although it has useful functions whereby it is possible to attach pictures, videos or files as 'geodatabase attachment' (Nasser 2014). The format specification of GeoJSON states that objects shall have a member of geometric data type (Butler et al. 2015). In fact, the geometry-centric system with multi-layers is still used by many GISs (Heywood, Cornelius, and Carver 2011). In those systems, geospatial data consist mostly of a set of physical features, and each feature has a one-to-one correspondence (bijection) with a symbol on a map.

Second hypothesis

The second hypothesis is valid because of at least two reasons. First, this hypothesis makes it easy to describe geospatial data for the purpose of geospatial analysis, because the data model allows relatively simpler structure as it is free from representation. For example, to predict the population density of the city X of 10 years later, the feature of the city should include geometric attribute (polygons) of the census districts to get the population density, but the result of the analysis used for publishing is a list of objects having a name of the city and one number representing the predicted population density. Second, geospatial data should be provided to the public for multi-purpose reuse, because, in general, the cost of acquiring original data can be recovered by selling enough number of copies. Therefore, geospatial data should be independent from its representation.

Example of virtual feature

Under the second hypothesis, geospatial dataset may contain both physical features and virtual features. For example, a university is a virtual feature having no geometric shape, because it is an organization, not a ground nor a school building(Figure 2.5-1) (Ota and Plews 2015). Even though, premises of the university have grounds and school buildings with geometric attributes, they are not a university. Moreover, there is a possibility that a university have no premises in the real world. In fact,

> *"In the late Middle Ages, as student populations grew and universities ceased to migrate, universities acquired buildings and movable property. For a long time in Paris and Bologna the administration had not needed to take care of university buildings, because there were none. Lectures were held in houses rented by the masters, examinations and meetings in churches and convents"*(Ruegg 2003).

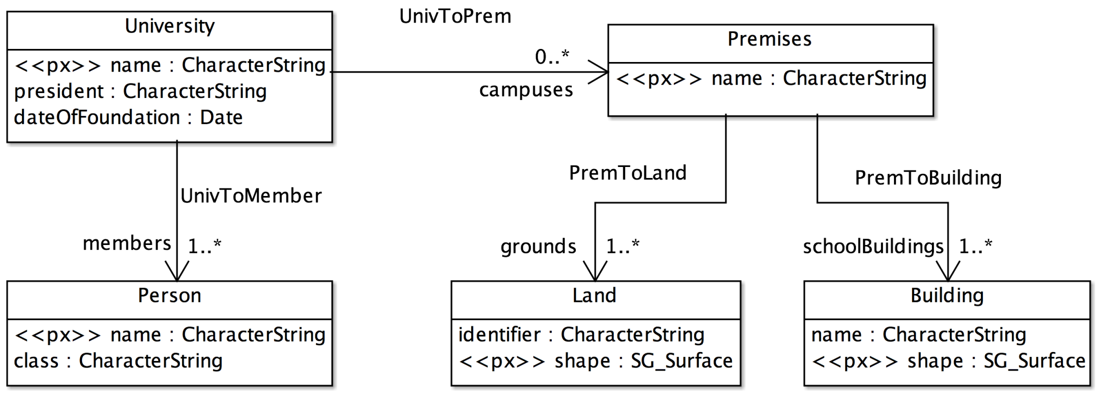


Figure 2.5-1. Class diagram to show university and its premises. University, Person and Premises are virtual in this case. The stereotype <<px>> means that this is a proxy attribute.

We also can imagine a university on the web without premises. As a consequence, there is an inconsistency between the deep structure and the surface structure, because a part of data in the deep structure cannot be transferred to the surface structure, as long as the surface structure requires physical features for graphic representation.

In addition, according to ISO 19101-1 (ISO/TC 211 2014a),

> *Non-geographic features are also of interest in ISO geographic information standards. Such features may be included in the application schema with no spatial characteristics.*

It means that ISO 19100 series of standards allow to define virtual feature types. However, this sentence seems to not think of representation of features, because the author could not find the specification about how to represent virtual features at least in ISO 19117:2012 – Portrayal (ISO/TC 211 2012a).

By the way, virtual feature does not conflict with the rule of GFM, as GFM does not make a demand that spatial attribute is a mandatory element of features (Figure 3.2-1).

2.5.3. Solution

The first hypothesis can be recognized as a special case of the second hypothesis, because it can be built in the second hypothesis in which the objects include more than one physical features and zero virtual feature. However, we are able to find a way to represent virtual features with non-geometric proxy attribute. A value of a proxy attribute may be discrete or contiguous, but it should be a unique non-geometric identifier of the feature. Therefore, a list of records that consist of proxy attributes and other attributes may be a solution of this problem. A list made from virtual features can be seen as a map of one-dimensional space as the data type of proxy attribute is one dimensional value. A list may be converted to other information styles such as a bar-chart, a linear graph, and a chronological table.

By the way, if physical features have a key attribute such as geographic name, they can be converted to a gazetteer as a combination of key attribute and geometric position. In this context, a gazetteer is an adjunct of a map, as its role is an index of physical features displayed on maps. Therefore, a gazetteer is different from a list. Figure 2.5-2 illustrates that physical features are represented by a map and a gazetteer, while virtual features are represented as a list. There is no line between physical feature and listing. If there is a demand to represent a list of physical features, it is possible to switch their proxy attributes from geometry to non-geometric attribute before listing.

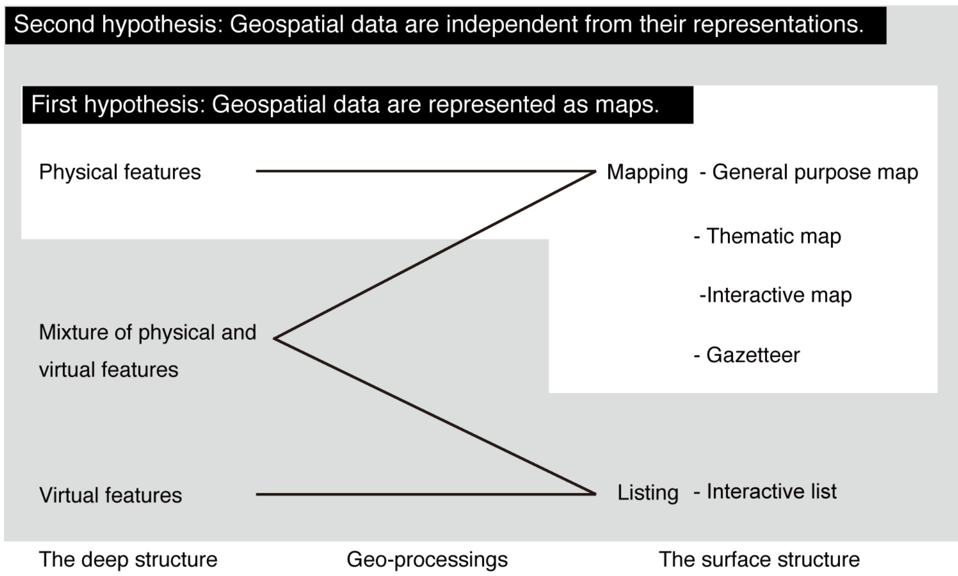


Figure 2.5-2. Two hypotheses regarding cartographic representations. White zone represents the extent of the first hypothesis. It represents a geo-processing to produce maps and gazetteer.

As a consequence, we can resolve the inconsistency between the deep structure and the surface structure by the proxy attribute and a list representing virtual features. A list will expand the paradigm of GIT. Accordingly, a metamodel for a list schema should be added in the general portrayal model

(GPM). The metamodel for a list schema as a part of GPM is called the general list model (GLM) in this thesis. GLM will be specified in 4.3.

2.6. Conceptual structures from the viewpoint of objects

In this section, we propose the conceptual structure of GIT from the viewpoint of objects. According to the concept map of GIT shown in Figure 2.4-2, the management structure consists of layers for metadata, metadata model, and general metadata model. The deep structure consists of geospatial objects, application models, and general geospatial models. The surface structure consists of layers of cartographic objects, portrayal models, and general portrayal models.

We briefly discuss the deep structure, the surface structure and the management structure from the viewpoint of objects in the following sub-sections.

Deep structure for geospatial objects

The deep structure consists of four layers in accordance with the four-layer metamodel hierarchy. According to ISO 19109, a feature is defined as abstraction of real world phenomena. A feature may occur as a type or an instance. A feature instance is an object as an element of geospatial objects. While, a feature type is defined as an element of application schema. Therefore, geospatial objects are located in the object layer, and application schema is located in the model layer. Application schema is modeled in accordance with GFM described in UML. Therefore, GFM is located in the metamodel layer and UML is located in the meta-metamodel layer. Figure 2.6-1 illustrates the detail image of the concept map of GIT from the view point of objects.

By the way, one of the main purposes of GI standards is to provide the protocol for data exchange. However, we should think GFM and application schema are not only for data exchange but also for designing of GI applications. Moreover, GFM is considered to be one of the principles of GI science. Because, the goal of GFM is that it will be able to schematize any geospatial phenomena in the world.

As long as the role of the deep structure is to describe geospatial objects, the object layer consists of geospatial objects brought from the real world by surveying, or existing geospatial objects and documents by transformation or digitizing from analogue resources. Geospatial objects may be encoded to or decoded from the system independent format for exchanging between other applications.

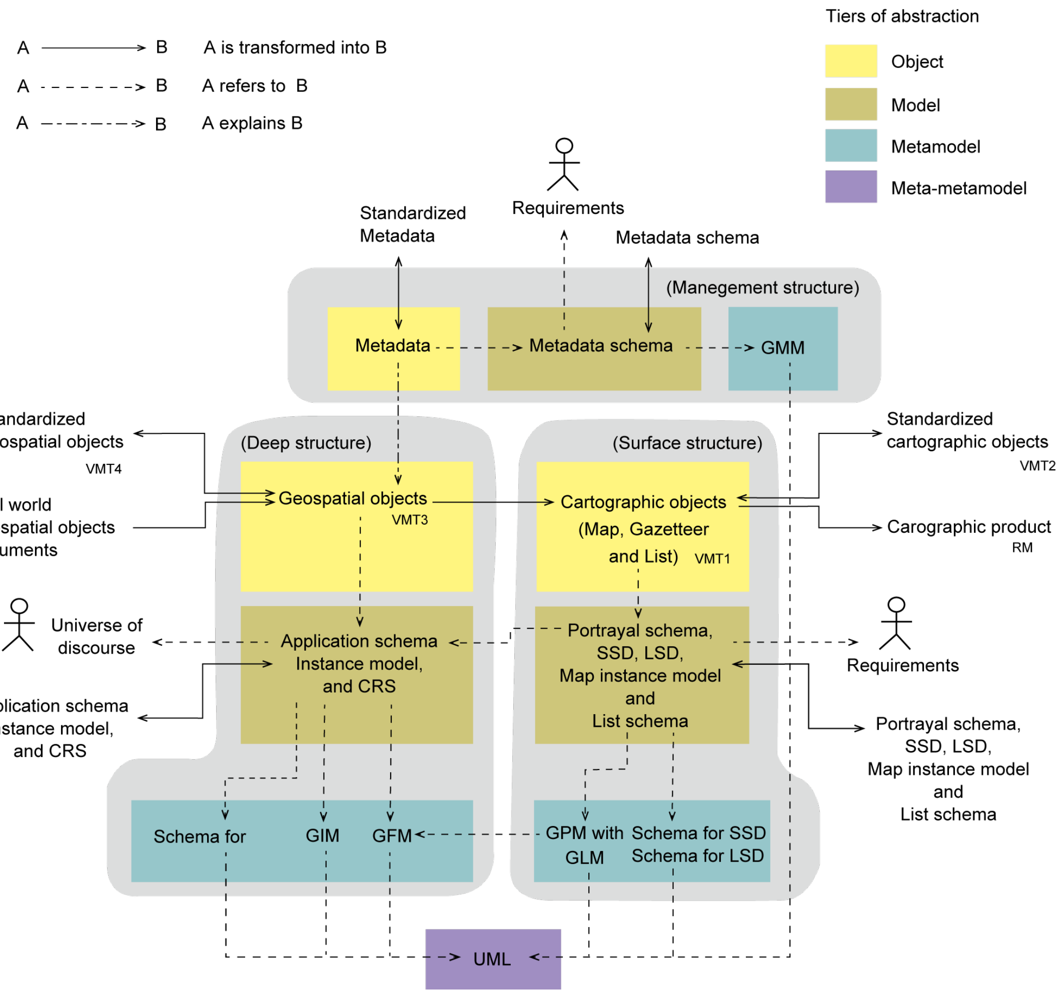


Figure 2.6-1. Detail image of concept map of GIT from the view point of objects.

Encoding and decoding become possible by transferring of application schema with parameters of coordinate reference system (CRS) for the description of coordinates in the geometric attributes. In addition, geospatial objects are translated to the surface structure and transformed to cartographic objects for information representation. By the way, we already discussed and resolved the inconsistency between geospatial objects and cartographic objects in the section 2.5. Therefore, it is possible to transform geospatial object (physical and/or virtual) to cartographic objects in the surface structure.

The model layer consists of schemata for describing the geospatial objects. The conceptual structure of geospatial objects is defined as an application schema. An application schema is created in accordance with GFM. To describe geospatial objects with the system independent style, the instance model as an instance of GIM is required. In practice, application schema is used to define the dataset stored in the repository. While, the instance model is used to define the structure of the dataset for exchange.

By the way, the parameters of CRS are required to describe coordinates as elements of geometric attributes. The structure of CRS is defined as the schema for CRS and it is located in the metamodel layer. Models in the model layer should be able to exchange between other applications in the system independent formats to ease the transferring between different applications.

GFM, GIM, and the schema for CRS are rules for creation of domain specific models in accordance with the universe of discourse. GFM will be discussed in detail at the section 3.2. GIM will be discussed in detail at the section 3.3. And the schema for CRS will be discussed in detail at the section 3.4.

Surface structure for geospatial objects

The surface structure consists of four layers in accordance with the four-layer metamodel hierarchy for the representation of cartographic objects as maps and lists.

Object layer consists of cartographic objects for maps, gazetteers, and lists. Map is a graphic representation of the real world and the result of analysis. Gazetteer is an index of features drawn on the map. And a list consists of connected items consecutively. The geospatial objects stored in the deep structure are transferred to the surface structure, and it is transformed to the primitive cartographic objects, then it is edited to cartographic produces with map marginalia in compliance with the portrayal schema and the map instance model. And a list is constructed in accordance with the list schema. In this thesis, map, gazetteer and list are called cartographic products. Bay the way, a gazetteer can be constructed by selecting place name attribute type and geometric attribute type. Gazetteer schema is not required as long as it is considered as independent from applications.

Model layer consists of a portrayal schema with symbol style dictionary (SSD) and a label style dictionary (LSD), a map instance model, and a list schema. Portrayal schema specifies the rules for symbols and labels. Moreover, if the representation will be done as an interactive map, multimedia attributes will be selected for each feature type. The symbol design and the label design are specified as SSD and LSD. The portrayal schema, SSD and LSD are designed in order to fulfil the requirements for mapping given by the representation designer. Next, to exchange cartographic product between other systems, cartographic product is transferred to the data with system independent format in accordance with the map instance model. These products are the main outcomes of geospatial information processing. In addition, list schema specifies the rules for list representation.

Metamodel layer consists of GPM, GLM, schema for SSD, and schema for LSD. GPM, schema for SSD, and schema for LSD will be discussed in the section 4.2, and GLM will be introduced in the section 4.3.

Management structure for geospatial objects

The object layer in the management structure is a geo-library including a set of metadata. The primary mechanism of geo-library is as follows. The user asks appropriate metadata by giving keywords to the geo-library. The user decides to access geospatial objects or not by evaluating metadata obtained by the geo-library. If the answer was ‘yes’, the geo-library send the objects to the user.

The model layer of the management structure includes metadata schema. Metadata schema for geospatial domain should have the geographic extent of geospatial objects at least. It will be introduced in the section 7.3.

The metamodel layer of the management structure includes a general metadata model (GMM). GMM defines the structure of metadata schema designed for the application domain. It will be introduced in the section 7.2.

2.7. Conceptual structures from the viewpoint of functions

GIT consists of six knowledge areas as already described in the section 2.4.5. They are modeling, acquisition, management, exchange, and representation. These knowledge areas are included in the deep structure, the surface structure or the management structure. Generally, representation is included in the surface structure, and acquisition is included in the deep structure. However, modeling and exchange are included in both. Modeling of application schema is included in the deep structure, and modeling of portrayal schema is in the surface structure. Geospatial objects are exchanged from/to the deep structure, and cartographic objects/products are exchanged from/to the surface structure. In addition, each knowledge area is structured by the four-layer metamodel hierarchy.

Meanwhile, knowledge areas can be seen as a set of functions, which is a process providing information from resources. Figure 2.7-1 illustrates the configuration of a function in GIT. Function name is represented as a verb in this thesis, because function is a command to do something. For example,

- Identify information sources such as the real world, geospatial object, documents, and so on to the universe of discourse according to the user requirements.
- Schematize the universe of discourse to an application schema in accordance with GFM.

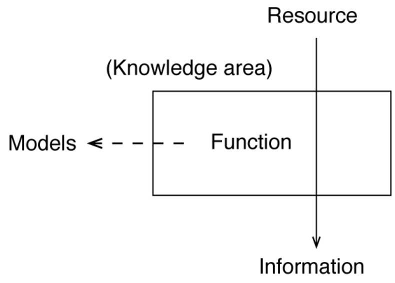


Figure 2.7-1. Metamodel for functions in GIT

As can be seen from this explanation, function is not necessarily a software module. Actually function 'identify' is executed by humans.

The structures for functions are described in Figure 2.7-2 and Figure 2.7-3. Figure 2.7-2 shows the functions in the deep structure. Figure 2.7-3 shows the functions in the surface structure.

Functions associating with the deep structure

- The universe of discourse is identified in the modeling from information sources such as the real world, geospatial objects, documents, and so on by referring to the user requirements. Geospatial objects, documents, images and so on are materials for the consideration of the universe of discourse. Usually the universe of discourse is often written in the natural language, because decision makers are sometimes not an expert of GIT.
- The application schema is schematized from the universe of discourse in the modeling by referring to GFM. Application schema is written in UML as a formalized description of the universe of discourse.
- Data and operation are extracted from geospatial objects by referring to the application schema in the analysis. Data for analysis are arguments for the operation, and the return values are also become values of the derived attributes defined in the feature. They are extracted in accordance

with the definition of the operation that is chosen by the user.

- Geospatial objects following the application schema are obtained by the execution of the operation in the analysis. At least two methods are conceivable for the execution of the operation. One is automatic execution. Every time the value of argument is changed, the operation is automatically executed and the latest value will always be in the derived attribute. Another method is a method of executing an operation according to a user instruction. If there are multiple arguments, this method will be used as it is difficult to execute the operation unless the user issues a command.

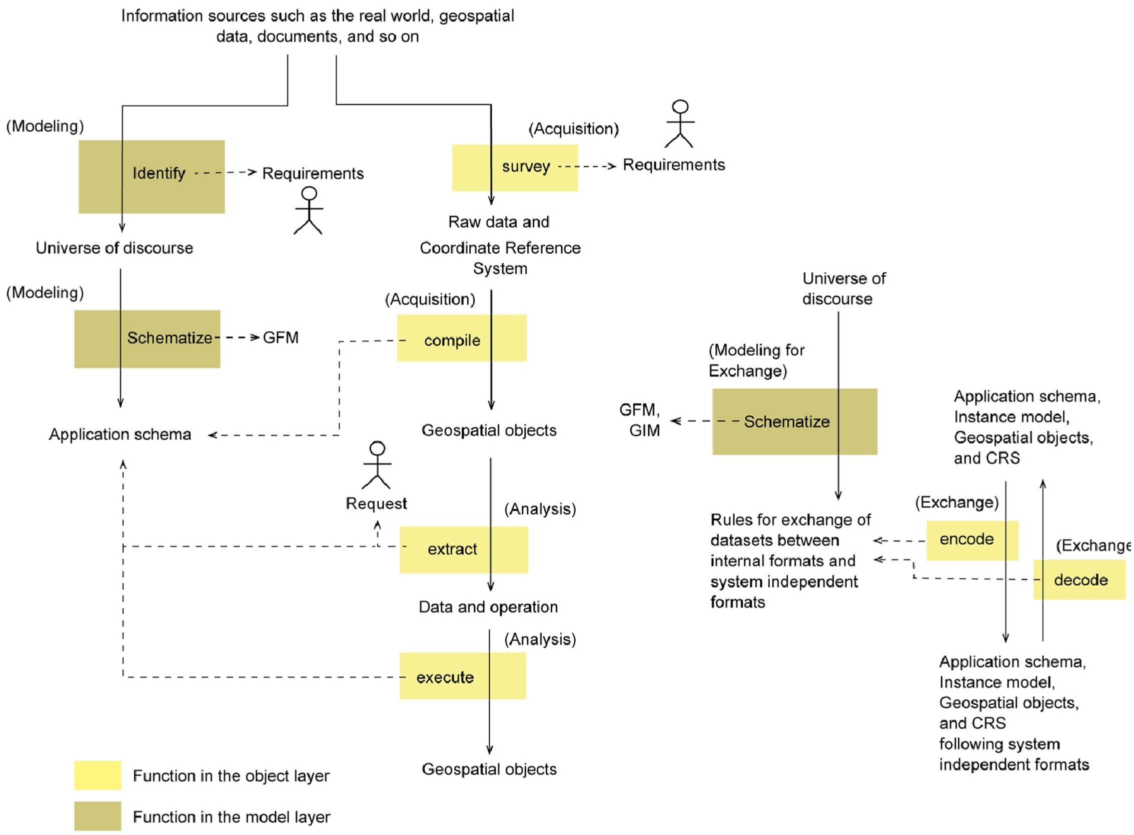


Figure 2.7-2. Functions in deep structure

Therefore, in this thesis, we follow the latter method.

- The universe of discourse is schematized in accordance with instance models for exchange of datasets between internal formats and system independent formats by referring to GIM and GFM. The schematizations of application schema and instance model are required as they are application dependent. Meanwhile, the metamodel for CRS is not required. Because the model for CRS is already the metamodel and CRS is a model for coordinate description.
- Application schema, instance model, geospatial objects and CRS are encoded as datasets following system independent formats as XML documents by referring to the rules for exchange. Not only

geospatial objects with CRS, but also application schema and instance model may be exchanged, because it is impossible to understand without these schemata.

- Application schema, instance model, geospatial objects and CRS as XML documents are decoded as datasets following internal formats by referring to the rules for exchange.

Functions associating with the surface structure

Figure 2.7-3 shows the functions for representation associating with the surface structure. Representation consists of map design, mapping, gazetteer compilation, list design, listing and exchange of information on representation. These sub domains are defined to fulfill the requirements from the concept map of GIT. Functions illustrated in Figure 2.7-3 are described as follows.

- The universe of discourse for map design is identified from information sources such as the real world, geospatial objects, documents, and so on by referring to the user requirements for map representation.
- Portrayal schema, SSD and LSD are schematized from the universe of discourse in the knowledge area of map design by referring to GPM, and schemata for SSD/LSD. SSD and LSD are dictionaries of symbol styles and label styles.
- Map is edited by referring to the portrayal schema with SSD/LSD, and it is distributed in compliance with the map instance model. This process is called map editing or mapping.
- Gazetteer is compiled by referring to the gazetteer instance model and it is distributed with a map as an adjunct information of the map.
- The universe of discourse for list design is identified from information resources such as the real world, geospatial objects, documents, and so on by referring to the user requirements for list representation.
- List schema is schematized from the universe of discourse in the list design by referring to GLM as a part of GPM.
- A list is edited by referring to the list schema and it is distributed in compliance with the list instance model. This process is called listing.
- Portrayal schema, SSD, LSD, map, gazetteer, list schema, and list are encoded as datasets following system independent formats by referring to the rules for exchange.
- Portrayal schema, SSD, LSD, Map, gazetteer, list schema, and list are decoded from datasets following system independent formats by referring to the rules for exchange.

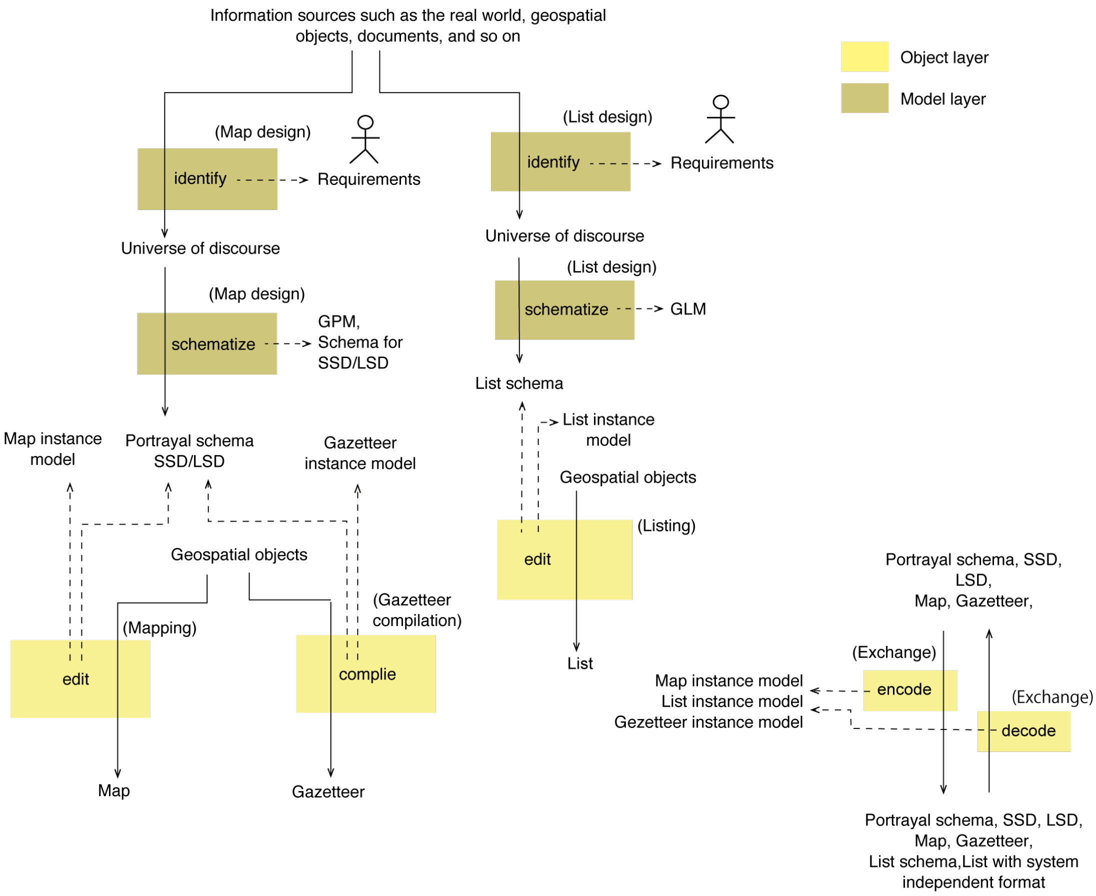


Figure 2.7-3. Functions in surface structure

Functions associating with the management structure

- The universe of discourse for metadata modeling is identified from information sources such as the real world, geospatial objects, documents, and so on by referring to the user requirements for metadata for the target domain.
- Metadata schema is schematized from the universe of discourse in the metadata modeling by referring to GMM.
- Metadata is edited in accordance with the metadata schema by using the geospatial objects.
- Metadata is stored in the geo-library to reply the request from the user.
- The user retrieves appropriate metadata from the geo-library for his/her purpose by putting the conditions.
- If the metadata is appropriate for the user's purpose, the user extracts the geospatial objects.

Functions in the management system will be discussed in detail at 7.4.

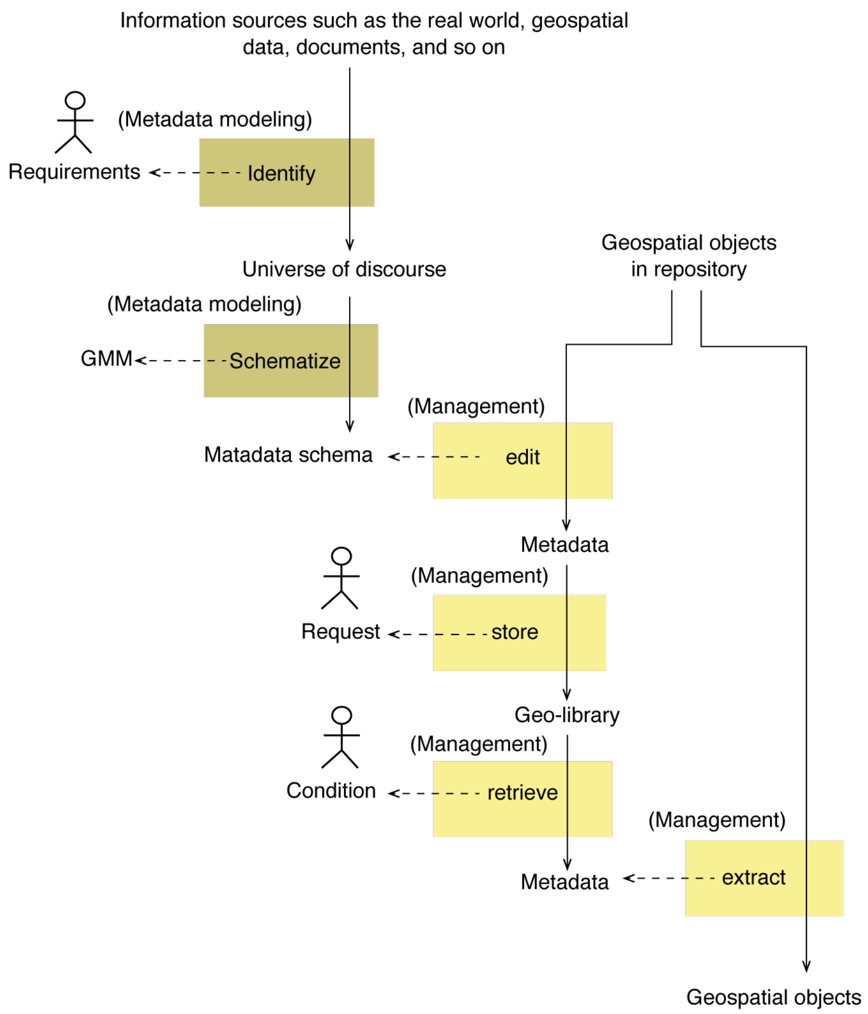


Figure 2.7-4 Functions in management structure

3. Conceptual structures for geospatial objects

3.1. Introduction

Geospatial objects are defined by an application schema, and the application schema is defined in accordance with GFM. Geospatial objects are transferred by following an instance model, and the instance model is defined by GIM. Coordinates in geometric attributes kept by feature instances are defined by a CRS, and the CRS is defined by the schema for CRS. Therefore GFM, GIM and the schema for CRS are metamodels for geospatial objects. In this chapter, we discuss these metamodels in detail.

According to ISO 19109 – Rules for application schema (ISO/TC 211 2005, 2015b), geospatial objects are described as instances of feature types or association types defined in the application schema. Application schema should commonly be understood by different systems, otherwise it is difficult to understand the meaning of elements in geospatial objects. Therefore, it is described in accordance with GFM.

Geospatial objects can be converted to the standardized geospatial objects with the system independent format. Typical example of such format is described in ISO 19118 – Encoding (ISO/TC 211 2011b). In principle, the rules are defined in order to fit into the demands from the universe of discourse, and formalized by the standardized metamodel. It is called general instance model (GIM) in this thesis.

Geospatial objects may have the geometric attributes that contain geographic coordinates in accordance with CRS. The CRS is defined in accordance with the schema for CRS that can be understood by the different systems. A schema of CRS is provided by ISO 19111 – Spatial referencing by coordinates (ISO/TC 211 2007a).

In this chapter, we discuss GFM, GIM and the schema for CRS used in gittok as follows.

i. Gittok GFM as a profile of the GFM provided by ISO 19109 is discussed so as to understand in the shorter period of time for beginners, and so as to be more practical to define operations.
ii. Gittok GIM as rules for instance model is defined to avoid misunderstanding of instance models provided by different application domains by referring to ISO 19118.
iii. An instance model for the description of the geospatial data acquired by map digitizing is defined

in compliance with gittok GIM, because gittok enables map digitizing as an example of data acquisition.

iv. A schema for CRS in gittok as a profile of the schema provided by ISO 19111 is discussed so as to understand in the shorter period of time for beginners.

3.2. General feature model (GFM)

3.2.1. Introduction

GFM specified in gittok is rules for formal definitions of feature types, association types and their properties such as attributes and operations by referring to GFM defined in ISO 19109. Geospatial objects may have various types of attribute, for example, character string, number, geometry, place including address, URL, and locations of video and audio. In ISO GFM, attribute data types are classified into spatial, temporal, thematic, and locational. We discuss and provide the definition of gittok GFM, spatial schema, schema for thematic attributes, and schema for place identifiers in this section. Location is rewarded by place. Place is defined as an identifiable part of any space. We will define not only the location in the real world, but also define the resource locater in the Internet. Consideration about temporal schema and its implementation in gittok are the future issues. The reason will be mentioned at the section 3.2.7.

3.2.2. GFM

GFM designed for gittok should be a simplified and practical profile of ISO 19109 for beginners to make possible to understand what GFM is. We referred to ISO 19109:2005 (ISO/TC 211 2005) to design gittok GFM, because we started to design gittok GFM before ISO 19109:2015 (ISO/TC 211 2015b) was published. Nonetheless, gittok GFM is a profile of ISO GFM as it follows the framework of ISO 19109:2005. Such profiles have been made also in other activities, for example, the International Hydrographic Organization (IHO 2017), the Defence Geospatial Information Working Group (DGIWG 2013) and GSI (http://www.gsi.go.jp/GIS/jpgis-jpgidx.html accessed 2018-07-25) have provided the GFM profiles for their standards.

Design policy of gittok GFM

Gittok GFM should be designed so as to be easier to understand for beginners, and more practical for implementation of GI applications. The differences between ISO GFM (Figure 3.2-1) and gittok GFM (Figure 3.2-2) are described below. Frequently, a type is named using a compound word that is a connection of words which first letter is capital, such as 'ApplicationSchemaType' in Figure 3.2-2. In the case of Figure 3.2-1, type name have the prefix 'GF_'. It indicates this is a member of GFM.

i. GF_InheritanceRelation to represent inheritance of a feature type is eliminated. Instead, the self-association between parent and children feature types is defined. Because the structure becomes simpler, and the inheritance is still possible by this association.
ii. Inheritance relationship between GF_FeatureType and GF_AssociationType is eliminated to make the structure of AssociationType simpler.
iii. GF_PropertyType is eliminated. Instead, OperationType and AttributeType directly associates with FeatureType and AssociationType, to make the structure of gittok GFM simpler. Even without GF_PropertyType, feature and association may have association between attributes and operations.

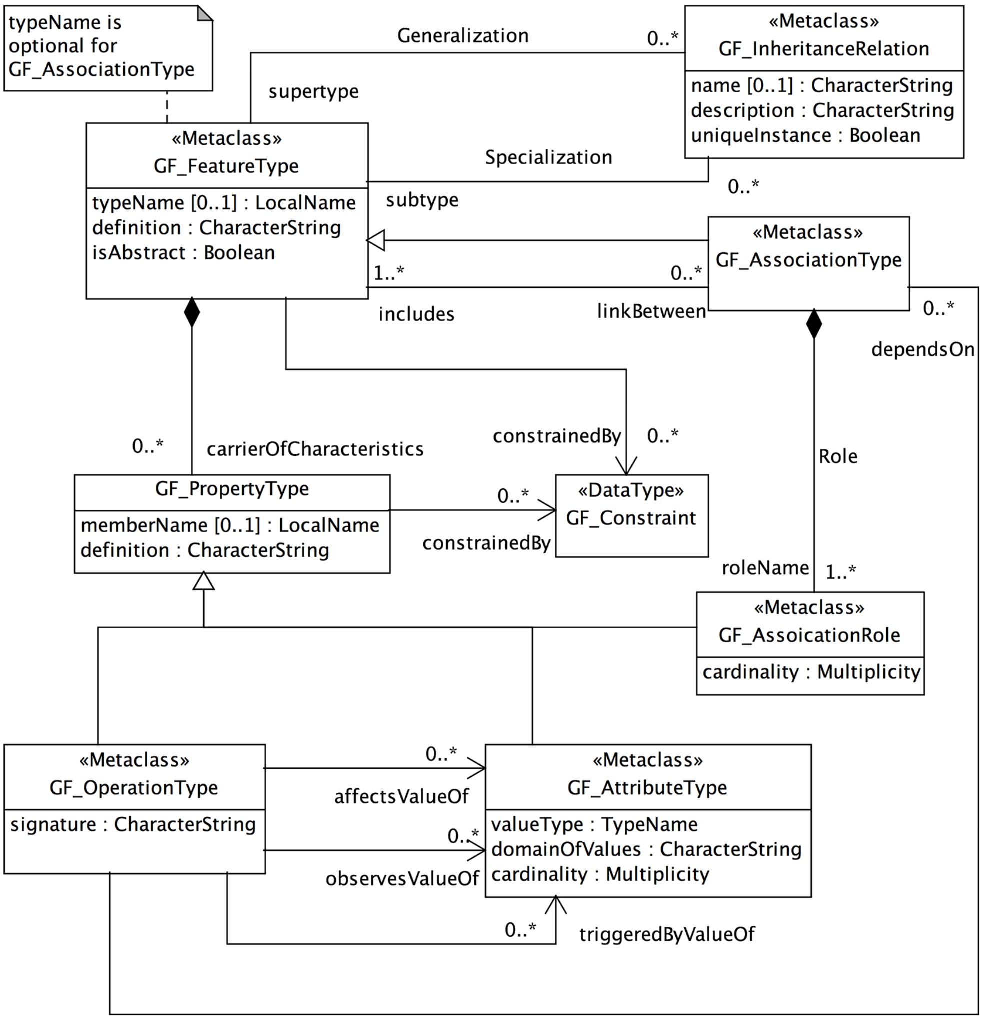


Figure 3.2-1. Class diagram of GFM as composition of Figure 5 and 8 in ISO 19109:2005

By the way, carrierOfCharacteristics is optional as its multiplicity is [0..*]. It means that GFM allows a feature type without attributes and/or operations. This rule supports the idea of virtual feature, as GFM allows feature type without spatial attribute. In fact, there is no restriction to declare the spatial attribute is mandatory in ISO 19109:2005.

iv. GF_Constraint to represent constrains of attributes, operations and feature type is eliminated to make the structure of GFM simpler. The constraint is important class to evaluate the quality of objects. However, the quality requirements can be specified in the data product specification separately.

v. GF_AssociationRole to represent roles of feature type associated with other feature type is eliminated to make the structure of GFM simpler. Instead, association role name become an attribute of AssociationType. In gittok GFM, association type is defined so as to link feature types as one-way connection.

vi. ArgAttPair to describe arguments and return value types for operation is added. And direct associations between operation type and attribute type are eliminated. Because, signature defined in GF_OperationType is just the character string and no detail description about the syntax of the operation. Instead, ArgAttPair enables the definition of operation type implementable. This class also enables the associations between attributes defined in the application schema and prepared operations already embedded in gittok.

vii. ApplicationSchemaType to describe a root element of application schema instance is added. Because, as far as application schema is an instance of gittok GFM, ApplicationSchemaType as a metamodel of the application schema is required.

Gittok GFM

In Figure 3.2-2, an application schema (ApplicationSchema) consists of feature types (FeatureType) and association types (AssociationType).

Feature association is defined as a relationship that links instances of one feature type with instances of the different or the same feature type. Therefore, the main role of feature association is to represent connection between features. In gittok GFM, AssociationType is defined by from-feature type and to-feature type, and a name, attributes, and operations. “From” and “to” means an associating direction.

Meanwhile, let’s imagine the distance between two objects A and B. Is distance an attribute of A or B? Basically, it is a characteristic of relationship between two objects. An operation to calculate the distance of two objects should be an operation in the association type. Therefore, AssociationType may have attributeTypes and operationTypes. As a matter of fact, an association type is a subtype of a feature type according to the ISO GFM, because it is also an abstraction of the real-world phenomena. However, inheritance and association between other associations are not allowed in gittok to maintain the structure of application schema simple. This is the reason that an association type is not a sub-type of a feature type in gittok GFM.

OperationType is a member of operationTypes and it defines attributes used as domains (arguments) and a range (return value) for the operation. The combination of the predefined argument of the operation and the attribute of the feature type are described in the pair of arguments and attribute (ArgAttPair) in gittok GFM.

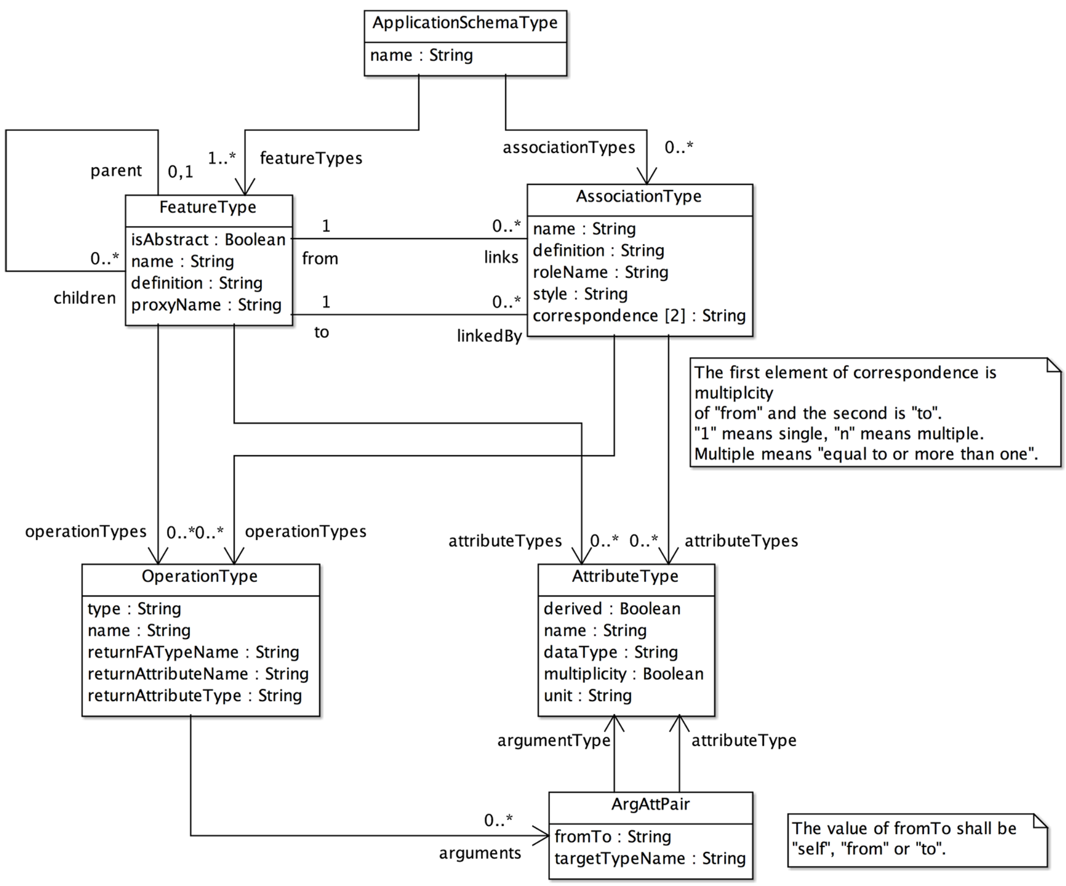


Figure 3.2-2. Gittok GFM as a profile of ISO GFM.

AttributeType is a type of members in attributeTypes. It consists of an attribute name and a data type. Data types are classified into spatial, thematic and place those are introduced in the following sections. There is no restriction to declare the spatial attribute is mandatory as same as Figure 3.2-1.

Restriction on definition of virtual feature

As already discussed the difference between physical and virtual features, a non-geometric attribute should be selected as the proxy for virtual features. Gittok GFM allows that FeatureType has an attribute ‘proxyName’. Currently, data type of a proxy attribute shall be a character string in gittok.

3.2.3. Spatial schema

Spatial schema is provided for describing the spatial characteristics of geographic features. Generally, spatial characteristics are described by geometric and topological objects. Spatial schema defined in gittok consists of two-dimensional geometries (point, curve, surface and their complex) with topology (Figure 3.2-3) , which is simplified the schema defined in ISO 19107 – Spatial schema (ISO/TC 211 2003a). Point (SG_Point) is a zero-dimensional primitive with a two-dimensional coordinate. Its co-boundaries are curves. Curve (SG_Curve) used in gittok is a C^0 class curve. It means that a curve is a one-dimensional primitive with an array of coordinates connecting from start to end point as boundaries, and it associates with left and right surfaces as co-boundaries. Surface (SG_Surface) is a two-dimensional primitive of which boundary has an exterior and more than zero interior rings. Ring (SG_Ring) is a sequence of orientable curves. Orientable curve (SG_OrientableCurve) is a curve with orientation. Curve has a direction as its nature, but a curve of opposite direction can be defined as an orientable curve. It is useful for the construction of a ring. Geometric complex (SG_Complex) is a set of primitives sharing only boundaries of adjacent primitives.

Meanwhile, ISO 19107 – Spatial schema specifies geometry independent from topology, but the gittok spatial schema defines geometry with topology between primitives. The reason is that geometry has boundary and co-boundary relationships with one dimensional lower and upper geometries by its nature. However, point has no boundary of minus one dimension and surface has no co-boundary of three dimension, because the extent of dimension in gittok is from zero to two. Off course, the minus one dimensional space is rather difficult to imagine. Therefore, we should think, a point is open as it has no boundary of minus one dominion. In other words, a point is open topologically, as it does not have topological boundary. Meanwhile, geometrically, point has a boundary. Let represent (a, b) as an open interval of real number, where 'a' is a lower limit, 'b' is an upper limit, and 'x' is located inside of this interval fulfill the condition (a < x < b). While the condition (a = b = x) is impossible, because the condition becomes (x < x < x). However, when we think a closed interval [a, b], (x ≦ x ≦ x) is possible. Therefore, {x} is closed in the one-dimensional geometric space. In the two-dimensional geometric space, (y ≦ y ≦ y) is also satisfied. It means a point (x, y) in the two-dimensional space is closed geometrically. As a consequence, a point has no topological boundary, but a point itself is also its geometric boundary.

In Figure 3.2-3, a point has associations between 'getIn' curves and 'goOut' curves. They are co-boundaries. By the way, a geometric complex is closed geometrically, as every primitive in the complex associates with its boundary primitives.
SG_Complex is defined in the spatial schema by referring to ISO 19107 (ISO/TC 211 2003a).

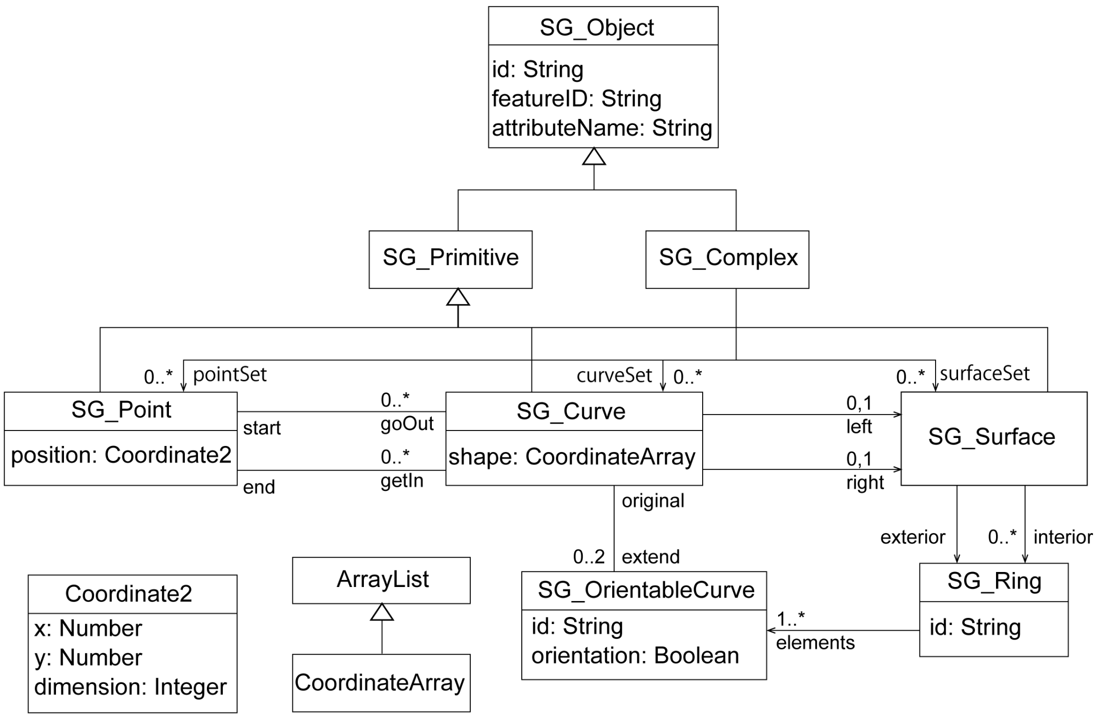


Figure 3.2-3. Spatial schema used for the introductory course of GIT.

However, the GI applications having a capability to describe geometry complex is not so common. At least, geometry complex is not usable in some GISs currently used. The reason is maybe that a large network (e.g., a water tube network) will be monopolized by one client. During data updating, other client cannot use this network data for example for risk assessment of water pipe network. However, theoretically, it is useful to provide geometry complex, if the return value of spatial analysis operation takes a graph structure. Therefore, spatial schema in gittok has SG_Complex so that it can be used as the return value of, for example, for shortest path finding.

CoordinateArray is an array list of coordinates. Array list supports dynamic arrays that can grow as it is required. Generally, we use an array list to represent a curve as an ordered set of coordinates. In gittok, a coordinate is restricted to two-dimension. However, it will be able to extend to three-dimensional, as long as the geometries can be mapped onto a plane without overlapping. The reason to restrict the coordinates in two-dimensional space is that the teaching period of time is limited. By the way, coordinates are described in accordance with the coordinate reference system (CRS). Schema for CRS will be discussed in 3.4.

In the history of GI systems, there has been a tradition of describing geospatial data with discrete object (vector data) and continuous field (raster data). However, gittok does not provide a special data type that describes continuously distributing phenomena such as terrain and temperature distribution. This is because continuously distributing phenomena are objects having a certain geographic extent. For example, an object obtained by airborne sensor may be described as a collection of points having two-dimensional coordinates (x, y), height and spectroscopic characteristics as attributes. And, in addition, this object may have an operation including an interpolation method which estimate the characteristic of an arbitrary point in the extent of the object. Therefore, we do not define special data type for continuously distributing phenomena in gittok. By the way, we should remind that gittok cannot describe a surface that is homeomorphic with a two-dimensional manifold as a surface shall be geometrically two-dimensional to be able to map onto a plane.

3.2.4. Thematic attributes

Thematic attributes are, for example, integer, real, Boolean, character string, date-time and memo in gittok GFM. Memo is an original data type for gittok. It is a composite data type, which consists of a title (or section name) and sequence of more than one sentence. This is useful to explain a feature as an article with sections. Other data types are primitive types (ISO/TC 211 2015a). It means they are kept in an object as attribute values.

Attribute with an integer or a real value may have a unit of measure. Date-time type is implemented in programming language and operating system. As gittok is implemented as an Adobe Air application coded by ActionScript V3.0, an instance of the Date class represents a particular point in time for which the properties such as month, day, hours, and seconds can be queried or modified. It is relative to Coordinated Universal Time (UTC) or relative to the local time, which is determined by the local time zone setting on the operating system. A function getTimezoneOffset():Number returns the time difference (in minutes) between UTC and local time.

Figure 3.2-4 shows the structure of primitive data types. GDate means 'Gregorian date and time' of which value type is Date class in ActionScript. Currently, GDate is used only as the publishing date of metadata.

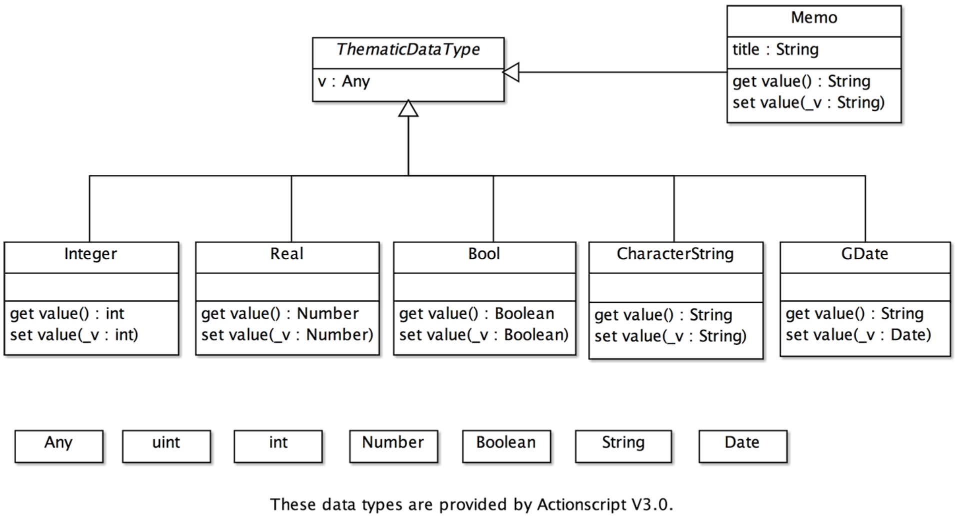


Figure 3.2-4. Thematic data types shown as UML class diagram adopted in gittok GFM.

3.2.5. Place identifiers

Place is defined in the Oxford English Dictionary as follows.

"a particular position, point, or area in space;

a location: I can't be in two places at once |

the monastery was a peaceful place |

that street was no place for a lady |

figurative: he would always have a special place in her heart."

Meanwhile, ISO 19112:2003 – Spatial referencing by geographic identifiers (ISO/TC 211 2012a) defines "location" as an identifiable geographic place; meaning location is a part of the real world. However, we usually use not only a street address or zip code but also uniform resource locators (URLs) of an image, movie, sound and home pages on the web as attributes of a feature. In fact, place is defined as *"identifiable part of any space"* in ISO 19155:2012 – Place identifier (PI) (ISO/TC 211

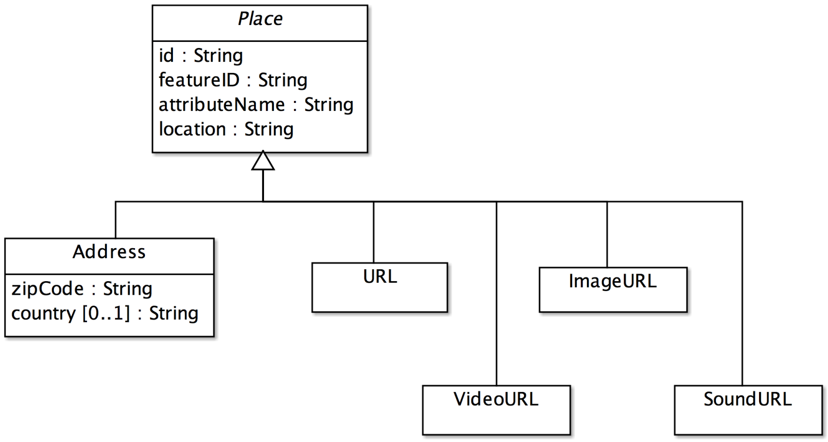


Figure 3.2-5. Place data types in gittok described by UML class diagram.

2012b). Therefore, we use the term place not location in this thesis. Place is not only a location in the real world, but also a location in the virtual world, for example, '*her heart*'. Place in Figure 3.2-5 has four attributes. Id is a unique identifier put by application. FeatureID is an id of the feature instance that has this place attribute. Attribute name is a name of the place attribute type held by the feature. Location as an attribute of Place is a uniform resource locator (URL) of the attribute, such as "www.attStock.net/images/photo001.jpg". However, if the place is address, we use Address. Therefore, location inherited from Place will be a local address without zip-code and country name, such as "2 Rokuban-cho, Chiyoda-ku,, Tokyo".

3.2.6. Application schema illustration

There are two methods to represent application schema. One is a description as a UML class diagram in accordance with the rules described in GFM. The other is a description as an object diagram of GFM class diagram. Both of them are used in this thesis. For example, Figure 3.2-6 (a) is a class diagram illustrated in compliance with GFM, while Figure 3.2-6 (b) is an object diagram in accordance with the UML notation rule as an instance of GFM. Application schema in gittok is described as an instance of GFM. Therefore, its graphic representation should be an object diagram. However, we also describe the application schema as a class diagram to ease to understanding.

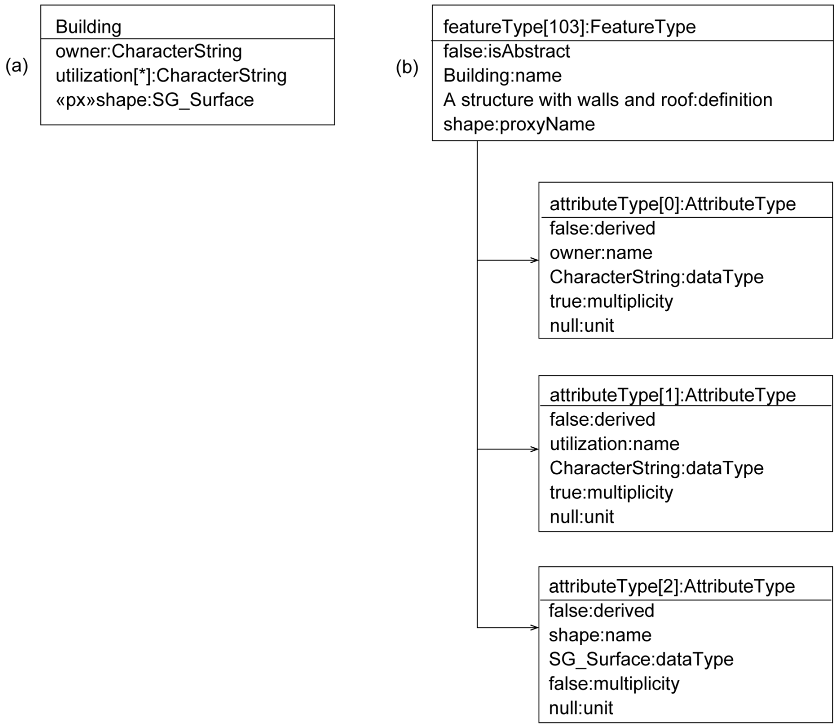


Figure 3.2-6 Illustrations of application schema as (a) class diagram and (b) object diagram of GFM

3.2.7. Temporal schema

Temporal attribute is a one-dimensional geometry (instant and period) with topological relationships. However, at this moment, it is not implemented in gittok. Because, temporal attributes are not only for feature and association, but also, they may be attributes of elements in application schema. Elements of application schema such as attributes in a feature type may be redefined when the universe of discourse changes. Therefore, we should consider how to integrate temporal characteristics such as schema extension or restriction in the temporal GFM. And we should consider the impact to feature instances caused by schema growing. Therefore, consideration about temporal schema and its implementation in gittok are the future issues.

3.3. General instance model (GIM) and instance model

Encoding is a process to convert from geospatial objects stored in the server to a system-independent data suitable for transferring to the client. The inverse process is called decoding. Encoding and decoding are performed by following the instance model. To make an instance model understandable for people and systems in different application domains, we should standardize a metamodel for instance models. Geography Markup Language (GML), Keyhole Markup Language (KML), shape, GeoJSON, TopoJSON, and the generic instance model defined in the ISO 19118. – Encoding (ISO/TC 211 2011b) are all GIM.

GML is a metamodel for instance models called GML application schema. It enables to describe geospatial objects as an XML document. GML is widely used for geospatial data exchange. Especially, OGC encourages to construct GML application schemata for data exchange in various application domains. Nevertheless, one of our aims is to introduce the learning assistance software tool on GIT and GI standards especially provided by ISO/TC 211. ISO GML is in compatible with GFM, however unfortunately, its structure seems to be too complex to learn in the introductory course of GIT.

Shape, provided by ESRI is probably the most commonly used and supported data format for geographic information (Portele 2011). GeoJSON (Butler et al. 2015) provided by the Internet Engineering Task Force is a geospatial data interchange format based on JavaScript Object Notation (JSON). And TopoJSON (Bostock and Metcalf, https://github.com/topojson/topojson-specification accessed 2018-11-03) is a topological geospatial data interchange format based on GeoJSON. They are also widespread across the world today. However, these formats are not designed in compliance with ISO GFM. Because, they define geographic data as a combination of geometry and other attributes. If these formats were in compatible with GFM, they would be able to adopt in gittok.

ISO 19118.:2011 – Encoding (ISO/TC 211 2011b) is the standard for the creation of instance models in accordance with the general instance model (GIM). GIM described in 'ISO 19118. Chapter 8. - Generic instance model' is defined in writing as follows.

> *"The basic unit of information in a dataset is the object. An object shall be an instance of a single concrete class. There are no instances of abstract classes and classes stereotyped as interface. Thus, properties defined by such classes are encoded as part of concrete classes inheriting or realizing them. Each class shall have a unique name within the application schema. The application schema may refer to or use classes defined in standardized schemas or other application schemas. The declaration of these classes shall either be included in the UML model that contains the application schema or accompany the application schema as a separate file.*

An object shall contain a set of property values. The object's class defines the properties and they can either be inherited through the 'class' supertypes or defined within the class itself. In order to differentiate between the different properties, the properties shall have names that are unique within its class. The property's data type governs the possible values and the multiplicity statement indicates the number of instances of the attribute in an instantiated object."

We specify the instance model by referring to GIM for transferring geospatial objects. In addition, we add auxiliary information about coordinate conversion in the geospatial dataset. When you digitize features drown on the base map, you need the geometric orientation to get parameters for conversion between map coordinates and ground coordinates. These parameters and data quality information such

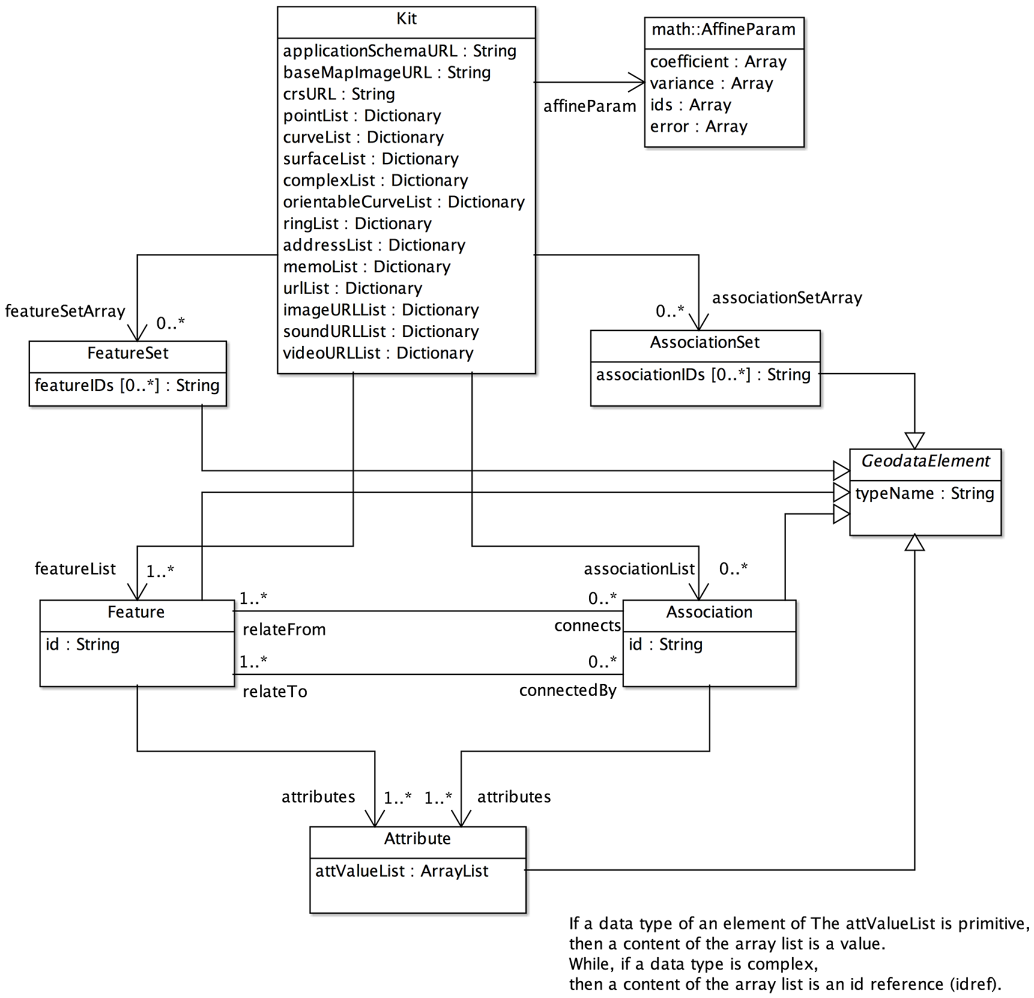


Figure 3.3-2. Instance model adopted in gittok.

as variance co-variance matrix of measuring errors are useful information for the transferring and the quality evaluation for reuse. Usually, such an information is not included in the application schema, because it is independent from the object description. Rather, such information may be included in the metadata. However, they should be included in the dataset, because it is possible to convert between map coordinates and ground coordinates without metadata. Information closely related to the dataset should be included in the dataset.

One more reason to define an instance model is that an application schema describes the conceptual structure of objects, while an instance model defines the physical structure to describe feature instances and association instances. The instance model shown in Figure 3.3-2 is schematized for gittok. This instance model aims for transferring of geospatial objects captured by map digitizing. In the figure, 'Kit' is defined as a class of geospatial objects including not only a collection of instances but also unique parameters for objects captured by map digitizing. Differences between application schema and instance model for gittok are explained below.

- Operations are not included in an instance model, because kit is a set of instances, but operation is so called a 'class method'. It should be implemented in the application as prepared function. Therefore, as far as a kit is transferred with its application schema to the client, operation syntax is un-necessary to include in each object.
- In the kit, the feature identifier and the association identifier are grouped by that type in the FeatureSet and the AssociationSet. And, elements of FeatureSetArray and AssociationSetArray are feature set and association set. These groups are index of instances. Instances can be searched effectively by their types. This mechanism is equivalent to a layer used in the conventional GISs. Indexing of instances are not included in application schema as application schema is a conceptual structure and not a physical structure of data set.
- Attribute values of complex data type are grouped and separated from feature instances. Attributes are stored in the dictionary, and instances links with their attributes by key-ids. In general, dictionary is a collection of (key, value) pairs called an associative array. In the case of attributes dictionary, key is a unique attribute id, and a value is the content of attribute. Conversely, for example, when geometric attributes are digitized by using the geometry editor, we think a geometry is a main object and feature instance is its adjunct. Such an equal relationship is possible by the separation between feature/association and attributes. Meanwhile, GFM does not care how to manage a set of attributes physically in a kit.

- featureList is a repository of all feature instances. A feature instance connects with a set of associations as from-features, and is connected by a set of associations as to-features. Similarly, associationList is a repository of all association instances. GFM also does not care how to manage and how to link different features by associations physically in a kit.

In addition, in the Kit class,

- applicationSchemaURL is a resource locator of the application schema.
- baseMapImageURL is a resource locator of the base map used for digitizing geometric and non-geometric attributes.
- crsURL is a resource locator of a set of parameters representing CRS including a geodetic datum, a vertical datum, a map projection, and a definition of the plane coordinate system. It is used for the conversion between plane coordinates and geodetic coordinates (latitude and longitude).
- affineParam is a set of parameters for the conversion between map coordinates and local plane coordinates on the ground. AffineParam contains coefficients of affine transformation (coefficient), a variance-covariance matrix for the quality evaluation of the orientation (variance), Identifiers of control points (ids), discrepancies between measuring points and control points acquired as a result of the map registration (error).
- pointList, curveList, surfaceList, complexList, orientableCurveList, and ringList are dictionaries as parts of geometric objects. In the case of geometric object dictionary, key is an id of the geometric object, and a value is the instance. addressList is a set of addresses indirectly showing the geographic positions.
- memoList is a set of memos.
- urlList, imageURLList, soundURLList, and videoURLList are dictionaries of unique resource locators for multifaceted (having multimedia attributes) data.

3.4. Schema for coordinate reference system (CRS)

CRS is defined as a “coordinate system that is related to an object by a datum” in ISO 19111(ISO/TC 211 2007a). The schema for gittok CRS is shown in the Figure 3.4-1. This is a profile of ISO CRS, which is simplified by the limitation of coordinate system (CoordinateSystem).

Geodetic datum (GeodeticDatum) is defined as a set of parameters to describe a shape of the earth as a reference ellipsoid (Ellipsoid). Name (name) of the ellipsoid, semi-major axis (semiMajorAxis) and inverse flattening (inverseFlattening) are the parameters of the ellipsoid. Inverse flattening is defined as, a/ (a - b). Where 'a' is a length of semi-major axis, and 'b' is a length of semi-minor axis.

The prime meridian (PrimeMeridian) is a starting meridian for measuring of longitude. It is a curve usually passing Greenwich in England and the east direction is plus. However, prime meridian can be put at any longitude. The offset of primary meridian is defined as the longitude from Greenwich (GreenwichLongitude). Usually it is not used in Japan.

Coordinate system (CoordiateSystem) consists of a definition of Axis and a definition of projection system. Each axis is defined by its name (e.g. 'x'), direction (e.g. 'north'), and the unit of measure (e.g. 'meter'). Projection system specifies the relationships between geodetic coordinates and plane rectangular coordinates. Gittok introduces Gauss-Krüger projection only. This projection is commonly used in the world especially as the Universal Transverse Mercator (UTM) projection. The plane

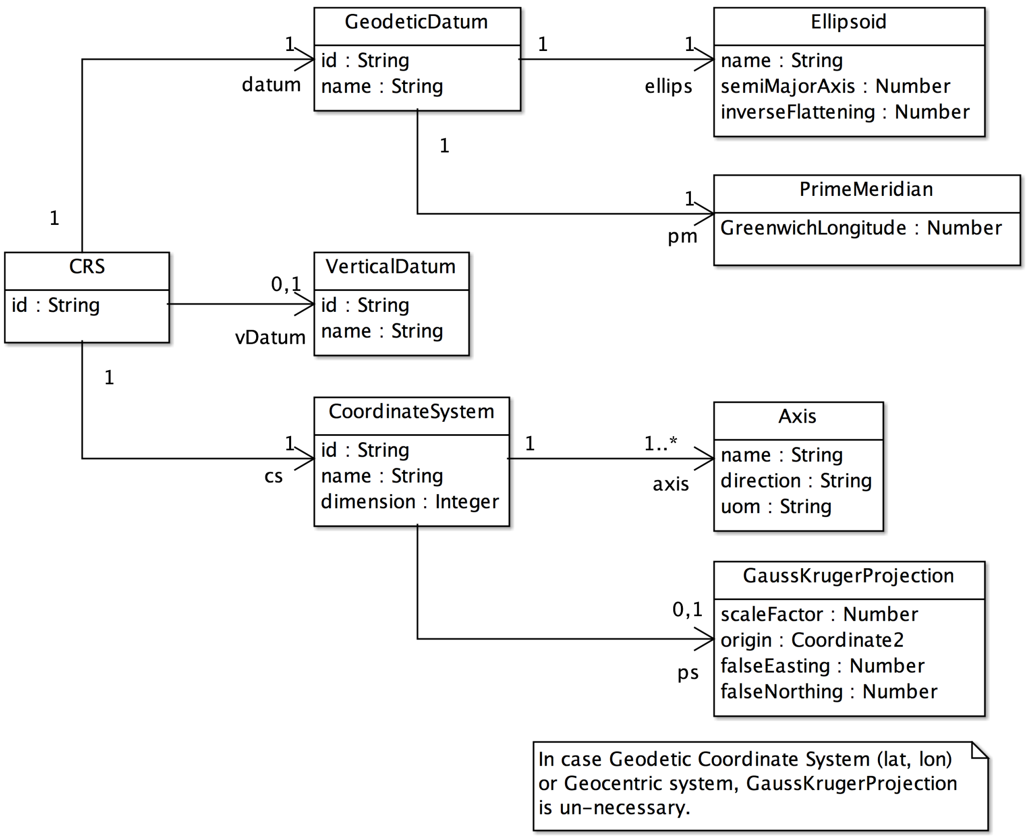


Figure 3.4-1. Coordinate reference system in gittok

rectangular coordinate system used in Japan is also the Gauss-Krüger projection system. The scale factor (scaleFactor) at the central meridian in each zone for the UTM is 0.9996, while the scale factor for the plane rectangular coordinate system is 0.9999. The territory of Japan is separated to 19 zones for the plane rectangular coordinate systems. Each system keeps the scale factor is between 0.9999 and 1.0001. The origin of plane rectangular coordinate system is defined for each local system. For example, for Zone No.9 including Tokyo, origin latitude is 36° N, and origin longitude is 139.8.333333° E.

Gittok adopted the formula of Gauss-Krüger projection proposed by (Kawase 2011), because, according to (Masaharu 2017), it was practically adopted in the calculation formulas of the Rules and Specifications of Public Survey in Japan replacing the former formulas (series expansion of longitude difference from the central meridian) in 2013.

4. Conceptual structures for cartographic objects

4.1. Introduction

According to the Oxford English Dictionary, the definition of cartography is "the science or practice of drawing maps." Meanwhile, more than 300 definitions of a map had been proposed since 1649 until 1996 (Andrews 1996). It means the definition of cartography had also evolved frequently. Recently, Kraak and Ormeling (2013) proposed the definition of cartography as follows,
"Making accessible spatial data, emphasizing its visualization and enabling interaction with it, aimed at dealing with geospatial issues." (Kraak and Ormeling 2013)

However today, according to the strategic plan described in the directory of the International Cartographers Association (ICA), the definition of cartography becomes rather un-clear as follows.

> *"With the introduction of computing and the growth of GI systems, the perception of Cartography is less clear to many than it was in the past. Definitions may be dated and open to different interpretations."* (ICA 2015)

The reason may be that the conventional scope of cartography has been limited to the graphic representation on a paper or a display device. Therefore, we should not hesitate to replace the term cartography by different interpretation. The new term should cover from conventional map making to advanced technology for augmented reality experiences. For example, virtual geo-experience technology may be one candidate to replace cartography. However, we need to make consensus before adopting the new term. Therefore, in this thesis, we tentatively use the term cartography as a technology for representation of geospatial information until the new term will be fixed.

Today, there are many chances to see cartographic products such as navigation map, facility guidance map, map for games, augmented (or mixed) reality, and so on. They are mechanisms to show geospatial information through man-machine interaction with personal computers, tablets, smart phones, headsets and so on.

However, it is essential that we have to understand the conversion procedure from geospatial objects to cartographic objects, then from cartographic objects to cartographic products. Cartographic objects in the surface structure are resource to produce a map face as a collection of symbols and labels. In gittok, these are expressed using symbol and label styles registered in SSD and LSD. Then, cartographic objects are edited and converted to the cartographic product for visualization on the man-machine interface.

In this chapter, we will discuss the issues on GPM and GLM enabling the software implementations for designing portrayal schemata and list schemata, and then instance models for map, gazetteer and list enabling the description of cartographic objects in the system independent formats.

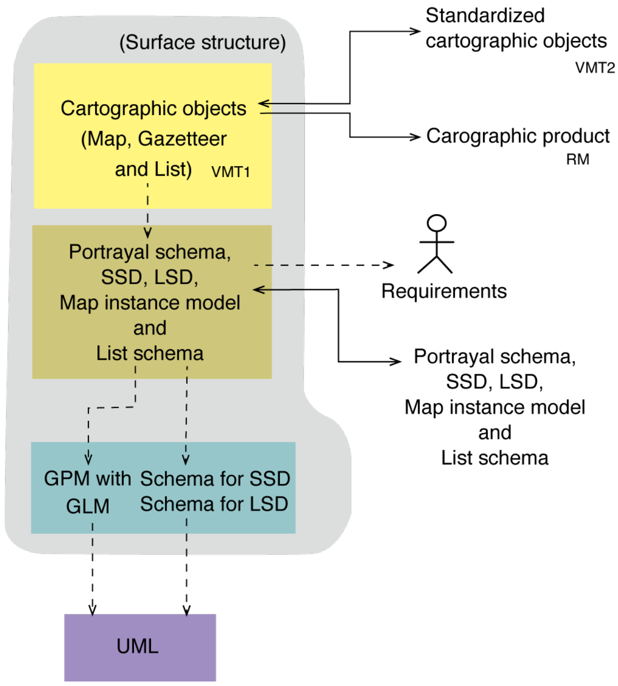


Figure 4.2-1. Meta models, models and an instance for mapping and listing.

4.2. General portrayal model (GPM)

4.2.1. Introduction

In this section, we introduce GPM, the schema for SSD, the schema for LSD, and GLM as a part of GPM. Figure 4.2-1 shows the surface structure including these metamodels. This diagram is a part of the concept map of GIT on the view point of objects shown in Figure 2.6-1.

4.2.2. GPM

Various kinds of software tools for geospatial representation are currently in use. General purpose GISs such as ArcGIS and QGIS have strong functions for map representation. Google, the Open Street Map foundation (OSM), National mapping organizations such as GSI of Japan, and so on provide the web map APIs to make applications on their services. The textbook on D3.js introduced four software tools for mapping (Kartograph, Leaflet, Modest Maps, Polymaps) (Murray 2013). Mapmap.js is a tool to design interactive thematic maps displayed in web browsers (Ledermann and Gartner 2015). These tools are widely used for map representations. However, many of them do not have capability to use geospatial objects following application schemata, which refer to GFM provided by GI standards, because they rarely use the object-oriented modeling of features. Therefore, it is not so easy to embed existing tools as a representation software component in gittok. This is the reason that representation mechanisms were developed from scratch.

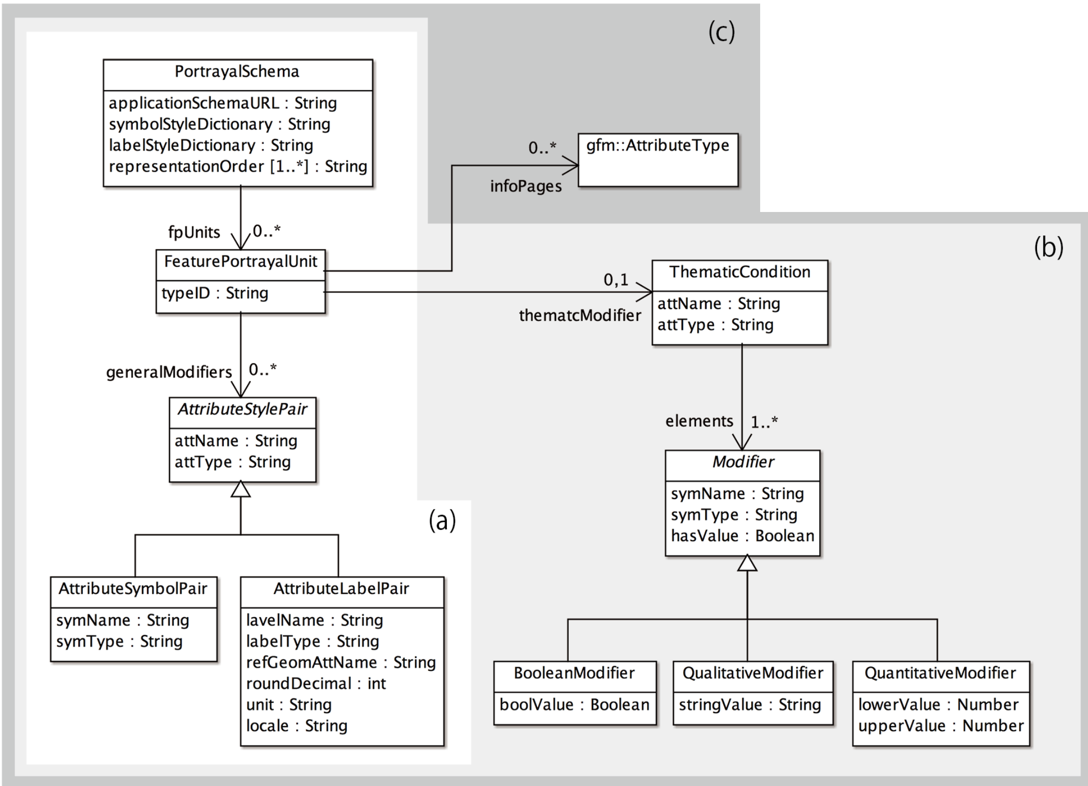


Figure 4.2-2. UML class diagram to show general portrayal model (GPM). Area (a) specifies a metamodel for general-purpose maps, (b) for thematic (choropleth) maps, and (c) for interactive maps.

Nevertheless, gittok GPM (Ota 2016) is a metamodel that enables the instantiations of portrayal schema for general-purpose maps, thematic choropleth maps and interactive maps to show information pages. Information page is a special balloon page to represent different kinds of feature attributes such as images, videos, audios, addresses, memos and websites.

Figure 4.2-2 shows the structure of gittok GPM. Primary objective of GPM is to determine the representation rules for each type of features. Therefore, a combination of feature type and its modifier to declare a representation style is an essential element for representation. In the figure, this is represented by FeaturePortrayalUnit. This type relates with general modifiers (AttributeStylePair) for general purpose maps, thematic modifiers (ThematicCondition) for choropleth maps, and information pages (gfm::AttributeType) for interactive maps. These three items may be used together to represent, for example, interactive general-purpose maps.

If a map is a general-purpose map, AttributeStylePair defines the symbol style or a label style used for the representation of a feature. ((a) in Figure 4.2-2)

In the case of a choropleth map, ThematicCondition is used to define a color included in the symbol type (symType) used to represent the ranks of attribute values (elements). The elements for the representation of ranks are defined in order to reflect the data type of an attribute. According to the attribute data type, specialized modifiers are selected as follows. ((b) in Figure 4.2-2)

- 'BooleanModifier' defines the symbols according to true or false.
- 'QualitativeModifier' defines the symbols according to the string value.
- 'QuntitativeModifier' defines the symbol according to the interval of extent (lowest number and highest number).

For example, to make a map to show population density distribution of census districts in a city, the geometry used to show a district is displayed as an area drawn by a suitable color for ranking, chosen from a set of area symbols. Those ranks indicate population density extent, which is defined by the quantativeModifer.

Furthermore, if feature type is multifaceted, its attribute types (infoPages) can be shown as information pages (Figure 4.2-2 (c)). Examples of information pages are illustrated in Figure 8.8-7.

4.2.3. Schema for symbol style dictionary

A schema for symbol style dictionary (SSD) specifies rules for a dictionary describing symbol styles (Figure 4.2-3). Symbol styles are classified into line symbol style, area symbol style, point symbol style, and complex symbol style, as geometric elements in two-dimensional space are point, line, area and their combination.

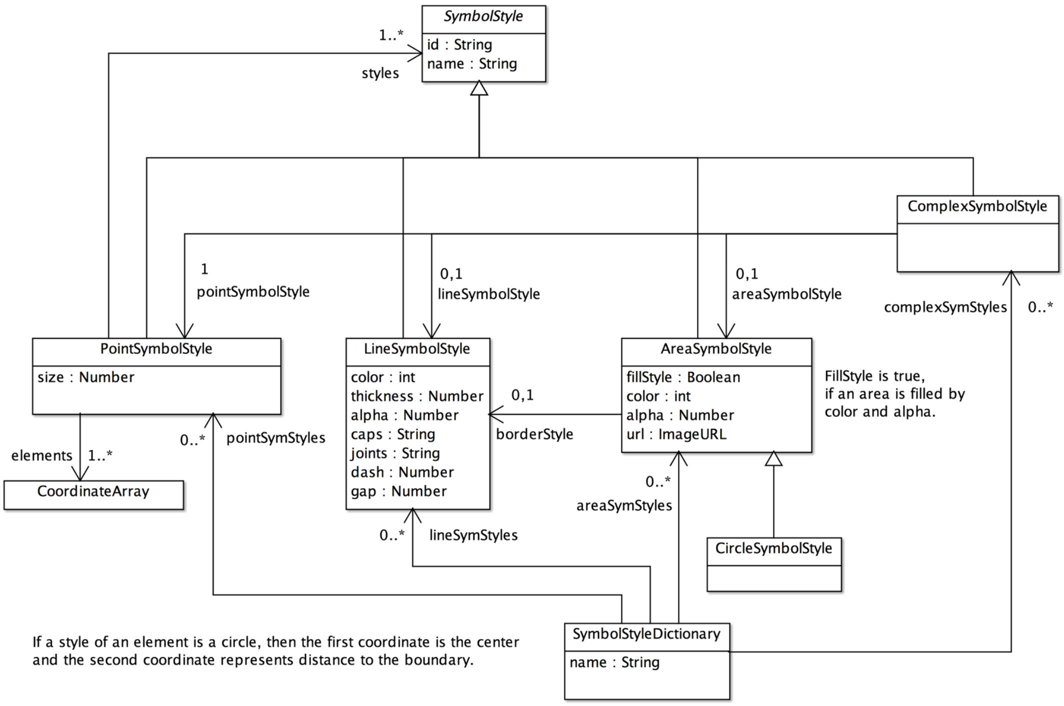


Figure 4.2-3. UML class diagram to show a schema for SSD.

In gittok SSD, line symbol style (LineSymbolStyle) is a style of a curve declared by color, thickness, alpha value (opacity), cap style of both edges of a curve, joints (shape of a vertex), and dash and gap of a dashed curve (real line, if both parameters are zero).

Area symbol style (AreaSymbolStyle) is a style of a closed surface declared by a style for filling inside of a surface. If the inside is filled by color, RGB code and alpha value are declared. If the inside is filled by texture pattern, the URL of image is declared. The style of the boundary curve is also declared, if it is required. Circle symbol style is a specialized type of area symbol style. All attributes of circle symbol are inherited from area symbol style.

Point symbol style (PointSymbolStyle) is a style of a point declared by its size, coordinate arrays representing curves and areas as parts of a point symbol. For example, if you want to make a point symbol as a circle that is inside of a square, you have to combine circle and closed polyline. This point symbol style is defined as a combination of circle symbol style and line symbol style. Therefore, point symbol style is defined as a set of other symbol styles.

Complex symbol style (ComplexSymbolStyle) is a style of a combination of point style, line style and area style. This is used to represent a geometry complex following SG_Complex.

Finally, a type of symbol style dictionary (SymbolStyleDisctionary) is defined as a combination of pointSymStyles, lineSymStyles, AreaSymStyles and complexSymStyles. SSD is an instance of SymbolStyleDictionary.

4.2.4. Schema for label style dictionary

A schema for label style dictionary (LSD) specifies rules for a dictionary describing a label style (Figure 4.2-4) as a set of label styles. The label style consists of a label identifier (id), label name (name), font type (font), font size (fontSize) in pixel, RGB value (color), opacity value (alpha), and reference position index (reference).

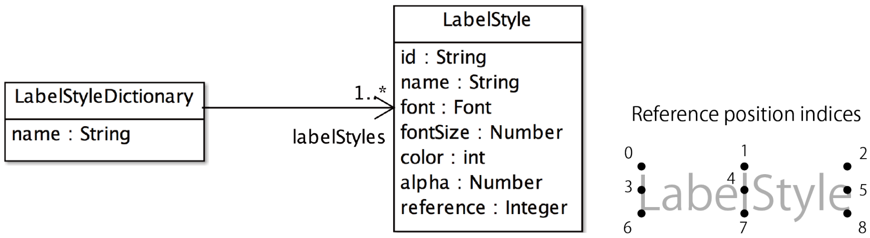


Figure 4.2-4. UML class diagram to show schema for LSD.

By the way, we do not discuss visual variables originally proposed by Jacques Bertin (Bertin 1983). Actually, position, size, shape, brightness, hue, orientation, and texture are essential parameters for designing symbols and labels. Therefore, we are providing the schema for SSD for designing symbols with size, shape, color (RGB and opacity), and texture pattern. However, the study about graphic design technique for symbol and label is out of scope for this thesis, because it is a subject regarding the cartographic design.

4.3. General list model (GLM)

General list model (GLM) is the metamodel for list schemata. It is described as a part of GPM, as list is a portrayal means for virtual features. The role of GLM is to make it possible to select the feature types included in the list, and the attributes contained in the list to explain each feature type. The structure of GLM is shown in Figure 4.3-1. A list schema (ListSchema) has a set of selected feature types (featureTypesForListing) for listing. The proxy attribute of selected feature types shall be non-geometric. Each feature type associates with selected attribute types (attributeTypes) for listing.

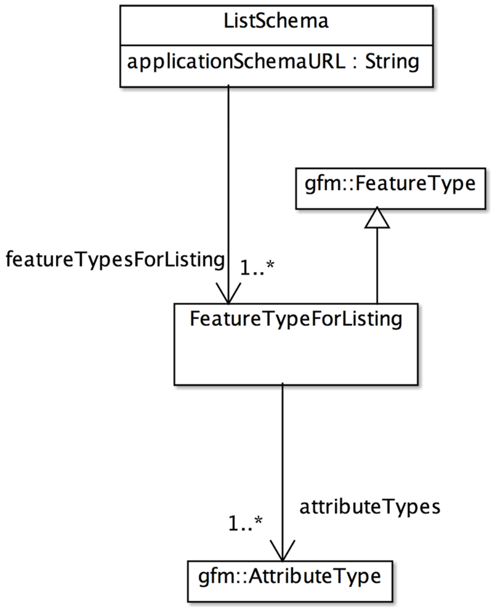


Figure 4.3-1. UML class diagram to show GLM.

4.4. Instance models for map, gazetteer and list

4.4.1. Map instance model

Map instance model in accordance with the portrayal schema is described in this section. A map can be shown on the display device by indicating the screen coordinate origin (x, y), map size (width, height), map scale, and color with alpha value for the background of the map are declared. All of them are attributes of a map. A map also keeps the correspondent kit file name as reference.

A map consists of two components. They are a map face as a set of symbols and labels, and marginalia as information about the map. They are illustrated in accordance with SSD and LSD. Marginalia consists of title, subtitle, north arrow, bar-scale, and legend. The subtitle can be freely used to describe production information including authorship, publication date, map projection and other information of that the user should know. Symbols may associate with information pages as required, if a map is interactive.

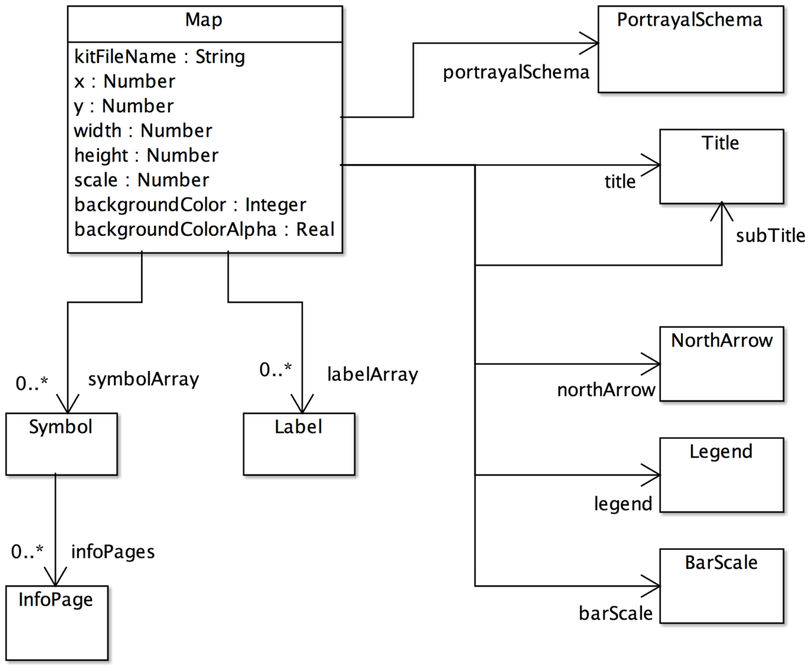


Figure 4.4-1. Instance model for general-purpose map briefly illustrated in UML.

4.4.2. Gazetteer instance model

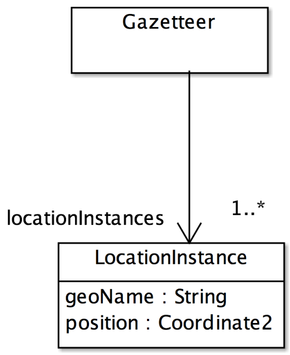


Figure 4.4-2. Gazetteer instance model for gittok illustrated in UML.

Gazetteer is an adjunct of a map, because it is an index of features drawn on the map. A gazetteer is produced as a part of mapping procedure. Figure 4.4-2 shows the instance model for gazetteer. Gazetteer is a set of location instances. It can be produced automatically as long as the combination of geographic name (geoName) and the position of feature (position) can be selected. According to ISO 19112 (ISO/TC 211 2003b), it is defined as a directory of geographic identifiers describing location instances. Therefore, LocationInstance is defined as a combination of non-geometric attribute and geometric attribute in a feature.

4.4.3. List instance model

The instance structure of a list is shown in Figure 4.4-3. An instance of GeoList is the list produced from the geospatial objects in compliance with the list schema. GeoList consists of two attributes. They are fTypeForListing and listElements. The former is a feature type for listing, and the latter is a collection of elements of the list. The list element (ListEelements) is formed by a set of attributes selected from each feature instance. ListElement keeps the name of proxy attribute (proxyValue) brought from the original feature. ListElement also has items (ListItem), in which the attribute name (itemName) and attribute values (itemValues) are stored.

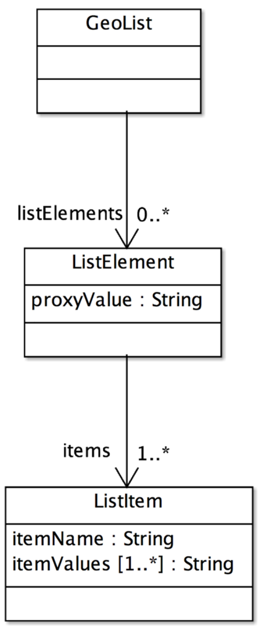


Figure 4.4-3. Instance model for list illustrated in UML.

5. Conceptual structure for geospatial functions

5.1. Introduction

Conceptual structure for geospatial functions is introduced in this chapter. As already discussed, geospatial functions are included in the deep structure. They are modeling, acquisition, analysis, and exchange. In this chapter, we will discuss those functions in detail and propose the conceptual frameworks for those functions.

5.2. Modeling

5.2.1. Introduction

Modeling consists of a function to provide application models by abstraction of the real-world phenomena, and a function to provide portrayal models for designing of cartographic products. A formal description of conceptual model for geo-processing is called an application schema, which is modeled in accordance with GFM. While, cartographer designs the geospatial representation from the given geospatial objects using the portrayal schema in compliance with GPM. Portrayal modeling will be discussed in the section 6.2, as it is included in the surface structure discussed in Chapter 6.

Today, it is not so easy to find GI systems applying GFM as a metamodel for application modeling. As a matter of fact,

> *"One of the most important advantages for the automated spatial data model was the development and application of dual data systems that handled graphics and attributes separately"* (Foresman 1998.)

The dual data system is explained as follows,

> *"The spatial key is distinct, as it allows operations to be defined which are not included in standard query languages"* (Goodchild 1992)

As of 1992, it certainly was as explained by Goodchild. However, in the same year, the object-oriented GIS called TIGRIS was introduced and the extension of SQL to realize geometric and topological data handling in DBMS was proposed (Herring 1992). Then, in 1999, ISO/IEC established "ISO/IEC 13249-3:1999 Information technology -- Database languages -- SQL Multimedia and Application Packages -- Part 3: Spatial", and then geometric data type can be handled in ORACLE database, PostgreSQL and so on (Ota 2014). Moreover, GFM defined in ISO 19109 (ISO/TC 211 2005) is a collection of rules for application schema, and GFM is modeled based on the object-oriented technology. According to Ahearn,

> *"geospatial models in object-oriented systems offer four distinct advantages over procedural layer-based models: (1) the real-world object is the basis for abstraction, not its geometry; (2) topology can be set not just between entities of a single type (i.e. within layer) but between multiple entity types; (3) a richer set of associations is possible and (4) modeling is more extensible owing to the principles of inheritance, encapsulation, and polymorphism"* (Ahearn 2008).

In addition, abstract class can be defined in the object-oriented technology. For example, we may declare that building, bridge, road, and park are all artifacts as an abstract class. Artifact is an imaginary feature (abstract class), but there is a possibility to define abstract concept to make the object structure consistent by inheritance. However, it is difficult to define abstract feature type for the conventional layer based dual data systems, because feature types should be defined as physical features as discussed in the section 2.5. It means that abstract feature type and a feature type without geometry is not so easy to define in most of the conventional GI systems.

Meanwhile, we should keep in mind that UML class diagram such as an application schema is essentially created for designing of the software. For example, UML is explained as follows,

> *"On the surface, the Unified Modeling Language (UML) is a visual language for capturing software designs and patterns. Dig a little deeper, though, and you'll find that UML can be applied to quite a few different areas and can capture and communicate everything from company organization to business processes to distributed enterprise software."* (Pilone and Pitman 2005).

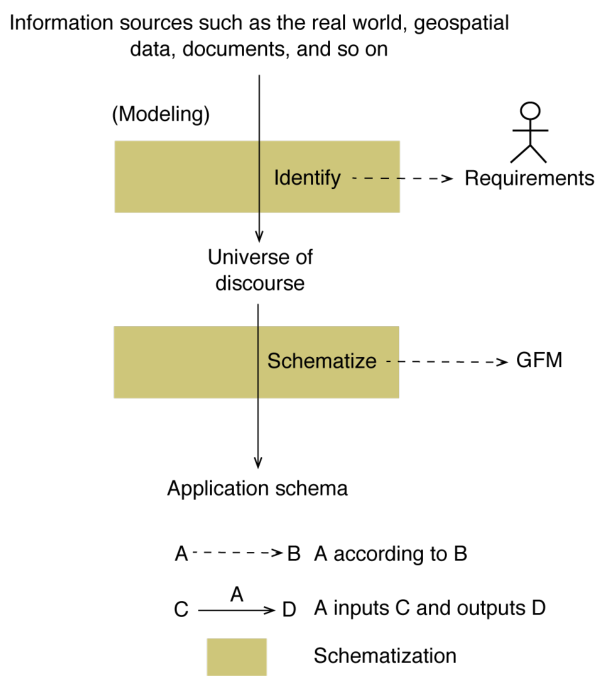


Figure 5.2-1. Operations for modeling

As far as an application schema is a formalized description of the universe of discourse, we need to identify the universe of discourse before schematization. Therefore, Figure 5.2-1 shows two functions, ‘Identify’ and ‘Schematize’, for modeling of an application schema.

Functions are instances of the metamodel for functions shown in Figure 2.7-1. In the following sub-sections, we discuss the identification of the universe of discourse, and the schematization of an application schema.

5.2.2. Identification

Identification of the universe of discourse is a process to make consensus. For example, road administrators in a municipal authority discuss and make consensus among officials about that features (street, sidewalk, lamppost, street furniture, roadside tree and so on) and their properties (name, geometric shape, date of announcement for shared use, record of road surface investigation, simulation algorithm to get road service level and so on, in the case of street) should be included in the universe of discourse to construct their road management system. The document as an output of identification describes a set of features included in the universe of discourse. Actually, the know-how of discovering user requirements is one of the most important knowledge for modeling (Hansen, Berente, and Lyytinen 2009). However, we do not discuss this issue in detail in this thesis, as it is mainly a matter of the stakeholders who have responsibility of the identification of the universe of discourse.

5.2.3. Schematization

Here, schematization means to provide an application schema by referring to the universe of discourse. The purpose of the application schema is the formalization of the universe of discourse. Features and feature associations defined in the application schema may hold attributes and operations as their properties. For example, imagine that the user's request is "I want to know the shortest path from any intersection to any other intersection". In that case, objects representing a road network is required. Also, the road network should equip an operation for the shortest path finding. The application schema shown in Figure 5.2-2 illustrates an example of schematization. A shortest path finding is performed by giving an information indicating the start point or the end point to the attribute of the intersection called startEnd. The WarshalFloyd method was selected for the path finding in this application schema, and the result is stored in the attribute called shortestPath. By the way this example will be used at the section 8.7.

Schematization is a collaboration work by modeler and stakeholders. Modeler draws a draft and improves it with stakeholders until consensus is made. Stakeholders participating in the schematization work should preferably have GFM literacy. Moreover, to create a reliable application schema, the software that can be used as a testbed should be required. Gittok is not a testbed software, but it will be useful to understand what the testbed should perform. We will discuss the software module called 'Modeler' in the section 8.4.

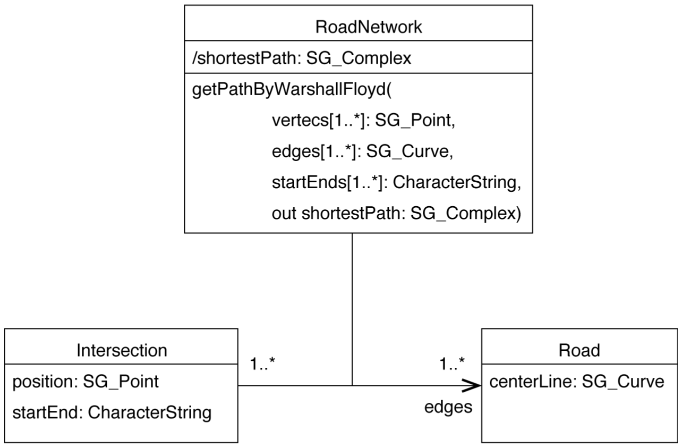


Figure 5.2-2 Application schema for shortest path finding.

5.3. Acquisition

5.3.1. Introduction

Acquisition means buying or obtaining geospatial objects. Today, there are at least two ways to get geospatial objects. If already the adequate objects exist, we will purchase it or get it free. If there are no objects exist, we will capture the objects by measuring or observing by ourselves, or request capturing to the professional. In actual projects, different methods are often combined to acquire a set of geospatial objects.

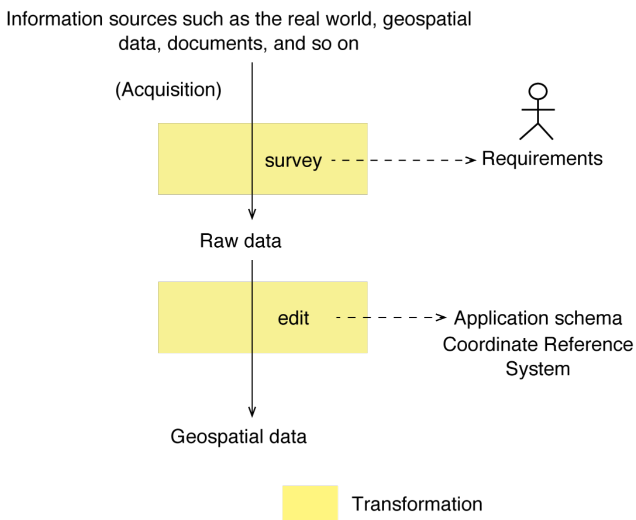


Figure 5.3-1. Data acquisition

Usually, data acquisition is done in accordance with the specification, which includes the application schema. In the case of Data Product

Specifications Version 1.0 provided by EuroGraphics as a part of Reference Information Specification for Europe (RISE) (Illert and Afflerbach 2007) , the specification consists of,

i. Specification scope
ii. Data product identification
iii. Data content and structure (application schema, Feature catalogue)
iv. Reference systems
v. Data quality
vi. Data capture
vii. Data maintenance
viii. Portrayal
ix. Data product delivery
x. Metadata

This document was provided by referring to ISO 19131 – Data product specification (ISO/TC 211 2007b). The purpose of the standard is to provide practical help in the creation of such specifications (Kresse, Danko, and Fadie 2011).

Nevertheless, a common sequence of data acquisition is listed below (Ota 2014) (Figure 5.4-1):

i. Surveying: getting the raw data including geometries (positioning, map digitizing, projection conversion, and so on), imageries (image capture using digital camera, remote sensing, laser scanning, rectification, and so on), environmental indicators (climate, sea level, pollutants of air, water and soil, wastes, and so on), movies, audios, and documents and so on, in accordance with the application schema.
ii. Editing: compilation of the geospatial objects by converting and embedding row data as attributes. Then, the geospatial objects are stored as a persistent file.

5.3.2. Surveying

There are various surveying methods in order to get the geospatial raw data. Various technologies such as 1) numerical surveys by using photogrammetric instruments, GPS, total stations, active and passive sensors and so on, 2) digitization of sheet maps and documents, and 3) transformation of legacy data, are commonly used (Nakamura and Shimizu 2000). In addition, surveying has to be performed in accordance with the corresponding laws and regulations. Moreover, data reliability depends on the skill of engineers. Therefore, the project planner should document the requirements for surveying methods, product specifications, expected skill of surveying engineers, and special demands for the project.

5.3.3. Editing

Geospatial object editing is a process to put together previously obtained items as attributes in the object. We should edit objects with consideration for the characteristics of feature attributes. For example, the coordinates in the geometric attributes should be converted so that they are included in the same coordinate reference system (CRS). Traditionally, such a conversion is called a map registration. However, simultaneously, we should remind, if the geographic extent of data exceeds the limitation of the CRS or not. We need to estimate the systematic error caused by the CRS. In the case of Japanese plane rectangular coordinate system using the Gauss-Krüger projection, the east and west limit of the geographic extent is about 130km and the distance from the central meridian will be about 13m longer than the length on the ground, because the scale factor is 0.9999 at the origin and 1.0001 at the limits of geographical extent. In reality, there are a few features with a size of 130 km. Even if there is a feature with a size of 1 km in the east-west direction near the boundary, the relative accuracy of size is 1000 * 0.0001 = 0.1 m. Therefore, the CRS should be selected after considering the size of the target area and the accuracy requirement of the coordinate data.

Next, we should be careful to digitize geometry from existing resources. For example, let's imagine that the geometries of buildings digitized from 1/1,000 map that was enlarged from 1/2,500 map. According to the Japanese public survey regulation, the positional accuracies (root mean square error (RMSE)) for 1/2,500 maps (σ_0) shall be less than 0.7mm (1.75 m on the ground), while the RMSE for 1/1,000 maps (σ_1) shall also be less than 0.7mm (0.70m on the ground). Even if buildings were digitized in compliance with the regulations, the RMSE (σ) of integrated data will be estimated by applying the law of error propagation as follows.

$$\sigma = \sqrt{{\sigma_0}^2 + {\sigma_1}^2} = 1.88$$

Similar problem happens, for example, when the ortho-image is used as the base map displayed on the screen for geometry digitizing. Thus, we should evaluate carefully that σ fulfils the user requirements or not.

Also, when we put together the attributes from different resources, we should confirm if the rules to distinct the quality level is the same. Moreover, if the extents of statistical area or the grid system are different, we have to solve how to accommodate the thematic attributes in one system. These problems happen when we compile raw data from different resources.

Today, open data policies steering by the national and local governments are widespread in the world. Volunteered geospatial information is also available. These trends will evolve forward more, if the people involved in the data compilation or mashup will accumulate know-how and literacy for making the reliable data for users. Meanwhile, ISO 19157 – Data quality (ISO/TC 211 2013) specifies five quality elements. They are completeness, logical consistency, positional accuracy, thematic accuracy and temporal accuracy. We need to consider the required level of data quality before surveying and editing by referring to such a standard. And we should indicate the quality level in the requirements specification. Then after the data editing, we should confirm the actual qualities and open them as the quality report for users.

5.4. Analysis

5.4.1. Introduction

In general, analysis is a process of separating a complex substance to smaller parts as new information in order to gain a better understanding of phenomena in the real world. According to Dibiase, et al.,

> *"This knowledge area encompasses a wide variety of operations whose objective is to derive analytical results from geospatial data."* (Dibiase et al. 2006)

However, from the viewpoint of GIT, the matter of interest is a methodology enabling analyses. Because, GIT is a platform technology for GI applications. Conceptual analysis model is illustrated in Figure 5.4-1. By referring to the request from the user, the analysis module extracts appropriate operation type included in the feature type or association type, and object attributes as arguments. Then the operation is executed, and the return value is embedded in an object as a derived attribute.

5.4.2. Extraction of attributes and operation

According to the structure of application schema, operation syntaxes are defined in the application schema. Arguments are parts of object. For example, a sequence of extraction in gittok is as follows.

i. Load a kit and an application schema in gittok. The kit is loaded by the selection of the file name, or by evaluating the metadata in the management module called ‘Manager’.

ii. Extract a feature or an association as a target of operation from the kit.

iii. Select the operation for the particular analysis from the operation list included in the feature type or the association type in the application schema.

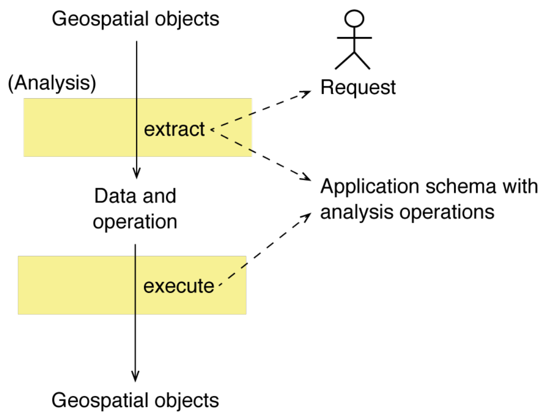


Figure 5.4-1. Conceptual function model for analysis

5.4.3. Analysis execution

The application calls the operation class contained in the class repository, designates the operation contained in the class, and executes it by entering necessary arguments. Most of the operations are the static method defined in the class. The return value is stored in the derived attribute variable in the appropriate feature or association.

5.5. Exchange

5.5.1. Introduction

Exchange is classified into encoding and decoding (Figure 5.5-1). Encoding transforms data to data with system independent format, and decode transforms data with the system independent format to data with internal format. There are many types of data. For example, kit as geospatial objects, application schema, CRS, metadata, portrayal schema, and so on may be exchanged between different systems. We will discuss encoding and decoding in detail at the following section.

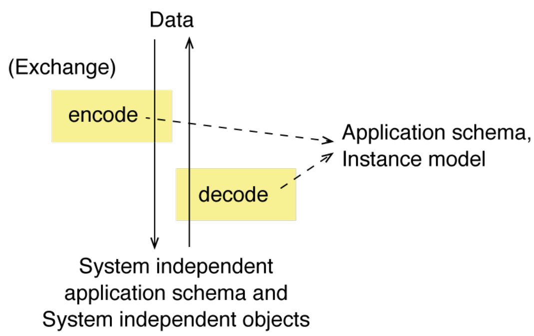


Figure 5.5-1. Conceptual function model for data exchange

5.5.2. Encoding and decoding

Figure 5.5-2 is an overview of data exchange between two systems illustrated in ISO 19118 –

Encoding (ISO/TC 211 2011b). According to this standard, the following logical steps are taken in order to translate objects from the system A to the system B.

i. The system A translates its internal data to a data structure following the application schema I. In Figure 5.5-2, this translation is denoted by M_{AI}. The result is an application schema specific data structure i_A. This data structure is system dependent and thus not suitable for transferring.

ii. Encoding service, which applies the encoding rule R following the instance model, creates a data structure that is system independent. This encoded dataset is called d.

iii. The system A then invokes a translation service to transfer the dataset d to the system B. Parties managing A and B must agree upon the transferring protocol used.

iv. The translation service in the system B receives the dataset d.

v. The system B applies the decoding rule R^{-1} in compliance with the instance model to interpret the encoded data in order to get an application schema specific data structure i_B.

vi. The system B translates the application schema specific data structure i_B into its internal data B. This is done by defining a translation M_{IB} from the application schema into its internal schema.

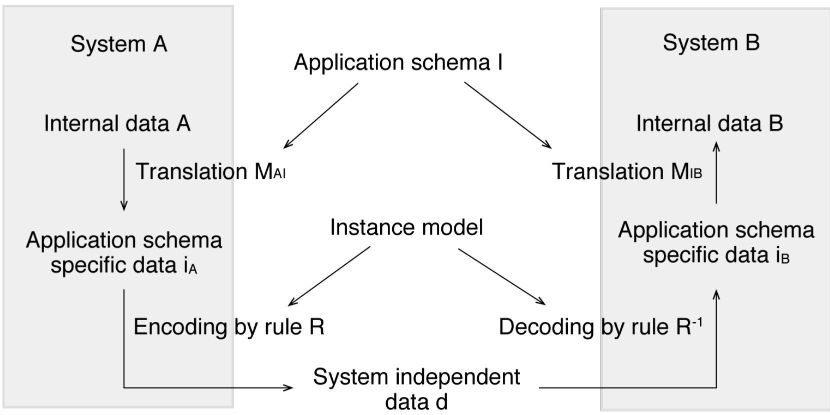


Figure 5.5-2. Overview of data exchange between two different systems (modified from Figure 1, ISO 19118:2011)

The exchange model illustrated in Figure 5.5-2 is not appropriate for exchanging the syntax of operation. Because, operations are not included in the instance model. However, the feature types and association types defined in application schema contain operation syntaxes. The operation can be thought as a dynamic attribute, because the return value will become the derived attribute in the object, as already described in the section 5.4. If the receiver can replicate not only attributes but also syntaxes of operations, the receiver would be able to consider the better reuse plan. Therefore, transferring or sharing of not only data but also the application schema with operation types should be realized. The syntax rule for gittok will be proposed in the section 8.7 in detail.

GML is not used for the exchange by gittok at this time, because gittok uses a simplified GFM, and because GML is difficult to comprehend for beginners in short period of time. Of more importance, it is expected that students who learned the concepts of exchange in gittok will be able to understand ISO standards and OGC standards easier in the advanced education.

Gittok does not support the data exchange with conventional geospatial formats in use today, because there is a possibility of data loss during conversion to other formats. Application schema is modeled in accordance with gittok GFM. Therefore, a kit may contain feature instances having derived attributes, inheritance relationship with the parent feature, and associations between features. While, the descriptions of them are not so easy for commonly used formats such as Shape and GeoJSON. For example, GeoJSON objects represent geographic features only and do not specify associations between geographic features (https://datatracker.ietf.org/wg/geojson/charter/ accessed 2018-08-30). However, gittok enables to output SVG files and GeoJSON files for mapping of physical features. And in future, it would be possible to develop the capability to convert and embed legacy data in kits.

6. Conceptual structure for cartographic functions

6.1. Introduction

We will discuss and propose conceptual structures for cartographic functions in this chapter. Representation is one of the six knowledge areas in GIT, but at the same time, it contains various functions in the surface structure. First, there is a function to convert geospatial objects into cartographic objects. A portrayal schema that is a set of rules for describing the cartographic objects is referred for this conversion. A map editing function is required to transform cartographic objects to a map. Sometimes a map also comes with a gazetteer. Also, in order to distribute cartographic products to users, it is necessary to convert cartographic objects to commonly used formats such as SVG. Meanwhile, virtual features included in geospatial objects are converted into a list. Then, the list is converted to commonly used language such as HTML for the distribution to users. By the way, maps and lists in SVG and HTML are not included in the knowledge of exchange, because, they are commonly-used graphics so far, and they do not follow the application schema nor the portrayal schema anymore. In this chapter, we will discuss and propose conceptual structures of cartographic functions.

6.2. Map design

6.2.1. Introduction

In this section, we discuss the identification of parameters for map design, and then the schematization of protrayal schema with SSD and LSD. Tyner (Tyner 2010) explained map design as follows,

"Design is a decision-making process. Many choices must be made in order to create an effective map whether for visualization or presentation. Before beginning, there are number of questions to ask. The answers to these questions determine what projections, symbols, scale, colors, type, and other components will be chosen".

Tyner proposes eight questions for map design as follows,

"Is a map the best solution to the problem?
What is the purpose of the map?
What is the subject or theme of the map?
What is the intent of the Map?
Who is the audience?

What is the format?

How will it be produced?

How will it be reproduced, disseminated, or viewed?"

Answers to these questions should be reflected in the identification of the map design parameters. Identifications of these parameters are described as the universe of discourse for the map representation (Figure 6.2-1). In this section, we will discuss the identification of map design parameters, and then the schematization of the protrayal schema with SSD and LSD.

6.2.2. Identification

Identification of map design parameters specify the universe of discourse by referring to the purpose of mapping, existing geospatial objects, and supplemental documents. The universe of discourse should comprise the design parameters for mapping at least as follows.

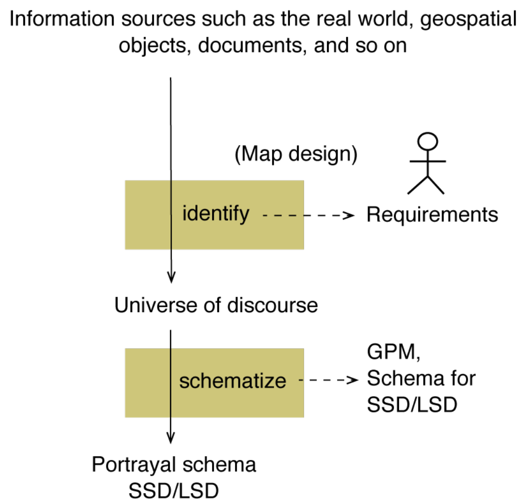


Figure 6.2-1 Conceptual function model for map design

- Purpose: general purpose or thematic.
- Media: paper, graphic display, headset, and so on.
- Projection: local plane coordinate system, UTM, Gauss-Krüger projection, and so on.
- Symbol style: geometry (point, line, and area), size, weight, color with opacity, and representation priority.
- Label style: font set, font size, weight, color with opacity, and reference position.
- Layout: subject area for map face, title, legend, scale, orientation, supplemental text, background color, and frame.

6.2.3. Schematization

The portrayal schema together with SSD and LSD is produced by schematization to realize the universe of discourse. However, the portrayal schema is a set of rules for describing the cartographic objects those represent a map face. Marginalia will be designed in the mapping process described in the next section. The procedure of the schematization in gittok is as follows.

i. Design of symbol styles: line, area, point, and complex

ii. Design of label styles for the representation of text

iii. Schematization of the portrayal schema for general-purpose map, choropleth map, or interactive map in accordance with GPM.

6.3. Mapping

6.3.1. Introduction

Principally, the mapping function transforms from physical features to a map face automatically in compliance with the map instance model, and by referring to the portrayal schema (Figure 6.3-1). The map instance model was described in the section 4.4.1. When, it is difficult to distinguish labels caused by overcrowding or overlapping on symbols, a map designer fine-tunes the label positions manually to avoid the difficulty of interpretation in the case of gittok. Then, the map designer sets up marginalia including the title, the north arrow symbol, and the bar scale in the compilation procedure. By the way, efficient algorithms for automatic label placement has already been proposed (Kameda and Imai 2003). We will try to implement such an algorithm in gittok in future.

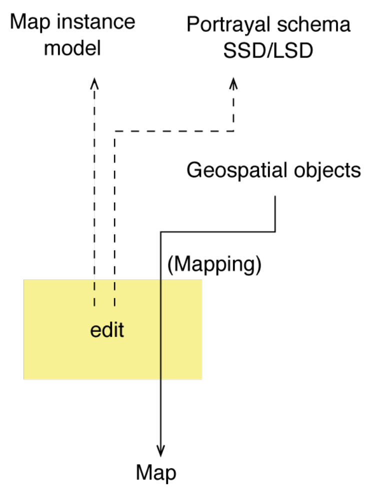


Figure 6.3-1. Function for mapping.

6.3.2. Map editing procedure

The map editing procedure is as follows.

i. Select the kit with physical features for mapping.

ii. Select the portrayal schema.

iii. Execute the conversion from geospatial objects into cartographic objects in accordance with the portrayal schema for the configuration of a map face.

iv. Tune the map scale and the centering of the map face.

v. Fine-tune the positions of labels, the legend, and the bar scale.

vi. Place the title, sub-title, and north arrow symbol.

vii. Save the map file for reuse. And save the html with svg elements or GeoJSON file for distribution.

6.4. Gazetteer compilation

6.4.1. Introduction

The function for gazetteer compilation transforms the physical features to a set of location instances automatically in compliance with the gazetteer instance model by referring to the portrayal schema (Figure 6.4-1). The gazetteer instance model was illustrated in the section 4.4.2. The gazetteer is used to shift the particular symbol to the center of the screen by indicating the place name listed in it.

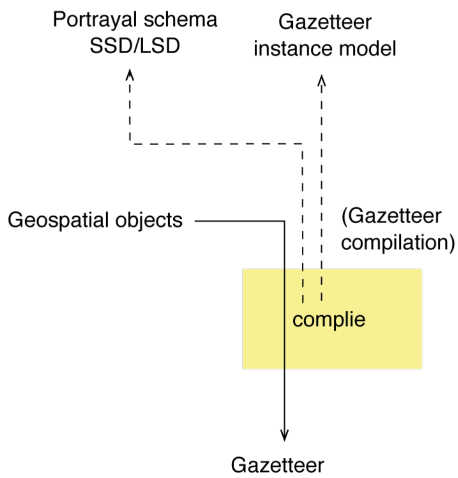


Figure 6.4-1. Gazetteer compilation procedure

6.4.2. Gazetteer compilation procedure

The gazetteer compilation procedure is as follows.

i. Select a kit for the compilation of the gazetteer.

ii. Select a feature type including a geographic name and a geometric attribute.

iii. Execute automatic compilation. If a geometric attribute is a curve or a surface, a center laying on the curve or a center of maximum inscribed circle (MIC) in the surface will be the reference position of a feature.

iv. Save the gazetteer in the storage folder.

6.5. List design

6.5.1. Introduction

GLM is a metamodel to show the rules for list schema. A list as cartographic objects is produced in accordance with the list schema as an instance of GLM shown in Figure 4.3-1. In practice, list is an array of tuples representing a combination of selected feature attributes. The list design consists of two functions: identification of the universe of discourse, and schematization of the list schema in compliance with the universe of discourse (Figure 6.5-1).

6.5.2. Identification

The function 'identify' is a transformation from the information sources into two parameters.

- Theme: a feature type appropriate for the purpose of the list.
- Items: combination of feature attribute types to represent a tuple in the list.

A list of these parameters is described as the universe of discourse, which is a result of consensus making by the people involved in the list design.

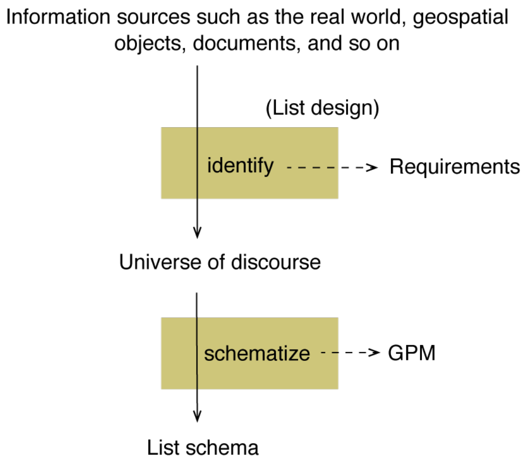


Figure 6.5-1. Conceptual operation model for list design

6.5.3. Schematization

The procedure to schematize the list schema is described below.

i. Select a feature type. Proxy attribute becomes the index keyword for the list
ii. Select attribute types for the creation of the tuple configuration.
iii. Save a list schema in the storage folder.

6.6. Listing

6.6.1. Introduction

Principally, listing is a process to transform virtual features to a list in compliance with the list schema provided through the list schematization (Figure 6.5-2).

6.6.2. List editing

The procedure of a list compilation is as follows.

i. Select geospatial objects for listing.
ii. Select a list schema.

iii. Select a feature type that includes appropriate proxy attribute and other attributes for listing.
iv. Execute automatic compilation.
v. Save the list described with the inner format and/or html file for distribution.

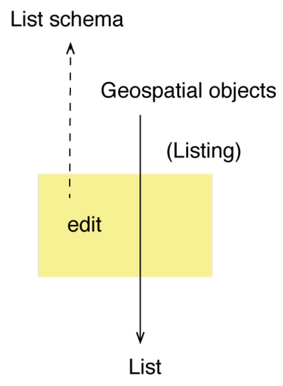


Figure 6.6-2. Function model for listing

7. Conceptual structures for Management

7.1. Introduction

Management in this thesis is a process of dealing with or controlling metadata and geospatial objects. According to the executive order 12906 – Coordinating Geographic Data Acquisition and Access: The National Spatial Data Infrastructure (U.S.Government 1994), National Spatial Data Infrastructure (NSDI) is defined as "the technology, policies, standards, and human resources necessary to acquire, process, store, distribute, and improve utilization of geospatial data." Metadata standard is one of the key rules to realize the SDI. In fact, the ISO metadata standard and its profiles are widely used in international, regional, national and local SDIs in the world (Moellering, Aalders, and Crane 2005). In Japan, GSI has published the Japan Metadata Profile (JMP) for using to register metadata in the national geospatial data clearinghouse (Akeno, Okuyama, and Takazawa 2005). The potential of recent developments in GIT cannot be fully realized without the creation of some form of SDIs (Masser 2008.).

According to the USA National Research Council, geo-library is defined as,

> *A digital library filled with geo-information – information associated with a distinct area or footprint on the Earth's surface – and for which the primary search mechanism is place (NRC 1999).*

Therefore, SDI is implemented as a geo-library used to find and retrieve the appropriate geospatial objects in response to the request by the users.

Although the content of the metadata varies depending on the type of the corresponding resource, it is possible to create metadata in accordance with a metadata schema, which is an instance of the same metamodel. In this chapter, we will discuss the metamodel for metadata schema that is called the General Metadata Model (GMM), then describe the metadata schema modeling based on GMM, and finally propose the conceptual structure of a geo-library.

7.2. General Metadata Model (GMM)

General Metadata Model (GMM) should be defined to model a metadata schema. However, there is no specific description as the UML class diagram of GMM in the ISO 19115-1 – Metadata – Part 1 (ISO/TC 211 2014b). Instead, the text representation as follows appears in the section 6.5.1 of ISO

19115-1.

"Metadata is composed of one or more metadata packages containing one or more metadata classes containing attributes. The relationships between metadata packages and between metadata classes are specified by composition and aggregation relationship symbols. Class attributes and relationships are referred to collectively as metadata elements."

This description is rather obscure as the rules for metadata schema. In addition, according to ISO 19115-1, metadata shall be provided for geographic datasets and may, optionally, be provided for other types of resources. However, it is very difficult to cover all domains by one metadata schema. While we are able to propose the common rules for metadata schemata used in the different domains, just like GFM for application schemata for different geospatial objects. Interoperability between metadata used in the different domain will gain, if domain specific metadata schema follows the same GMM. Therefore, GMM should be independent from local domains.

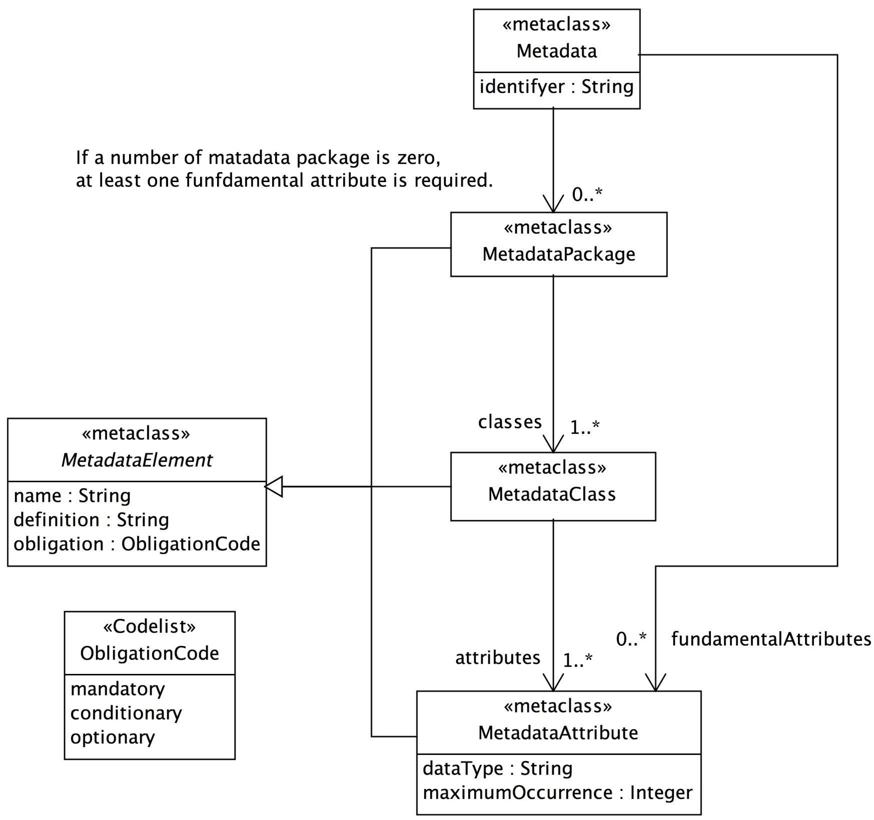


Figure 7.2-1. General Metadata Model formalized by referring to ISO 19115-1

We propose Figure 7.2-1 as GMM by referring to the text at the section 6.5.1 in ISO 19115-1 and the tables that define attributes of metadata classes also described in ISO 19115-1. Metadata schema consists of zero or more than one metadata package. Metadata package is a set of metadata classes gathered with the same theme. Metadata class consists of one or more than one attributes. The unique point of this diagram comparing with ISO 19115-1 is that the direct association from Metadata to metadata attribute. This association enables a design of simple metadata schema.

7.3. Metadata schema modeling

Metadata schema modeling is a function to create a metadata schema. It consists of two sub-functions; identification and schematization.

Identification

The universe of discourse for metadata modeling is identified from information sources by referring to the user requirements for metadata in the target domain.

Schematization

Metadata schema used in gittok is shown in Figure 7.3-2. The number of attributes is limited. However, the beginners may understand the structure and the role of metadata easily.

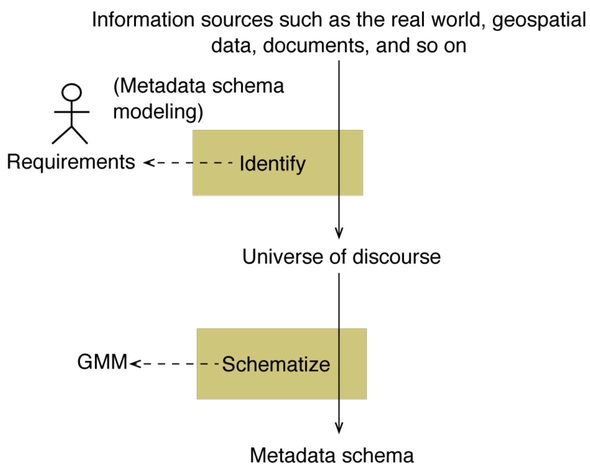


Figure 7.3-1 Metadata schema modeling

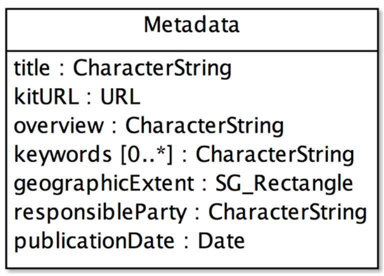


Figure 7.3-2. A metadata schema used in gittok as an implementation in accordance with GMM.

Attributes of metadata, their definition, obligation, data type and maximum occurrence are described in Figure 7.3-1. Keywords is optional element and its maximum occurrence is not limited.

Table 7.3-1 Attribute types in gittok metadata schema

name	definition	obligation	data type	maximum occurrence
title	A name of metadata	mandatory	CharacterString	1
kitURL	Unique resource locator of the kit	mandatory	URL	1
overview	A general review of the kit	optional	CharacterString	1
keywords	A word used in an information retrieval	optional	CharacterString	*
geographic extent	The size of the kit described by rectangle. Corner positions are represented by geodetic coordinates.	mandatory	SG_Rectangle	1
responsible party	The party having obligation to publish the metadata	mandatory	CharacterString	1
publication date	The Gregorian date of publication	mandatory	Date	1

Note: Maximum occurrence * means "more than or equal to zero".

7.4 Geo-library

7.4.1. Introduction

In the case of gittok, the management component called 'Manager' provides capabilities of metadata editing (creation and update), metadata storing in the geo-library, metadata retrieve, and geospatial objects extraction for re-use. In this section, we discuss functions on the geo-library shown in Figure 7.3-3 as the management structure.

7.4.2. Editing

Metadata contains attributes those to be edited automatically and those to be edited manually. For example, the geographic extent can be obtained automatically from the geometric attributes including in the geospatial objects, while a name of responsible party cannot be detected from the geospatial objects in general. Several researchers have challenged the automated metadata editing (Greenberg 2004) (Manso-Callejo, Wachowicz, and Bernabé-Poveda 2010). For example, Olfat, et al. (Olfat et al. 2013) proposed the prototype system for the automatic metadata updating. However, the metadata elements automatically editable are limited, because geospatial data do not always contain enough information for metadata attributes. In the case of gittok metadata, kitURL, geographic extent can be obtained automatically.

7.4.3. Store

In practice, the responsible party of metadata and the custodian of the geo-library should agree the publication of metadata, before storing metadata in the geo-library. In general, if the responsible party asks the entry of metadata, the custodian should confirm the validity of the metadata candidate to avoid declination in quality of the geo-library.

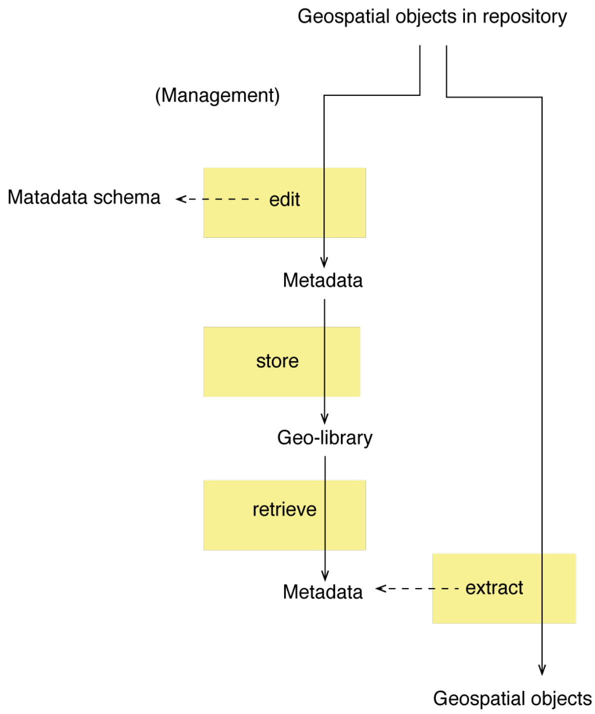


Figure 7.3-3 Management functions in Geo-library

7.4.4. Retrieve

Main functions of geo-library are discovery, use, evaluate, and retrieval of geospatial data by using metadata (Manso-Callejo, Wachowicz, and Bernabé-Poveda 2010). 'Discovery' enables users to find the location of geospatial objects. 'Use' enables users to apply geospatial objects. 'Evaluate' enables users to explore whether geospatial objects fulfil their needs. And 'retrieval' enables users to know how geospatial objects can be extracted.

7.4.5. Extraction

The user can extract the geospatial objects stored in the repository by the user's decision as a result of metadata evaluation. Management component should realize the automatic transferring of objects to the components for acquisition, analysis, exchange and representation, to ease the following works.

8. Software tool for learning about GIT

8.1. Introduction

In this chapter, an implementation of gittok will be introduced. At first, we overview the trend of education regarding GIT, the social demand for education, and we point out that there is a shortage of human resources corresponding to GIT application developments. Next, we introduce the development policy for gittok. Gittok is the assistance tool for learning about GIT based on the concept map of GIT. Its purpose is to understand an overview of the whole image of GIT. In addition, gittok has been developed as a prototype GI system to prove that the concept map of GIT is implementable. Moreover, it should not be a black box. It should be allowed free use of gittok and the source code should be open to public. It should start immediately after downloaded from the site of gittok. And, it should be a desktop application to avoid troubles due to network vulnerability as much as possible.

8.2. Education trend

A comprehensive survey (Okabe et al. 2008) on GI Science education in Japan and USA was carried out as a part of a research initiative project from fiscal year 2005 to 2007. This project was supported by Grant-in-Aid for Scientific Research (A), Ministry of Education, Culture, Sports, Science and Technology (Project number: 17200052. Project leader: Atsuyuki Okabe). Two papers written in English that report the result of survey as a part of this project were also published (Sasaki et al. 2008.) (Kawabata et al. 2010).

In March 2006, Sasaki, et al. (Sasaki et al. 2008) received information on 202 GI-related courses from 39 universities, including both undergraduate and graduate schools throughout Japan. An examination of the results showed that the largest number of courses were focusing on basic concepts, followed by the courses dealing with the collection and processing of spatial data, introduction of GIS applications, cartographic representation, spatial analysis and reading of articles. Both proprietary software such as ESRI's ArcGIS and domestic free software such as MANDARA (http://ktgis.net/mandara/index.php accessed 2018.-01-06) were used for practical training. Many undergraduate education courses take place during the student's third year, and more than 70% of these are one-semester courses.

The research conducted by Mizuki Kawabata et al. (Kawabata et al. 2010) examined GIS education programs at US colleges and universities paying close attention to the degree of multidisciplinary cooperation between geography-related and computer science/information-related disciplines. In this research, 2565 courses at 163 GIS education programs that offered online curriculum information at US colleges and universities during the 2007-2008 academic years were analyzed. As a result, cooperation between geography and information technology disciplines was discovered in only about one-tenth of the GIS education programs. About this issue, Kawabata, et al. discussed as follows.

> *"Given that geospatial technology is evolving into a main stream technology, this situation may change, resulting in computer science/information-related disciplines being more likely to become actively involved in GIS education. Moreover, given that geospatial technology requires a deep understanding of geographic concepts (Gewin, 2004), the computer science/information-related departments may collaborate more actively with geography-related departments."*

So far, about ten years have passed since the surveys were carried out. Since then, useful reference books describing comprehensive knowledge of GIS&T (Kemp 2008) (Kresse and Danko 2012) (Unwin et al. 2012), and papers for the study of GIS&T higher education in USA and Germany (DiBiase et al. 2010) (Schulze, Kanwischer, and Reudenbach 2013) and so on were published. Both DiBiase et al. (2010) and Schulze et al. (2013) introduced the Geospatial Technology Competency Model (GTCM) published by the U.S. Department of Labor, Employment and Training Administration (DoLETA). The GTCM consists of nine tiers. They are,

Tier 1: Personal Effectiveness Competencies
Tier 2: Academic Competencies
Tier 3: Workplace Competencies
Tier 4: Industry-wide Technical Competencies
Tier 5: Industry-sector Technical Competencies
Tier 6: Occupation-Specific Knowledge Competencies
Tier 7: Occupation-Specific Technical Competencies
Tier 8.: Occupation-Specific Requirements
Tier 9: Management Competencies

Tier 2 includes "*knowledge and abilities learned primarily in a school setting*". GIS appears in the geographic skills, and the requirement is "*Use GIS to acquire, manage, display, and analyze data in digital form*". At least, according to this requirement, students learn how to use GIS. While, according to the requirement in Tier 5, workers in the industry sector are expected to have competency of software and application development. And "*Market research indicates that this sector accounts for* *the largest share of sales revenue earned in the geospatial industry"*. However, there is no description about the education for software and application development at least in Tier2. It seems that there is a gap between the competency obtaining by academic education and the professional skill that the industry needs.

Meanwhile, GIS Certification Association, which is an umbrella organization of the GIS Association of Japan (GISA), started the GIS Expert Certification program in 2006. This certification program concepts and structure are similar to the GIS Professional Certification in the USA. Meanwhile in 2013, the Association of Precise Survey and Applied Technology (APA) in Japan started the certification of professionals in GI standards. These efforts have been promoting the professional competency of Japanese engineers. However, university level courses seem to be nearly unchanged from the same status as ten years ago, at least in Japan. As a matter of fact, it is extremely difficult to make a comprehensive curriculum and introduce full-scale education of GIT, as existing departments related to GIS&T have unique characteristics and traditions. Therefore, in this research, we assume replacing the semiannual GIS introductory education with the introduction to GIT, and we consider the introductory education for the beginners who develop GI application in industries.

8.3. Development of gittok

8.3.1. Introduction

Gittok is an object-oriented software referring to the GI standards. There are several GISs as excellent examples of the object-oriented GISs such as GE Smallworld (Longlay et al. 2015) (Chance, Newell, and Theriault 1990). However, they are not exactly in compliance with the GI standards and not the specialized software for learning about GIT.

The following sections discuss the policy for the development of gittok at first, then introduce the software components in gittok. Finally, we confirm that gittok was developed so as to meet the education requirements in terms of functional suitability.

8.3.2. Policy for development of gittok

Methods of GIS education are classified into two categories; One "To teach about GIS" and another "To teach with GIS" (Sui 1995). To teach about GIS concentrates on GIS technology itself. To teach with GIS focuses applications of GIS. Sui wrote as follows.

"Some researchers have argued that as the technical part of GIS has increasingly become the domain of computer sciences, the real challenge in GIS education is to educate students to use GIS to teach geography and conduct geographical research."

Even if this statement is correct, it will not be possible to avoid the education about GIS for students aiming to become GIT researcher and engineer. Moreover, it is difficult to conclude that the research about basic concepts of GIT, for example GFM and GPM, are included in the domain of computer science, as long as they are modeled under the deep understanding of the nature of the real world phenomena and design knowledge of cartographic products.

There are two types of software have been used to teach 'about' and 'with'. The first type is a software for a specific focus on the education. The second type is a general-purpose GIS. Therefore, based on this concept, four patterns of education can be classified (Ota and Plews 2015).

The first pattern centers on teaching about GIS by using specialized software. GISTutor (Raper and Green 1992) was a typical example of this pattern. This very intuitive software was developed using HyperCard and enabled various exercises. The developers introduced concepts of object oriented modeling to the design of the software, as a tool of spatial data modeling. Today, students can learn GIS using the web browser and html (HyperText Markup Language) pages with hyperlinks. One web-based example "The Nature of Geographic Information" is an open and multimedia textbook provided by Dibiase et al. (https://www.e-education.psu.edu/natureofgeoinfo/node/1672). However, they do not have explicit relationship between GI standards.

The second pattern centers on learning about GIS by using general-purpose GIS. Most GIS vendors provide introductory online training courses. Skilled trainers employ a number of training methods to teach about their software products. For example,

- ESRI, http://www.esri.com/training/ accessed 2018-08-03.
- HEXAGON, https://hexagonppm.com/training/virtual-training accessed 2018-08-03.

The third pattern teaches the students of a specific field with specialized software. For example, universities, software vendors and national mapping agencies are offering specialized software targeted at specific groups of users (EDINA, JISC Collections, and Ordnance Survey, http://digimapforschools.edina.ac.uk accessed 2018-08-03) (Yuda 2011).

Similar to the third pattern, the final pattern teaches students of a specific field with general-purpose GIS. This method is convenient for teachers because they do not need to develop the software by themselves and in most cases, they can get teaching resources directly from the software vendors.

The author decided to develop gittok as the software that corresponds the first pattern, which is “teaches about GIT with the specialized software”. Because, there are few GISs adopting the GI standards as it’s architecture, and there are few applications specialized for the introductory course of GIT. The society needs the experts who have skill to develop and maintain the GI applications including general-purpose GIS, SDI and software for specialized purpose referring to the GI standards. “Teaches about GIT” will be possible by the computer science/information-related departments will collaborate more actively with geography-related departments, as Kawabata et al. discussed (Kawabata et al. 2010).

On the other hand, it is almost impossible to learning about GIT in detail within the short period. Therefore, gittok aims that student are able to get the desirable experiences and they will have minds that they wish to obtain further in-depth knowledge related to GIT. However, it means that gittok can be expected to be a breakthrough toward the comprehensive GIT education realized by multidisciplinary collaboration between geography related departments and IT-related departments in the institution of higher education.

To achieve the purpose, gittok should have capabilities enabling the practical exercises to get the fundamental knowledge of the six knowledge areas (Figure 8.3-1). The software components should be harmonized with other components. The schemata implemented in gittok are designed principally in compliance with the conceptual structures and functions introduced in the above chapters. Therefore, architecture of gittok should maintain logical consistency and harmonization between components to fulfill the requirements shown below.

- The modeling component should have functions to schematize application schemata in accordance with GFM. The application schema is transferred to the acquisition component.
- The acquisition component should produce a set of objects (kit) describing not only features but also feature associations in compliance with the application schema. Kits are transferred to the management component and the analysis component.
- The analysis component should enable executions of feature operations and association operations,

those are defined in the application schema as instance methods. The result of analysis is embedded as a derived attribute of feature or association to be able to transfer to other components.

- The management component should have functions to manage kits through their metadata. Metadata is used for search and evaluation of the kit. The kit selected in the management component will be able to use for maintenance, analysis, representation, and exchange.
- The representation component should enable 1) schematization of portrayal schema in accordance with GPM, 2) designing of list schema in accordance with GLM, 3) editing of cartographic objects, 4) conversion from cartographic objects to cartographic product, and 5) distribution of cartographic product to other systems such as web browsers.
- The exchange component should enable encoding and decoding between inner dataset and the system independent data in order to transfer data between different systems.

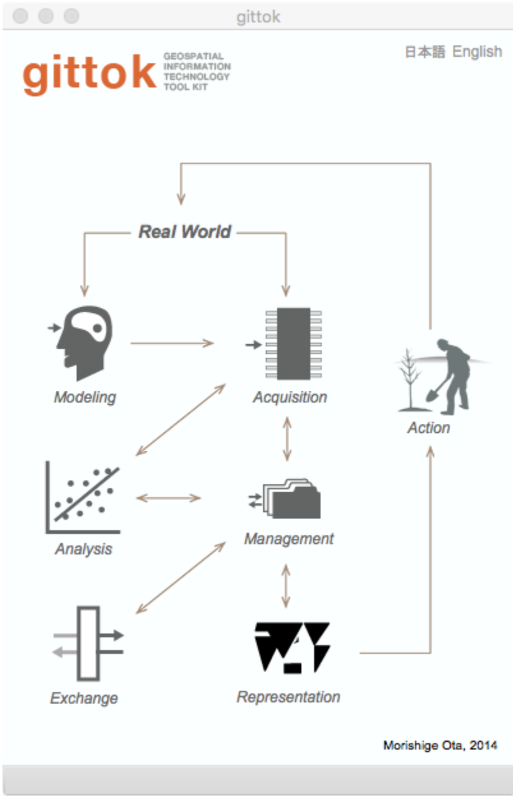


Figure 8.3-1. Title page of gittok

8.3.3. Metamodel for software components

Gittok is an interactive system. Every software component has a graphical user interface (GUI), which roles are the requester sending message from the user to the component, and the presenter showing the response from the component to the user. The component uses objects as a resource, and produces the other objects as a result of processing (Figure 8.3-2). For example, the acquisition component produces a kit (result) by map digitizing (function) based on the base-map and other materials (resource), in accordance with the user demands indicated though the GUI (requester). Then, the user can confirm the result of request on the GUI (presenter).

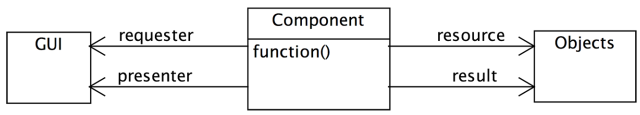


Figure 8.3-2. Meta model of software component illustrated in UML.

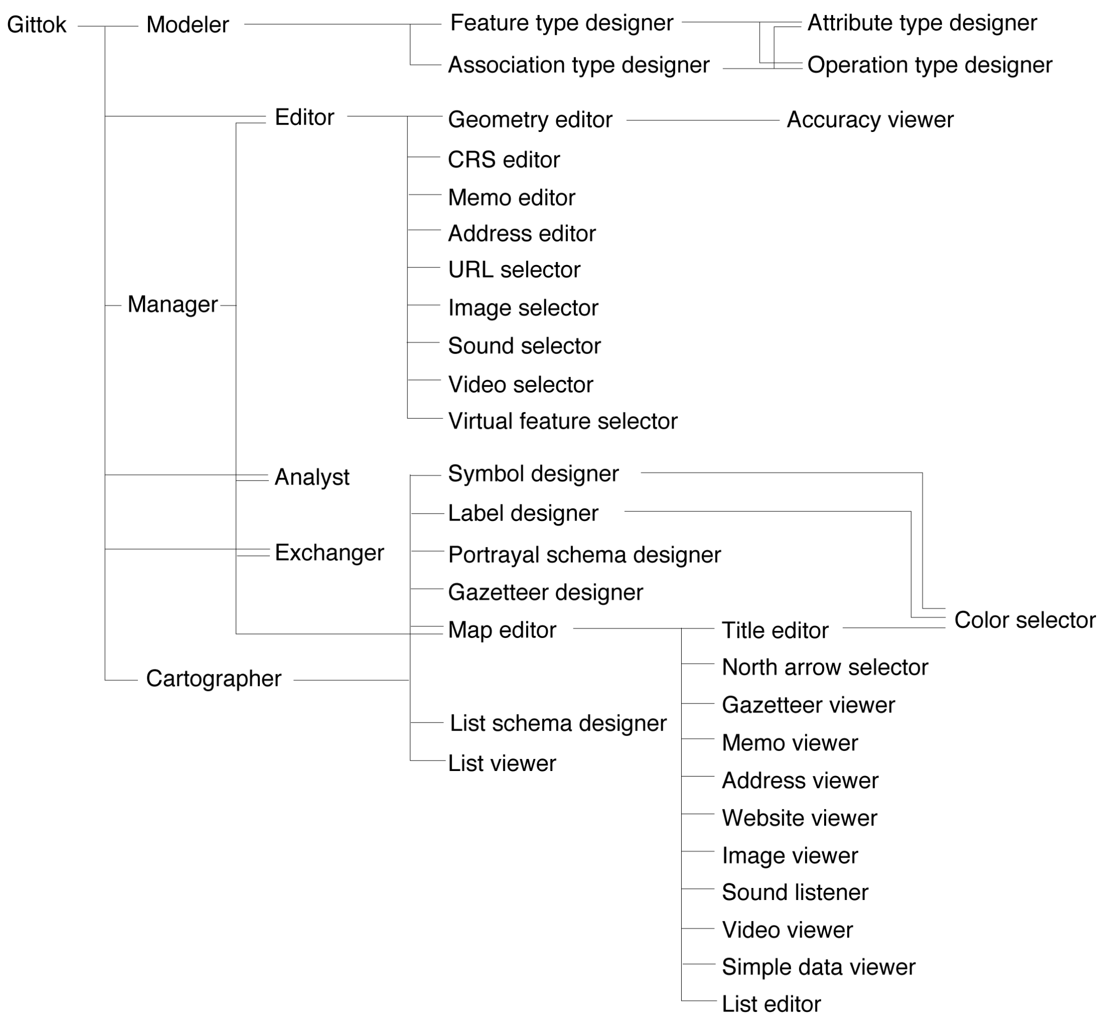


Figure 8.3-3. Hierarchical structure of 40 components in gittok

According to Norman, D.A. (2011),

> *"Modern technology can be complex, but complexity by itself is neither good nor bad: it is confusion that is bad. Forget the complaints against complexity; instead, complain about confusion"* (Norman 2011).

GIT is a complex technology. Therefore, we have systematized the object structures and function structures so as to avoid confusion from complexity by referring to GI standards. Figure 8.3-3 shows the structure of 40 components included in gittok. Every component has GUI called a 'page'. This hierarchical structure is effective to navigate the user what to do next, and it helps the user can understand the conceptual structure of GIT.

8.3.4. User interface policy

In the case of gittok, pages are designed in accordance with the design policy described below, so that the user can experience the sufficient usability. Figure 8.3-4 illustrates these rules.

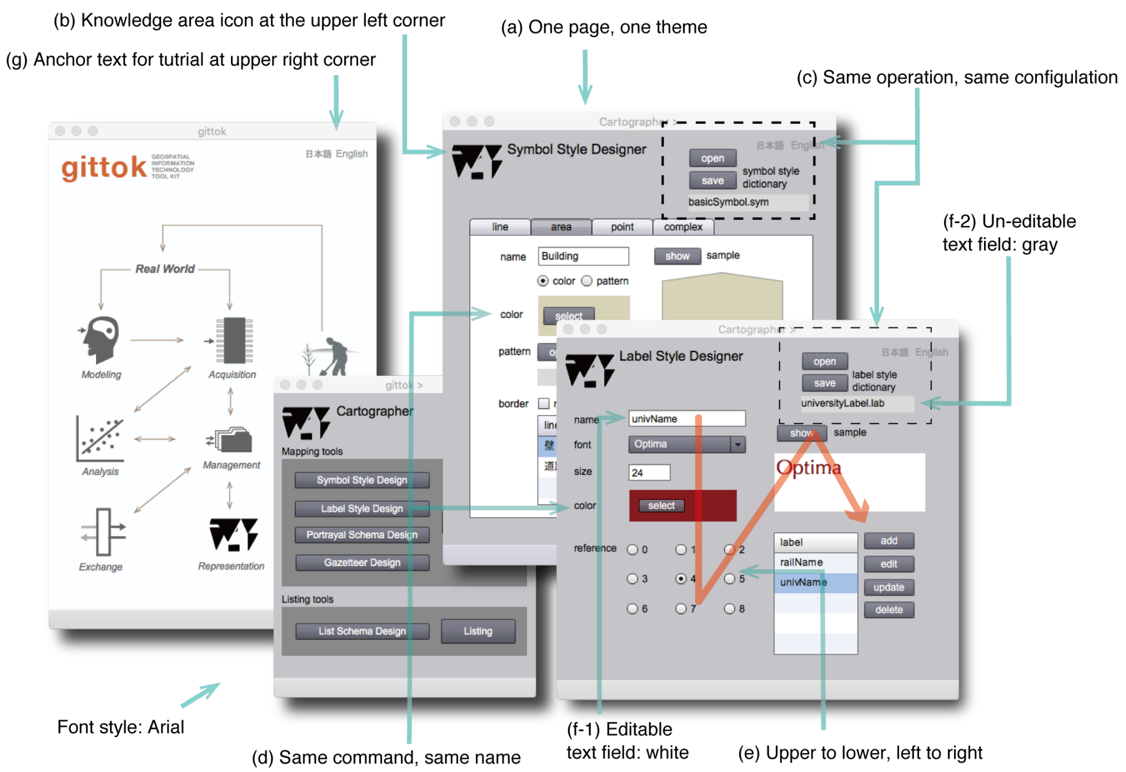


Figure 8.3-4. GUI design policy for gittok

a. One page, one theme.
b. The icon put at the upper left corner of the page shows the knowledge area of GIT.
c. The same configuration of UI components is used for the same type of operation.
d. The same command has the same name. For example, a name of the command for color settings is always 'color'.
e. The operations are ordered so that the user's eyes move from upper to lower and left to right of the page.
f. The text field with white background is editable (f-1), while gray background means un-editable (f-2).
g. The user can refer to the tutorials in Japanese and English by clicking anchor texts provided at the upper right corner. These tutorials are helpful for the self-directed learning.

The components corresponding to the six knowledge areas will be introduced in detail from the next section. They are called Modeler, Editor, Analyst, Manager, Exchanger and Cartographer.

8.4. Modeler

8.4.1. Introduction

Modeler is a software component for schematization of application schema. This is an implementation of the schematization in the modeling introduced at the section 5.2.3. The implementation of Modeler

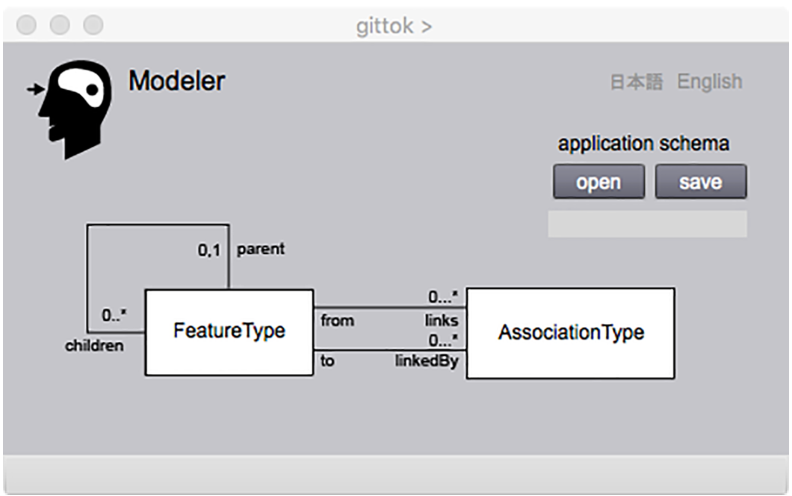


Figure 8.4-1. Modeler GUI. Class boxes work as buttons to open Feature type modeler and Association type modeler.

was one of the major contributions of this thesis. At least by the Internet research, there are few similar educational tools in existence.

Modeler has two sub-components for modeling of feature types and association types. This configuration is the implementation of gittok GFM. These sub-components associate Attribute type modeler and Operation type modeler. The two class boxes shown in Figure 8.4-1 work as the buttons for activating the Feature type modeler and the Association type modeler.

Bolstad noted,

> *"Software developers have had difficulty developing generic tools that may quickly implement object models, so there is an added level of specialized training required."* (Bolstad 2016)

Actually, it is rather difficult to learn modeling in detail at first for beginners, because they need to have knowledge of UML class diagram and GFM as prerequisites. Therefore, the modeling exercise is put after the exercises of data acquisition and analysis, when the students familiarize how to use an application schema. Although, according to the education practices, there is no significant difference of understandability between modeling and other knowledge areas (Figure 10.3-3).

Figure 8.4-2 illustrates the application schema describing the association between road and building. The association instance may have a derived attribute 'distance' between a road and a building as a result of operation. The stereotype <<px>> means it is a proxy attribute. Such an application schema

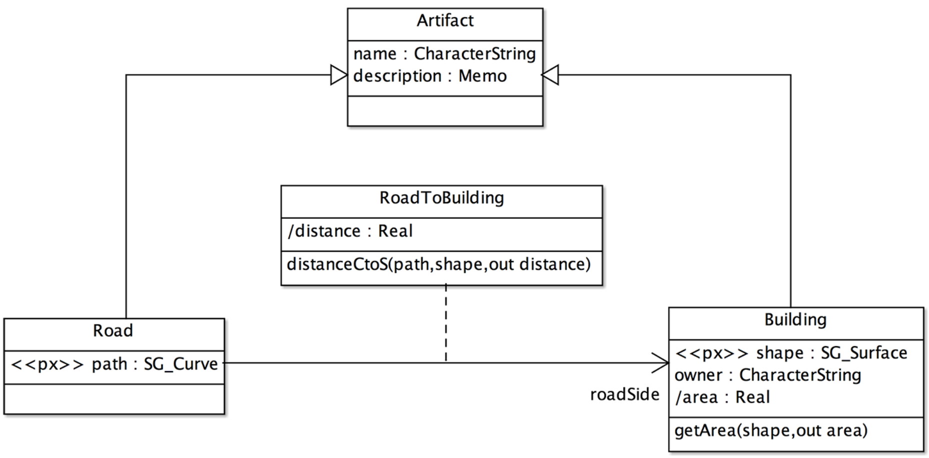
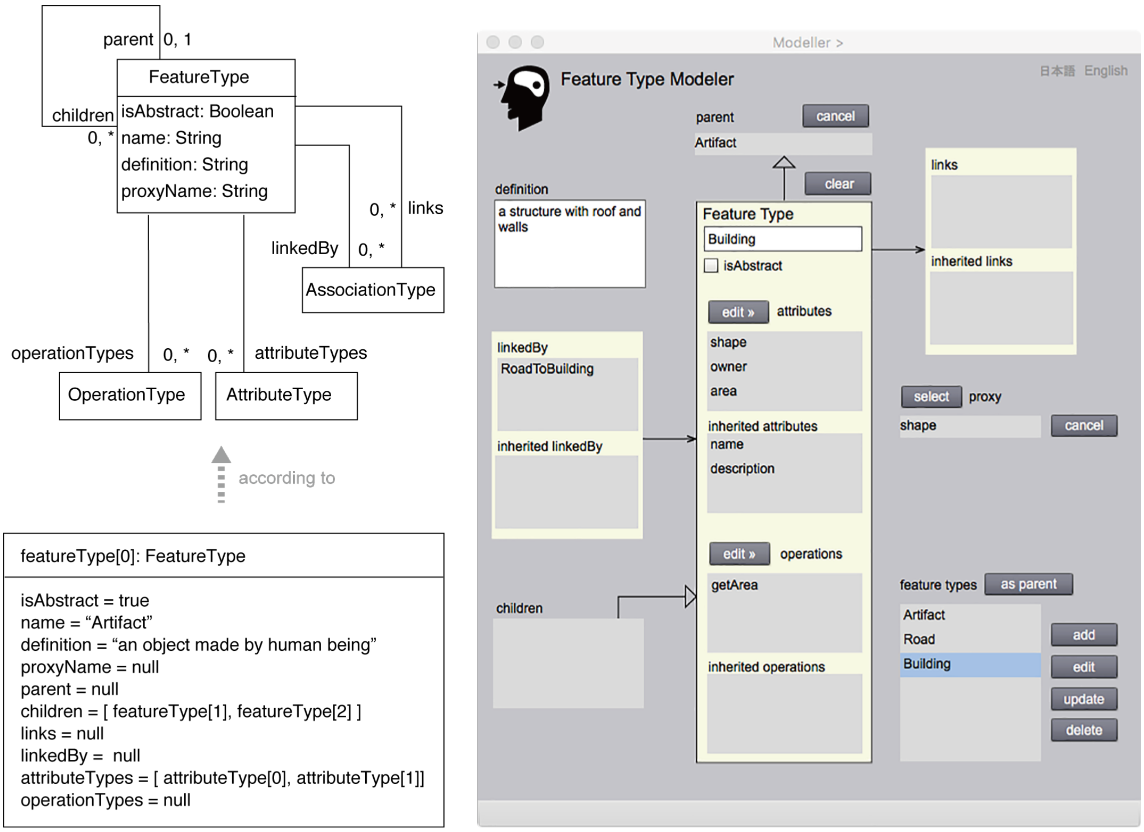


Figure 8.4-2. Application schema to describe association between road and building.

will be able, for example, to evaluate the road side environment. How to model this schema in gittok will be introduced in the following sections as an example of modeling.



featureType[0]: FeatureType

isAbstract = true
name = “Artifact”
definition = “an object made by human being”
proxyName = null
parent = null
children = [featureType[1], featureType[2]]
links = null
linkedBy = null
attributeTypes = [attributeType[0], attributeType[1]]
operationTypes = null

featureType[1]: FeatureType

isAbstract = false
name = “Road”
definition = “long and narrow space pathing from one place to another”
proxyName = “path”
parent = [featureType[0]]
children = null
links = null
linkedBy = []
attributeTypes = [attributeType[0]]
operationTypes = null

featureType[2]: FeatureType

isAbstract = false
name = “Building”
definition = “a structure with roof and walls”
proxyName = “shape”
parent = [featureType[0]]
children = null
links = null
linkedBy = []
attributeTypes = [attributeType[0], attributeType[1], attributeType[2]]
operationTypes = [operationType[0]]

Figure 8.4-3. GUI for Feature type modeling. Object diagrams shown above are obtained by the modeling of feature types as instances of FeatureType.

8.4.2. Feature type modeler

Figure 8.4-3 shows the schematization of the feature types in accordance with the metamodel of feature type (FeatureType) defined in GFM. The user can define the feature type as an instance of FeatureType. In Figure 8.4-3, three feature types are described as object (or instance) diagrams. 'Artifact' as an abstract type is featureType[0]. It is a parent of the concrete types 'Road' (featureType[1]) and 'Building' (featureType[2]).

The GUI shown at the right upper side of Figure 8.4-3 illustrates the UML class for the definition of Building. The attribute types (shape, owner and area), the attribute types inherited from Artifact (name and description), and the operation type (getArea) are also shown. 'Shape' is selected as a proxy attribute of Building. Three object diagrams as instances of FeatureType defined in GFM are shown at the left bottom of Figure 8.4-3. They show the structure of 'Artifact', 'Road', and 'Building'. The GUI and the object diagram of 'Building' show the same content. By the way, the scope of the feature type array is global, but the scope of the attribute type and operation type array are local in the class.

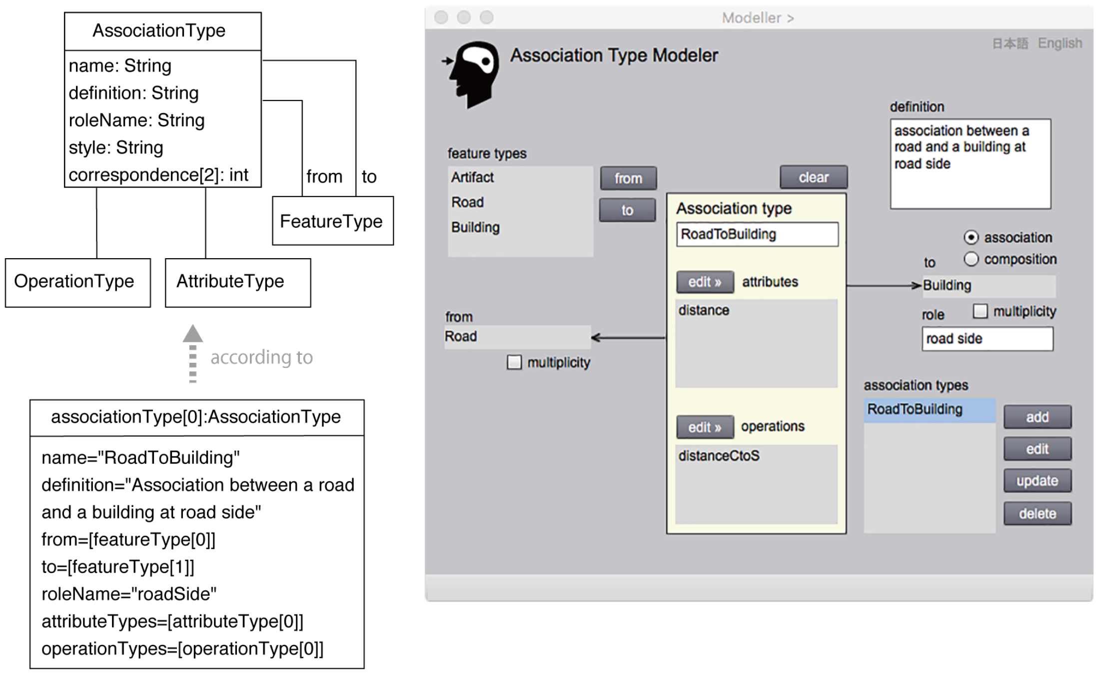


Figure 8.4-4. GUI for association type modeling. Object diagram shown above are obtained by the modeling of association type defined according to AssociationType.

8.4.3. Association type modeler

Figure 8.4-4 shows an example of the schematization process from GFM to the association type. The defined association type is shown at right side. In the case of gittok, two-way navigation is not allowed. Instead, the user can declare opposite association. Meanwhile, an inheritance of association type is not allowed in gittok to keep the structure of application schema simple. Therefore, there is no commands to declare a parent of association type in GUI.

Figure 8.4-4 shows the association type RoadToBuilding. Correspondence between Road (featureType[1]) and Building (featureType[2]) is one to one. It is confirmed by the fact that the multiplicity check boxes are not checked. RoadToBuilding has a derived attribute '/distance' as attributeType[0], which will be a return value of the operation 'distanceCtoS' as operationType[0]. We will discuss the declaration of operation type in the section 8.4.5.

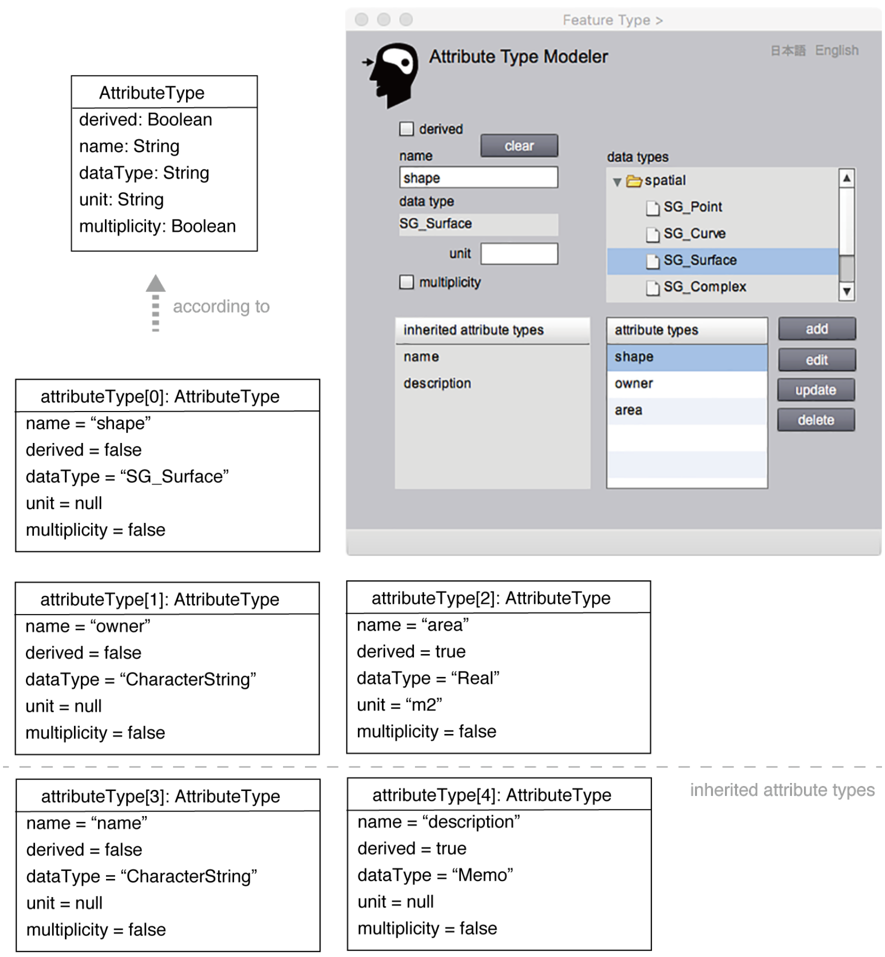


Figure 8.4-5. GUI for modeling of attribute types in Building.

8.4.4. Attribute type modeler

Attribute type modeler is activated by clicking ‘edit » attributes’ buttons on the Feature type modeler and Association type modeler. The user can define attribute name, attribute data type, multiplicity, derived or not, and measuring unit as optional. The GUI in Figure 8.4-5 shows ‘shape’ as an attribute of Building, which data type is SG_Surface with no measuring unit. ‘Shape’ takes single value as it is indicated by the multiplicity check box is not checked. Other attribute types ‘owner’, ‘area’, and attribute types ‘name’ and ‘description’ inherited from Artifact are also declared in Figure 8.4-5.

8.4.5. Operation type modeler

Operation type modeler is activated by clicking ‘edit » operations’ button on the Feature type modeler or Association type modeler. The user corresponds the arguments required by the operation prepared in gittok, with the attributes of the feature type and/or association type related to the operation. Then,

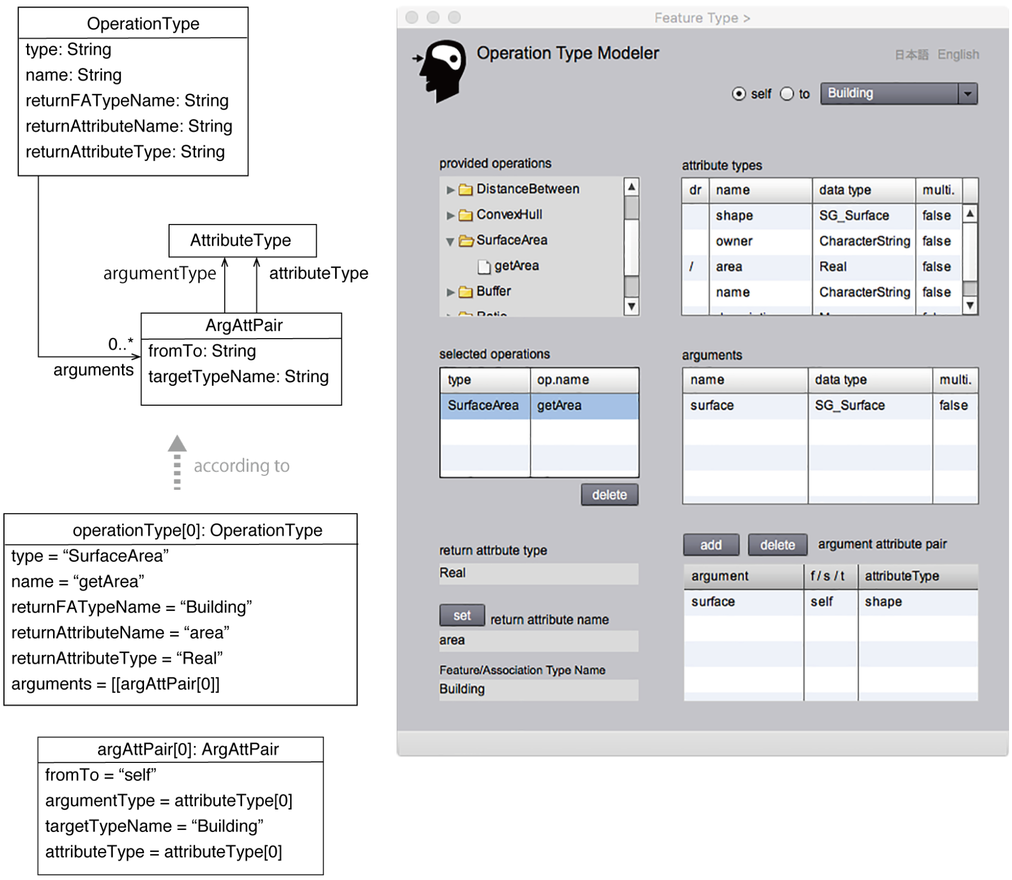


Figure 8.4-6. GUI for operation type modeling.

the user designates the derived attribute for saving the return value. For example, Figure 8.4-6 shows that 'SurfaceArea::getArea' is selected as an operation for Building. The user selects 'shape:SG_Surface' as an argument that the operation requests to connect with 'surface:SG_Surface'. Then, the user selects '/area:Real' in Building for the return value type of the operation.

8.5. Editor

8.5.1. Introduction

Editor is a software component for transformation from real world phenomena and other resources into geospatial objects in accordance with the application schema. The functions in Editor include geometry digitization and multimedia data compilation. Students can learn how to capture geospatial objects in accordance with the application schema. Students can experience the fundamental editing sequence as follows.

i. Map registration,
ii. Quality evaluation for the registration,
iii. Geometric attributes digitization,
iv. Compilation of various kinds of properties, such as geometry, image, audio, video, text, and location identifier to create and update objects.
v. Association editing by the connection between from and to features and by putting attributes of the association.

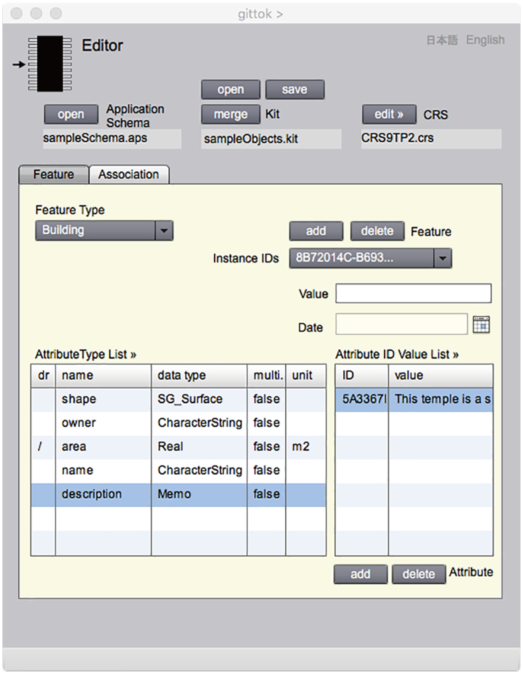
Editor consists of 10 sub-components. They are classified into four groups: geometry, thematic, place, and list. In this section, the main component of Editor is introduced at first, then the geometry editor, the thematic editors, the place editors, and the virtual feature selector will be introduced in the following sections.

8.5.2. Editors for capturing feature and association

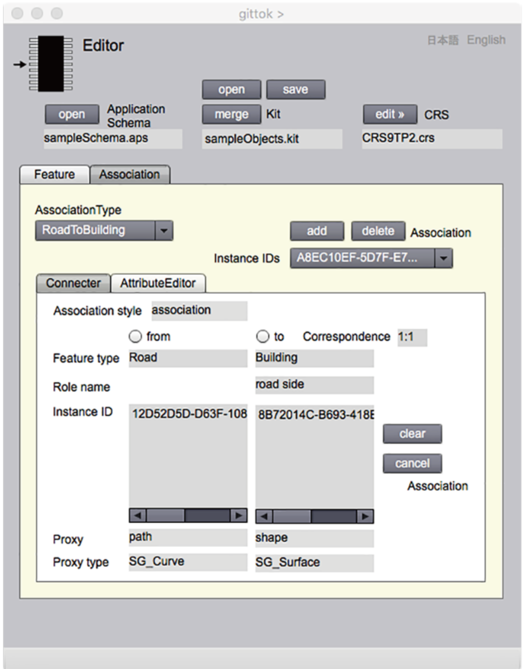
The user can edit instance of the feature type or association type by selecting the tab in Editor (Figure 8.5-1).

To create a feature instance, the following operations are required (Figure 8.5-1, (a)).

i. Create an empty feature instance by clicking 'add Feature' button.
ii. Select attribute type in the AttributeTypeList.



(a) Feature Tab for feature editing



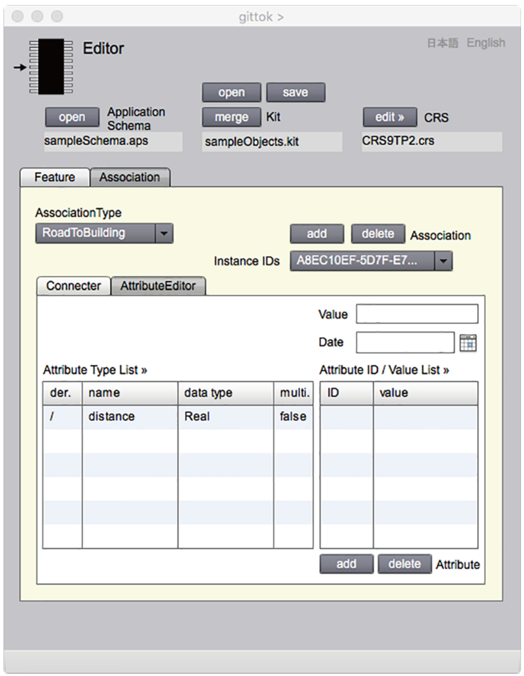
(b) Association: Connecter Tab for association editing

feature[0]: Road
path = [curve[0]] name = “Kagurazaka street” description = [memo[0]]

feature[1]: Building
shape = [surface[0]] owner = “Zenkoku-Ji Foundation” /area = null name = “Zenkoku-Ji” description = [memo[1]]

association[0]: RoadToBuilding
/distance = null from = [feature[0]] to = [feature[1]]

(c) Instance diagram for features and association



(d) Association: AttributeEditor Tab for association attribute editing.

Figure 8.5-1. Editor for editing feature instance and association instance.

iii. Create the attribute value by the appropriate method, such as digitizing, key-in, file selection, and so on. For example, the value of 'description' (Figure 8.5-1, (a)) is created by using Memo editor. Attribute capturing will be described in detail at the following sections.
iv. Add the created attribute value in the feature instance by clicking 'add Attribute' button at the lower right side of Editor.
v. Repeat ii to iv until all attributes are added in the feature instance.

To create an association instance, the following operations are required in general (Figure 8.5-1, (b)).

i. Select the association type for editing.
ii. Create an empty association instance by clicking 'add Association' button.
iii. Select Connector tab and declare the from-feature by selecting 'from features' after 'from' radio button is clicked, then the to-feature by selecting 'to features' after 'to' radio button is clicked.
iv. Select AttributeEditor tab and create attribute values as same as the attribute creation of feature instance. However, the value of derived attribute as the result of analysis operation will be obtained later.
v. Repeat ii to iv until all attributes are added in the association instance.

Figure 8.5-1 (c) illustrates the feature instances of Road and Building, and the association instance of RoadToBuilding. Figure 8.5-1 (d) illustrates the attributes creation for the association attributes. The operation is same as the operations from ii to v of the feature instance creation.

8.5.3. Accuracy evaluation page for map registration

Map registration is defined as the calculation of the parameters of mathematical equation for the transformation from map coordinates to projection coordinates. Usually, Affine transformation is used (CC_in_GIS and NCGIA 1990). More than three registration points are required to get parameters by the least square method, since three points provide six values which can be used to solve for six unknown parameters. Let (x, y) is a map coordinate of the registration point, and (u, v) is a projection coordinate of the corresponding point. The equation of Affine Transformation is as follows.

$$u = ax + by + c$$
$$v = dx + ey + f$$

These equations are represented by matrix notation as follows.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

If the number of registration points is n, the equation will be represented as follows.

$$\begin{bmatrix} u_1 & \cdots & u_n \\ v_1 & \cdots & v_n \\ 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \\ 1 & \cdots & 1 \end{bmatrix}$$

Let these matrices are represented as U, A, and X. Then, the matrix A can be solved as shown below.

$$U = AX$$
$$UX^T = AXX^T$$
$$(UX^T)(XX^T)^{-1} = A(XX^T)(XX^T)^{-1}$$
$$\therefore A = (UX^T)(XX^T)^{-1}$$

By the way, the existence of inverse matrix is confirmed by the determinant of XX^T is non zero.

Today, the web services to provide local plane coordinates at any locations can be found in various countries. For example, Geospatial Information Authority of Japan provides the web site for the conversion from the position on the map displayed on the screen to a plane rectangular coordinate, the true bearing, and the scale factor (http://vldb.gsi.go.jp/sokuchi/surveycalc/surveycalc/bl2xyf.html, accessed 2018-10-28). By using such services, the matrix X can be obtained. While, U is obtained by using the Geometry editor in gittok, which allows the user to digitize the map coordinates of the registration points in the process of map registration (Figure 8.5-2).

By the way, the term 'registration' or 'registering' has been used in the field of color printing and photography. According to the Oxford English Dictionary, 'register' has the meaning that the exact correspondence of the position of colour components in a printed positive. In the case of map digitizing, this term has been used , for example, as follows (Aronoff 1989).

"The different data layers are registered to a common coordinate system or to one data layer that is used as a standard."

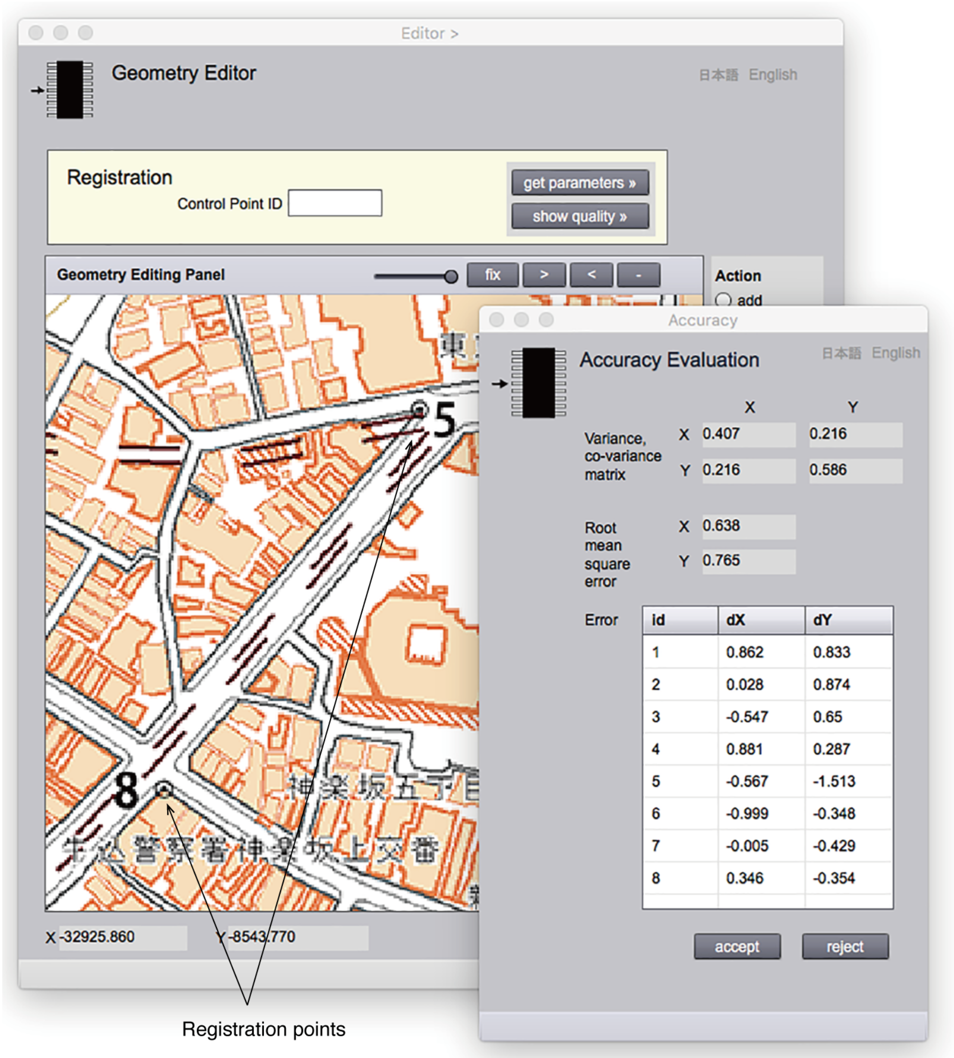


Figure 8.5-2. Map registration and accuracy evaluation. The base map in this case is a copy of GSI Map provided by Geospatial Information Authority of Japan.

We should keep in mind that this type of registration is different from the entry of information in books or lists such as 'touroku' in Japanese. The registration (or register) mark for printings such as Ukiyo-e has been called 'kentou' and registration work has been 'kentou awase' in Japanese (Sakabe 2016) (Musashino Art University, http://zokeifile.musabi.ac.jp/見当/ accessed 2018/11/03). Therefore, we may guess that the term 'registration' was brought for geometric transformation from the vocabulary of multicolor printing when the hand working was replaced by the digital processing.

The quality of map registration is evaluated by errors at the registration points, the root mean square error, and the variance-covariance matrix. These values are displayed in the Accuracy Evaluation page shown in Figure 8.5-2. If the obtained accuracy is enough in accordance with the universe of discourse, the user clicks 'accept' button. However, if the quality is not enough, the user can retry the registration again by clicking 'reject' button.

8.5.4. Geometry editor

Geometry editor is used to capture geometric attributes. Gittok allows to edit two-dimensional geometry such as point, curve, and surface. Digitizing is performed from geometries in the lowest dimension to ease to connect boundaries and co-boundaries. In Figure 8.5-3, coordinates are described in accordance with the plane rectangular coordinate system used in Japan of which x axis is vertical axis toward the north direction and y axis is the horizontal axis toward the east direction. The data type of coordinate values are real. The measuring unit is meter. Editing operations for point, curve and surface are described as follows.

Point

i. Set the radio button 'add' in Action group at the right side of Geometry editor.
ii. Set the radio button 'SG_Point' in Geometry group.
iii. Push the mouse button at the appropriate position.

Curve

i. Set the radio button 'add' in Action group at the right side of Geometry editor.
ii. Set the radio button 'SG_Curve' in Geometry group.
iii. Push the mouse button at the start point after the shape of cursor mark become cross, because the cross mark indicates that the start point was found.
iv. Push the mouse button at the vertices of the curve until near the end point.
v. Push the mouse button at the end point after the shape of cursor mark become the cross.

Curve is a co-boundary of a point, while a point is a boundary of a curve. Therefore, boundary and co-boundary relationships between the curve and its boundaries (start point and end point) are recorded both in the curve and the points.

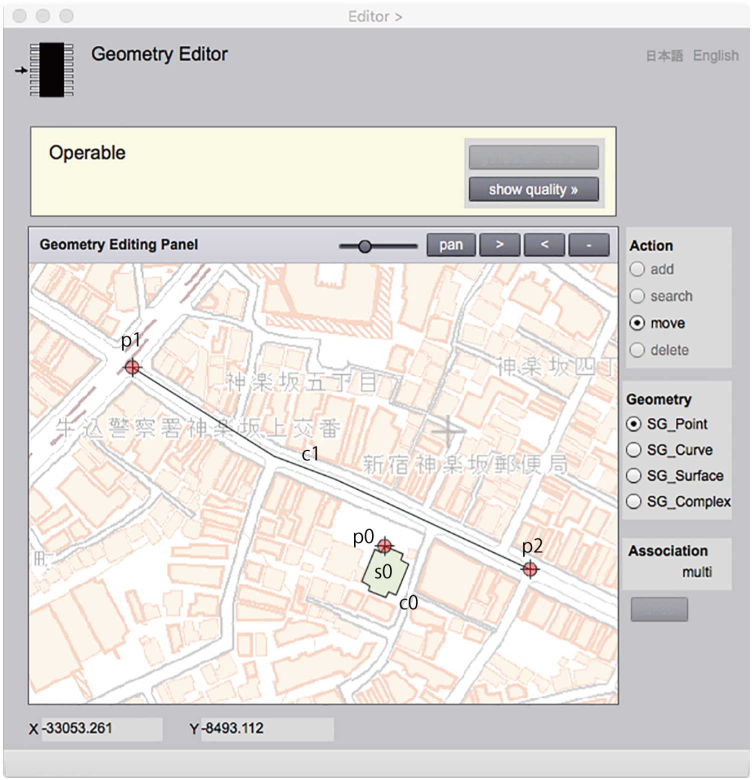
p[0]: SG_Point
position = [-33112.3, -8496.0]
getIn = [c[0]]
goOut = [c[0]]

p[1]: SG_Point
position = [-33034.9, -8606.8]
getIn = null
goOut = [c[1]]

p[2]: SG_Point
position = [-33123.8, -8431.4]
getIn = [c[1]]
goOut = null

c[0]: SG_Curve
start = [p[0]]
end = [p[0]]
shape = [[-33115.7,-8487.8],
[-33118.3,-8488.8],[-33120.2,-8485.4],
[-33133.6,-8490.9],[-33132.2,-8494.1],
[-33134.6,-8495.4],[-33131.7,-8503.1],
[-33129.6,-8502.5],[-33127.5,-8506.0],
[-33113.6,-8500.2],[-33115.1,-8497.5]]
left = [s[0]]
right = null
extend = [oc[0]]

c[1]: SG_Curve
start = [p[3]]
end = [p[4]]
shape = [[-33072.8,-8544.8],
[-33082.7,-8518.3],[-33091.0,-8500.1]]
left = null
right = null
extend = null



s[0]: SG_Surface
exterior = [r[0]]
interior = null

r[0]: SG_Ring
elements = [oc[0]]

oc[0]: SG_OrientableCurve
orientation = false
original = [c[0]]

Figure 8.5-3. Instance diagrams and geometry editor to represent Curve 0 and 1, and Surface 0. Base map is the GSI map provided by Geospatial Information Authority of Japan (https://maps.gsi.go.jp).

Surface

i. Set the radio button ‘add’ in Action group at the right side of Geometry editor.

ii. Set the radio button ‘SG_Surface’ in Geometry group.

iii. Confirm the boundary curves are closed.

iv. Push the mouse button at the inside of the closed curves. A surface is automatically created. If the error, for example, curves are open, is found, warning message is displayed on the page.

Boundary and co-boundary relationships between the surface and the boundary curves are recorded in the surface and the curves simultaneously. Actually, a surface is defined by a ring that is a sequence of identifiers of its boundary orientable-curves. While, each curve has identifiers of co-boundary surfaces at the left-hand side and the right-hand side in accordance with the gittok spatial schema (see Figure 3.2-3).

8.5.5. CRS editor

This section describes the conversion between plane rectangular coordinates and geodetic coordinates. Conversion between map-coordinates and plane rectangular coordinates were already introduced in the section 8.5.3.

Japanese government adopted ‘Japanese Geodetic datum 2011’ and which parameters are,
semiMajorAxis = 6377397.155 m
inverseFlattening = 298.257222101
verticalDatum = “mean sea level of Tokyo bay”

This ellipsoid is originally provided as GRS80 by International Union of Geodesy and Geophysics (IUGG) in 1980. Meanwhile in October 2011, GSI released Japanese Geodetic Datum 2011 (JGD2011) as the new standard datum of Japan, because the wide area crustal deformation happened caused by the ‘2011 off the Pacific coast of Tohoku earthquake’ at March 11, 2011. Parameters of the ellipsoid did not change but the origin point of latitude and longitude moved about 27cm east, and the reference position of height sank 2.4cm. These parameters were used for resetting of triangulation points and benchmarks (GSI, http://www.gsi.go.jp/ENGLISH/page_e30030.html accessed 2018-01-14). However, these changes do not affect the parameters of CRS without its name.

The button to call the CRS editor page is located in Editor. The CRS editor enables to edit the CRS in accordance with the CRS schema, which was already introduced in Figure 3.4-1. The parameters can be edited by using the CRS editor shown in Figure 8.5-4. The coding rule of CRS id is provided by the Japanese Industrial Standard JIS X 7115:2013 Amendment 1 – Metadata Annex 2 (Normative) - The notation of Japanese coordinate reference system (http://www.kikakurui.com/x7/X7115-2013-02.html accessed 2018-01-14). By the way, the author was the project leader for this standardization. JIS X 7115 (APA 2005) was the translation of ISO

19115:2003 – Metadata. In this work, Annex 2 and other normative annexes only for Japan were added. For example, the id shown in Figure 8.5-4 means "JDG 2011: Japan Geodetic Datum 2011, TP: Tokyo Pail, 9(X, Y): Plane rectangular coordinate system Zone No. 9, H: Elevation". This standard published in 2005 was amended in 2013 as the name was changed from JGD2000 to JGD2011.

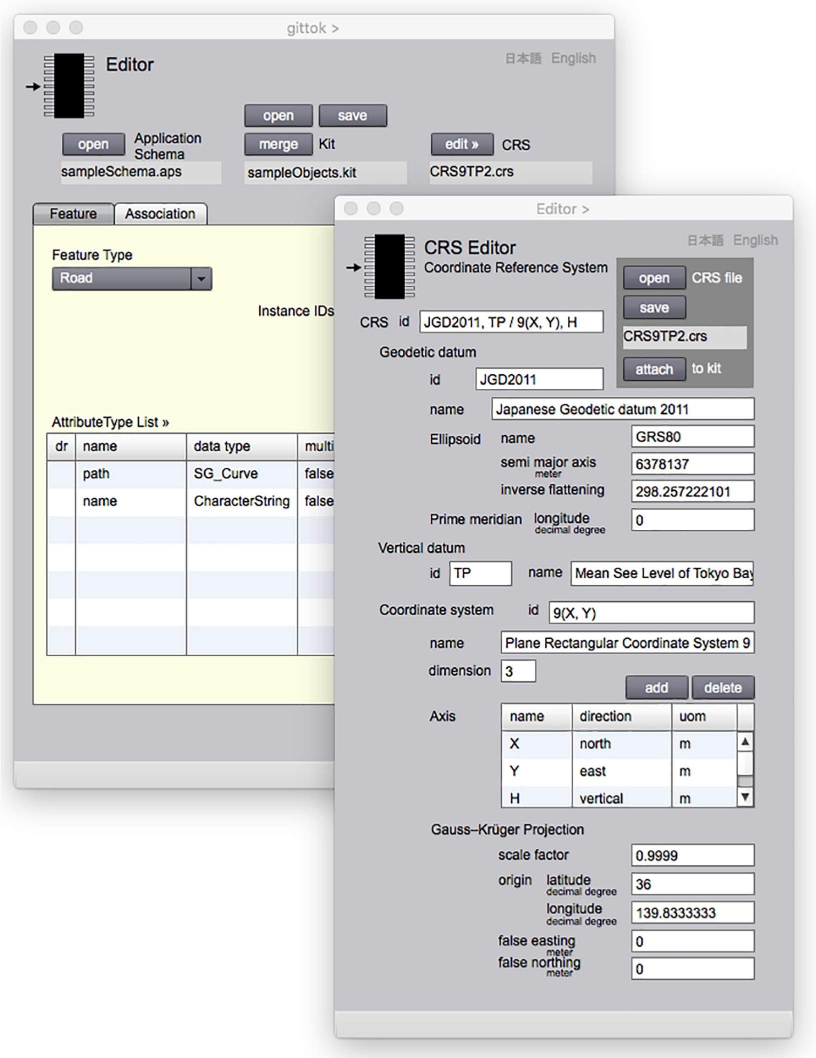


Figure 8.5-4. CRS Editor.

8.5.6. Editors for non-geometric attributes

Gittok allows to use two types of non-geometric attributes; thematic and place. Integer, Real, Bool, CharacterString, and GDate are primitive data types for the description of thematic attributes. Place identifiers are implemented as place attribute types: Address, URL, imageURL, soundURL, and videoURL.

Attributes of primitive data types can be entered in Editor. Attributes of composite data types are compiled with pages provided for particular data types. A feature with more than one attributes of different composite data types can be thought as it is multifaceted. Figure 8.5-5 shows the pages for the compilation of multifaceted features.

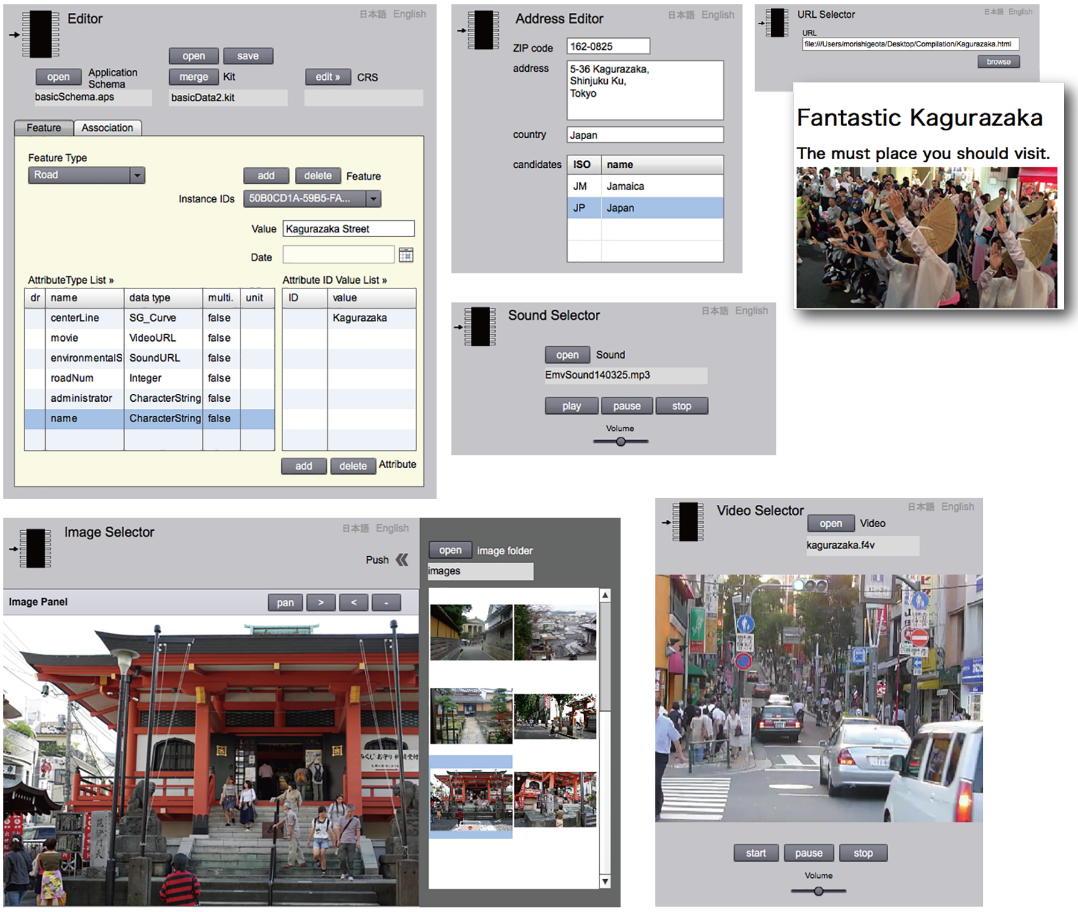


Figure 8.5-5. Attribute editors for a multifaceted feature. The author captured the photographs, sounds and video.

8.5.7. Virtual feature selector

Gittok has a function to create a kit as the mixture of physical and virtual features. Association tab in Editor is used to create an association instance. A physical feature is assigned by a proxy geometry displayed on Geometry editor, while a virtual feature is assigned by its proxy attribute on Virtual feature selector. Two different types of associations, which are not considered in the conventional GISs are introduced in gittok. They are “association between virtual features and different virtual types” and “association between virtual feature and physical features”.

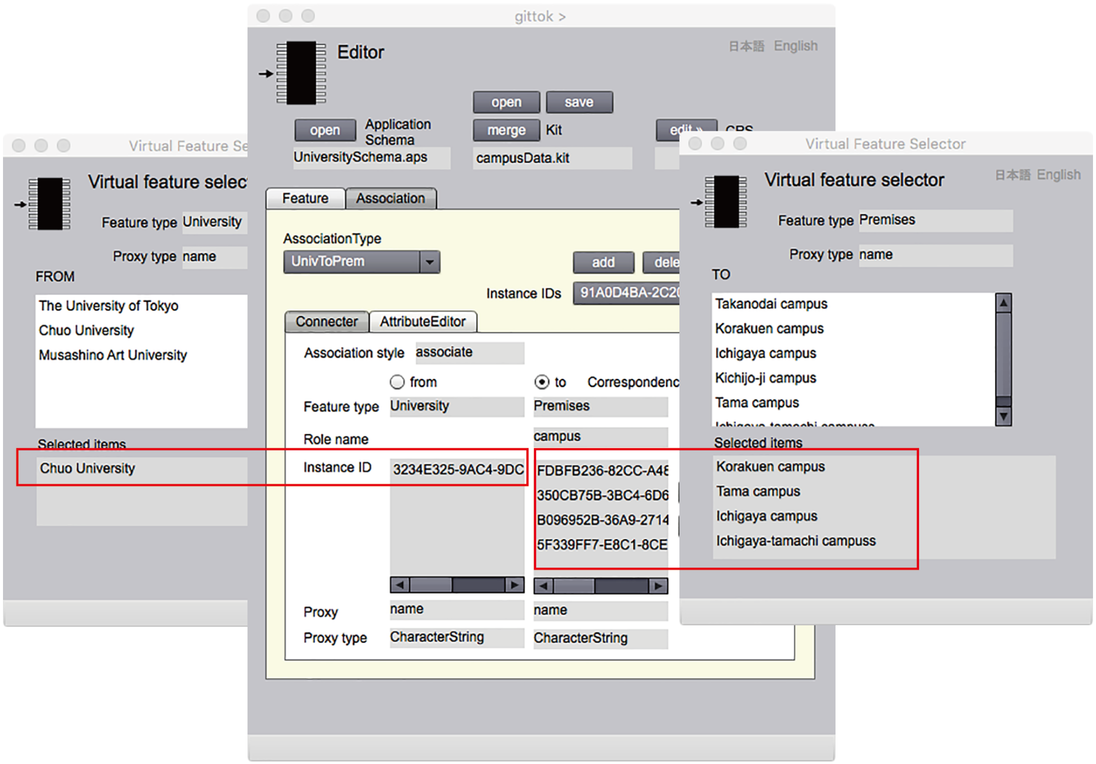


Figure 8.5-6. Association between Chuo University and four campuses assigned on the association tab in the Editor. Features in both side are virtual.

Figure 8.5-6 shows the creation of the instance that represents the association between virtual features of different types: Chuo University and its 4 campuses. The application schema was defined in Figure 2.5-1. Chuo university is selected as a ‘from’ feature in the Virtual feature selector for ‘FROM’ at the left-hand side. Then four premises are selected as ‘to’ features in the Virtual feature selector ‘TO’ at the right-hand side.

Figure 8.5-7 shows the association between virtual feature and physical features; the campus and the buildings in it. By clicking the ‘from’ radio button in the Connector tab, Virtual feature selector is called. In the selector, Korakuen campus is selected as a ‘FROM’ feature. Then buildings in the campus are selected as ‘TO’ features. The association instance between campus (premises) and buildings is defined by these operations.

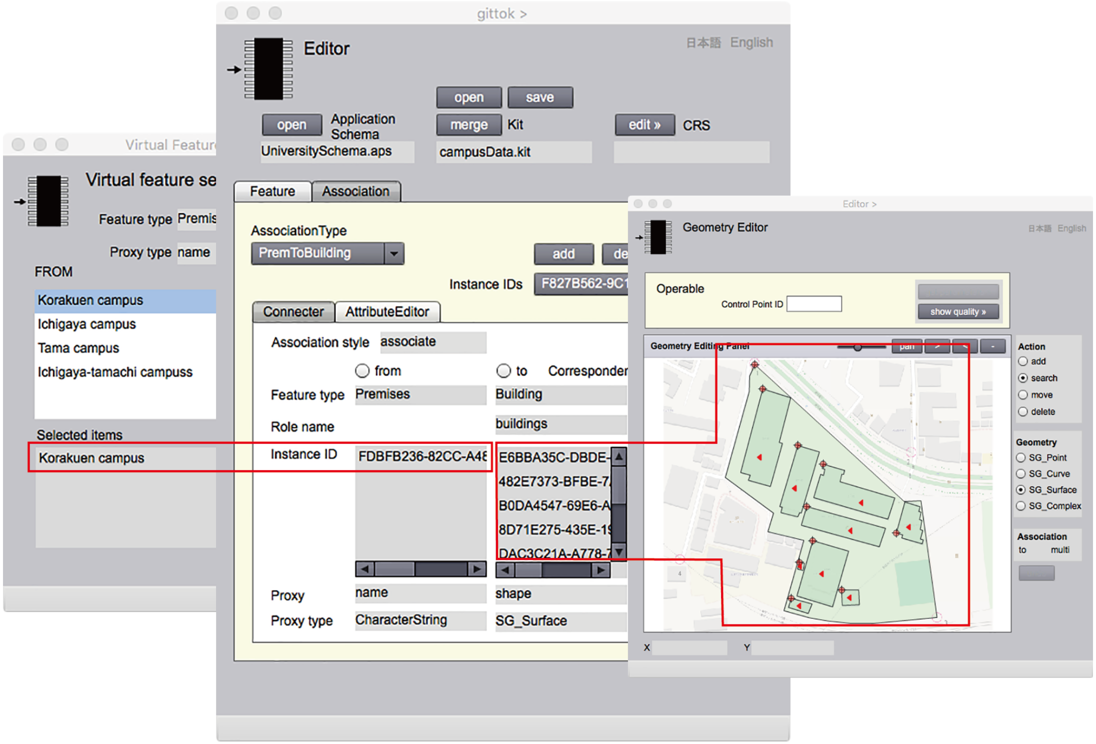


Figure 8.5-7. Association between Korakuen campus and school buildings assigned on the association tab in the Editor. In accordance with the application schema shown in Figure 2.5-1, Data type of a campus is Premises and it is a virtual feature. While Building is a physical feature type.

8.6. Manager

8.6.1. Introduction

Manager consists of the metadata editor and the geo-library. Gittok currently does not use DBMS and the geo-library is not distributed on the Web. However, students can make metadata, and they retrieve and reuse a kit by using the personal geo-library (Figure 8.6-1). To retrieve a kit by a geographic extent, one of the Open Street Map provided by the OpenStreenMap Foundation (http://www.openstreetmap.org accessed 2018-08-06) or the GSI Map provided by the Geographic Information Authority of Japan (https://maps.gsi.go.jp/ accessed 2018-08-06) can be used as the background map.

8.6.2. Metadata creation and editing

The creation procedure of metadata in compliance with the metadata schema is as follows. The geographic extent is automatically calculated by using the coordinates in the kit.

i. Open a kit.

ii. Declare items of metadata.

iii. Save the metadata.

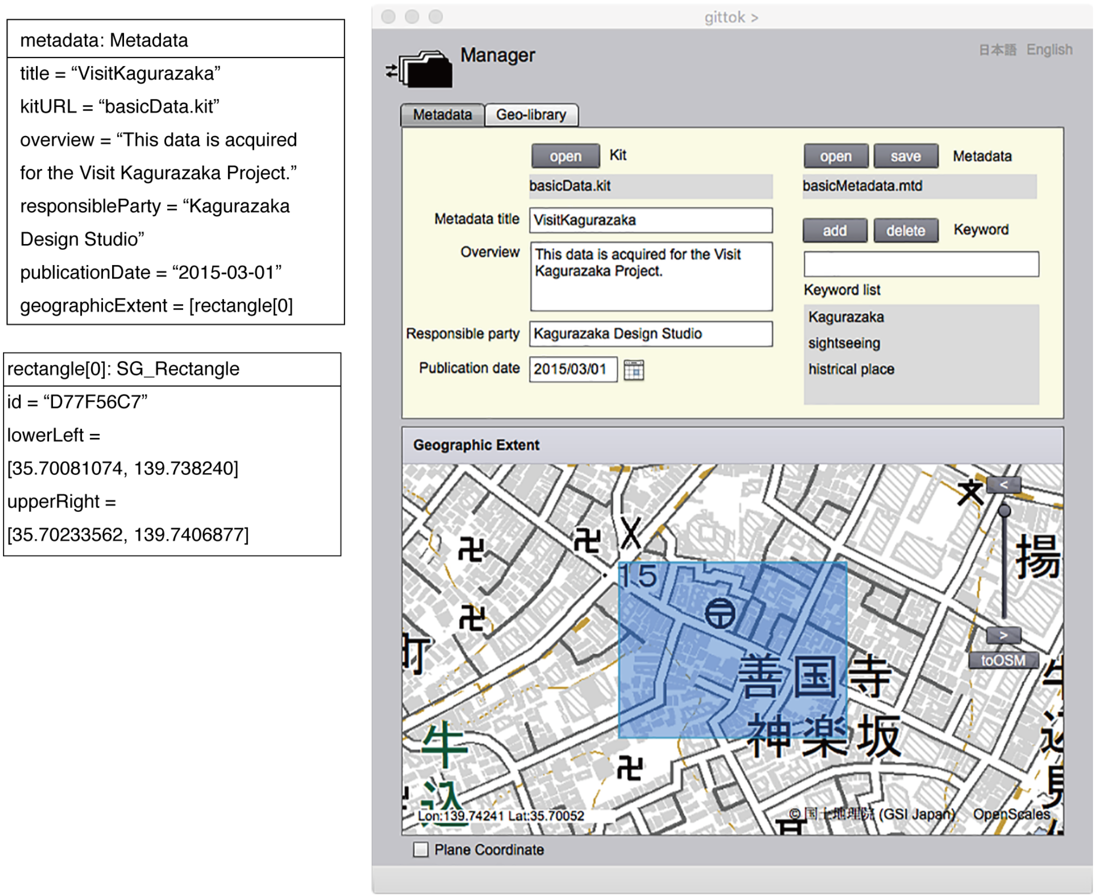


Figure 8.6-1. The GUI for metadata creation and editing, and object diagram for metadata on the GUI. Lowerleft and upperRight are geodetic coordinates represented in decimal degree. The background map is the GSI map.

The procedure of metadata updating is as follows.

i. Open a metadata on GUI.

ii. Change the values of items on GUI.

iii. Save the revised metadata.

8.6.3. Geo-library

Geo-library in gittok is used to retrieve the appropriate metadata (Figure 8.6-2). By clicking 'metadata' edit button, the GUI is changed to Metadata tab, and the content of the retrieved metadata can be confirmed. If it is suitable for the user's purpose, the user can send a kit to the appropriate component by clicking one of the five buttons (metadata editing, kit maintenance, analysis, exchange and mapping) located at the right side of the geo-library GUI.

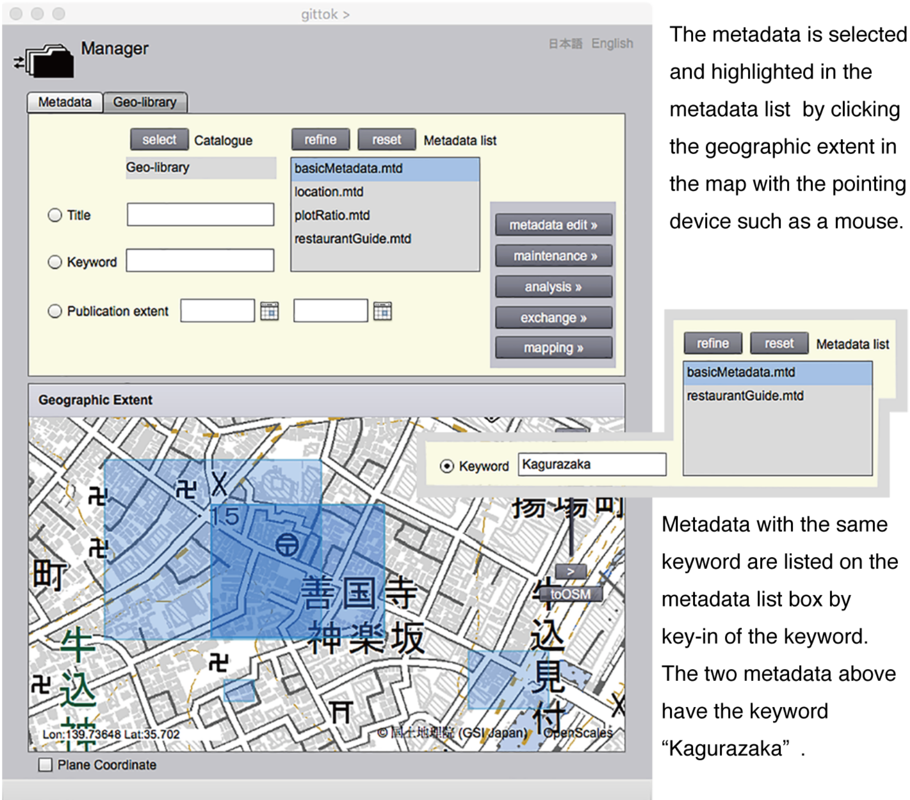


Figure 8.6-2. Metadata retrieval by using the GUI of Geo-library.

8.7. Analyst

8.7.1. Introduction

Analyst is a software component for execution of geospatial analyses. Students may study an object-oriented analysis to obtain meaningful information, through applying operations declared in feature types and association types. However, there are many analytical approaches including spatial analysis, spatial modeling, geostatistics, spatial econometrics, spatial statistics, qualitative analysis, map algebra and network analysis (DiBiase et al. 2006) and so on. Providing and Learning all methods are almost impossible because of the enormous variety. And it is rather not so important to implement the wide range of analyses for the study of the technology to activate the analysis methods as operations. Therefore, analysis tools executed through the Analyst in gittok are limited to software modules applying basic computational geometry such as

- Simple buffering from a curve to geometries,
- Center of geometry
- Convex hull,
- Shortest Euclidian distance between geometries,
- Length of a curve,
- Shortest path finding,
- Estimation of plot ratio (floor area ratio).

Table 8.7-1 describes analysis tools included in gittok in detail.

Table 8.7-1. Analysis tools provided by gittok

Operation types	Operations	Descriptions
Buffer	getOnBufferPoints	Get points of which distances from base curves are less than the given value.
	getOnBufferCurves	Get curves of which minimum distances from base curves are less than the given value.
	getOnBufferSurfaces	Get surfaces of which minimum distances from base curves are less than the given value.
Center	centerOfPoints	Get a gravity center of a set of points
	centerOfCurve	Get a gravity center of a curve
	centerOfSurface	Get a gravity center of a surface
	centerOfComplex	Get a gravity center of a geometry complex
	centerLayingOnCurve	Get a point laying on the curve of which extended distances from start and end points are same.
	centerOfMIC	Get a center point of a maximum inscribed circle in a surface. The center is located always inside of the surface.
	centerOfNetwork	Get a gravity center of a network
	locateFacility	Get a gravity center of points with weight. This operation is used for the exercise of the store siting.
ConvexHull	getComvexHull	Get a convex hull for a set of points
DistanceBetween	distancePtoP	Get a distance between two points
	distancePtoC	Get a minimum distance between a point and a curve
	distanceCtoC	Get a minimum distance between two curves
	distanceCtoS	Get a minimum distance between a curve and a surface

Length	lengthOfCurve	Get an extended length of a curve
Ratio	getPlotRatio	Get a plot ratio of a building on a parcel
ShortestPath	getPathByWarshallFloyd (*)	Get a shortest path from a start point to an end point in a directed graph

(*) The Floyd-Warshall algorithm (Cormen et al. 2009) is also called 'Warshall-Floyd' in Japan (Kubota 2015). This algorithm is valid, as long as there are no negative-weight cycles in the network.

By the way, gittok equips various kinds of basic mathematical functions behind of software components. They are listed in Table 8.7-2. They will be able to re-implement as operations for analysis, if they will be required.

Table 8.7-2. Mathematical functions provided for gittok

Function types	Functions	Descriptions
Affine transformation	getParameters	Get coefficients, errors, RMSE, and variance co-variance matrix by applying the least square method
	conversion	Convert coordinates from map to plane rectangular coordinate on the ground
	inverseConversion	Convert coordinates from plane rectangular coordinate on the ground to map
Angle	decimalToRad	Convert angle from decimal degree to radians
	radToDecimal	Convert angle from radians to decimal drgree
	degToRad	Convert angle from degree of which style is 'ddmmss.ssss' to radians
	radToDeg	Convert angle from radians to degree of which style is 'ddmmss.ssss' to radians
	azimuth	Get azimuth from a coordinate to another coordinate
	traverseAngle	Get traverse angle of two connected vectors of one's end point is the same as the start point of another
Distance	p2p	Get distance between two points
	nearestPfromPs	Get nearest point from a set of points
	p2l	Get distance between point and line segment
	p2ls	Get distance between point to a coordinate array
	p2clb	Get distance between point to a circle boundary
	lsAnls	Get distance between two coordinate arrays
EastWest	isLEastOfP	If a return value is plus, then a line segment is at east side of a point
	is LSEastOfP	If a return value is plus, then a coordinate array is at east side of a point
GaussKruger	setParameters	Set CRS to Gauss-Kruger projection
	BLtoXY	Convert from geodetic coordinate to plane rectangular coordinate
	XYtoBL	Convert from plane rectangular coordinate to geodetic coordinate

Hyperbolic	sinh cosh tanh arctanh	Get hyperbolic sin of a given number Get hyperbolic cos of a given number Get hyperbolic tan of a given number Get hyperbolic arctan of a given number
Length of line string	lengthLS	Get extended length of coordinate array
Matrix	determinant transpose multiple inverse add sub getAffineParams	Get determinant of matrix Get transpose matrix Get multiple of two matrixes Get inverse matrix Get addition of two matrixes Get subtract of two matrixes Get coefficients of linear transformation (the section 8.3.3)
Relation	nearestPonL nearestPonLS nearestVertexIndex intersectToLineString pointInRing	Get the point on a line segment nearest from a point Get the point on a polyline nearest from a point Get the vertex on a polyline nearest from a point Get the intersection points between line segment and polyline Get if a point is inside of a simple closed coordinate array (This is so called point-in-polygon problem)
Sort	lexicoMin sortRecords	Sort 2-d coordinate with lexicographic order (*) Sort records by key column

(*) Lexicographic order in this case is that if more than one same x-values are found after sorting with ascending order, then the sorting of y-values with ascending order will be done in this extent.

8.7.2. Example of Analyst operation

Figure 8.7-1 shows the shortest path finding as an example of spatial analysis. The finding operation is put at RoadNetwork defined as an association between intersections and roads. Because a road network can be defined as a directed bipartite graph from node points to link curves of which boundaries are node points.

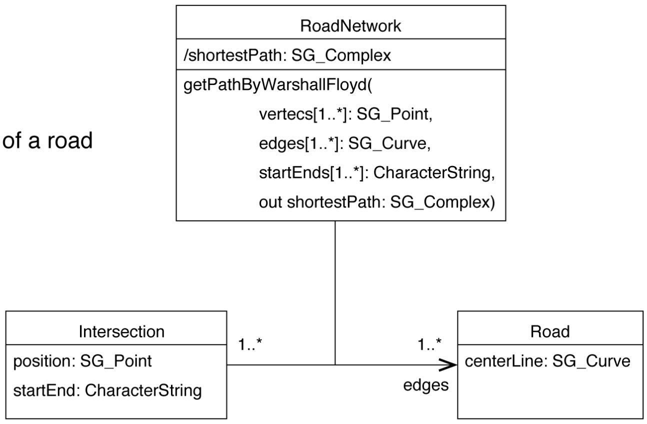
The return value of this analysis is a linear graph that consists of node points and link curves. The spatial schema of gittok enables the description of the datatype called SG_Complex as shown in Figure 3.2-3 in the section 3.2.3. SG_Complex is a geometry complex such as a graph. Therefore, the data type of the return value of this operation is SG_Complex as a chain of curves.

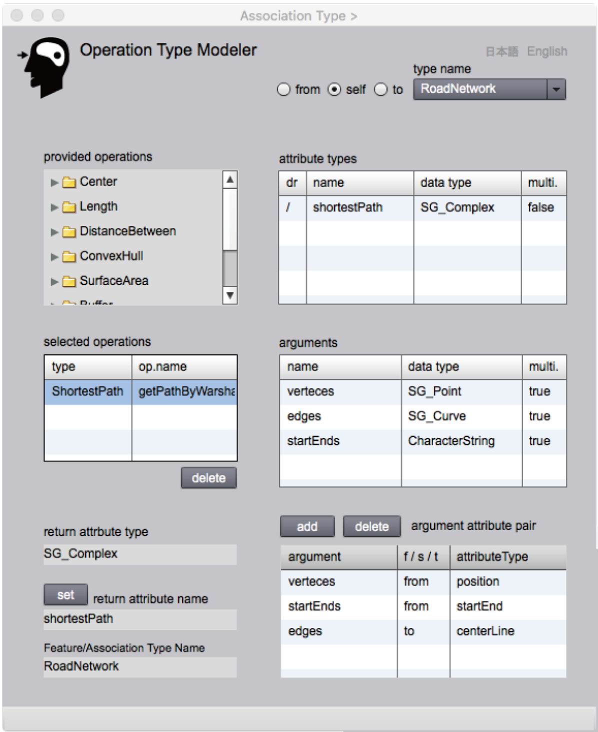
The elements of the operation syntax are described as follows. The examples are taken from Figure 8.7-1 (b).

i. Operation type name (e.g., ShortestPath)

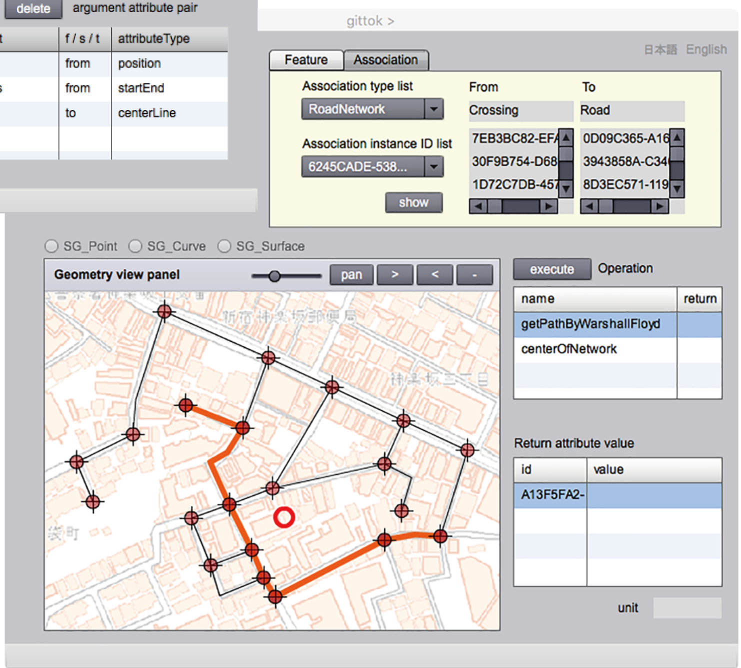
ii. Operation name (e.g., getPathByWarshallFloyd)

(a) Application schema of a road





(b) Definition of the operation to get a shortest path by the Warshall-Floyd algorithm.



(c) Analyst GUI that shows the shortest path as an output of the operation. Single circle is a gravity center of the shortest path to indicate where it is.

Figure 8.7-1. Example of analyst operation

iii. Array of argument and attribute combinations, which element is the argument name of an operation and the attribute name of a feature/association. An argument and an attribute may be multiple. (e.g., {{vertices, position, “multi”}, {edges, centerline, “multi”}, {startEnds, startEnd, “multi”}})
iv. The name of the return attribute type and its type name. (e.g., return attribute name: shortestPath, feature/association type name: RoadNetwork)

When the return value is stored in the attribute ‘shortestPath’, Analyst shows the identifier (id) of the attribute in the list of ‘Return attribute value’ in the GUI as shown in Figure 8.7-1 (c). The user can confirm the shortest path on the Geometry view panel by clicking the return attribute id.

8.8. Cartographer

8.8.1. Introduction

Cartographer is provided for designing of cartographic products such as a map, a gazetteer and a list. It consists of two component groups: ‘Mapping tools’ and ‘Listing tools’ (Figure 8.8-1). Mapping tools enable to create maps and gazetteers. The Map editor is used to compile maps. Map editor also allows that the user can create cartographic products following the internal format, SVG, and GeoJSON for the distribution of geospatial information.

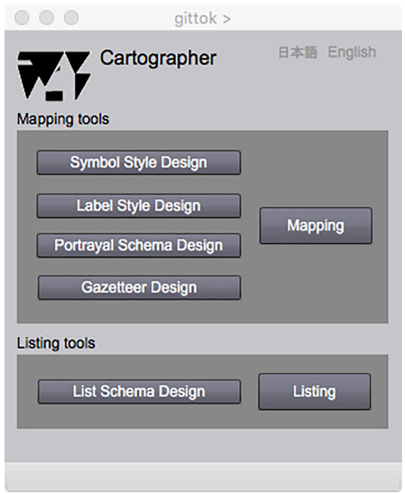


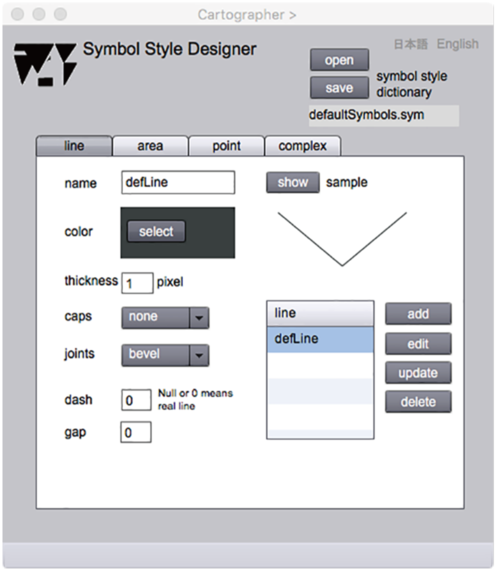
Figure 8.8-1. Cartographer GUI

Meanwhile, listing tools are prepared for listing of virtual features. Listing tools consists of the List schema designer and the List editor. The List schema designer is used to create a list schema. The List editor provides the lists following the list schema. The lists follow the internal format defined in gittok, or HTML for distribution to the public on the Web. In this section, the listing tools will be introduced after the introduction of the mapping tools.

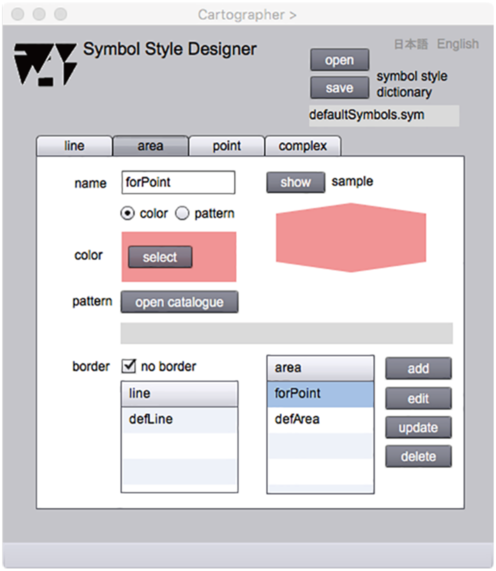
8.8.2. Mapping tools

Introduction

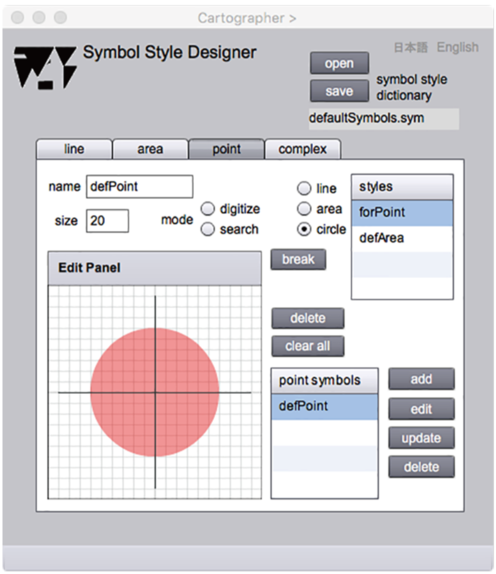
Mapping tools consist of five components: Symbol style designer, Label style designer, Portrayal schema designer, Gazetteer designer and Map editor. The Symbol style designer enables to design symbol styles such as point, line, area, and complex symbol styles. The Label style designer enables to design label styles for labeling of a feature displayed on a map. Gazetteer designer enables to compile a gazetteer for searching of a feature on a map.



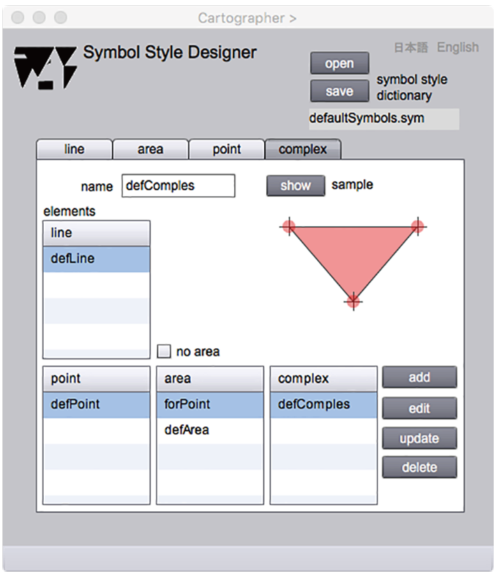
(a) line symbol style



(b) area symbol



(c) point symbol style



(d) complex symbol style

Figure 8.8-2. GUIs in Symbol style designer.

Map editor enables to compile a map in accordance with the portrayal schema, and it can encode the map to the system independent format for the distribution on the web. These tools are used to produce a general-purpose map, a choropleth map and an interactive map.

Symbol style designer

Symbol styles are designed as an element of a symbol style dictionary (SSD) in accordance with the schema for SSD, which was introduced in the section 4.2.3. Figure 8.8-2 shows the GUIs in the Symbol style designer. Four tabs for line, area, point and complex symbol styles are prepared. The order of design operation is basically line at first, then area, point and finally complex. According to the schema for SSD, line symbol is the primary symbol. A boundary of area symbol is a line symbol. Point symbol is basically represented by a coordinate sequence modified by a line symbol, and point symbol may be represented by a combination of line symbols and area symbols. Complex symbol is a combination of a point symbol, a line symbol and an area symbol. The schema for SSD was illustrated in Figure 4.2-3.

Color selector

Colors for a line symbol and an area symbol is selected as RGB and alpha value by the Color selector shown in Figure 8.8-3.

Figure 8.8-3. Color selector

Label style designer

Label style is designed in accordance with the LabelStyle class implemented in the Label style designer (Figure 8.8-4). Label style name, font type, size, color, and reference position can be declared. Font type can be selected from the dropdown menu. It shows the font types those user's machine provides. The user can confirm the design by clicking 'show sample' button. The font type name as a sample of the label appears in the text box under the button. The LSD is saved in the storage folder after designing.

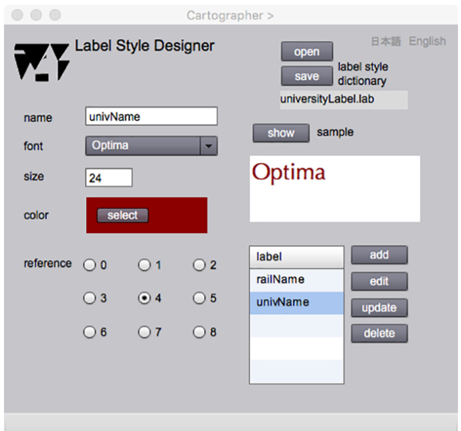


Figure 8.8-4. Label style designer

Portrayal schema designer

Portrayal schema is created in compliance with GPM introduced in the section 4.2.2. GUI of the Portrayal schema designer consists of three blocks (Figure 8.8-5).

The upper block enables the selection of the application schema, SSD and LSD. This block also is used for saving and opening of the portrayal schema. The middle block enables the selections of feature/association types, attributes for representation, and, if it is required, attribute types for the

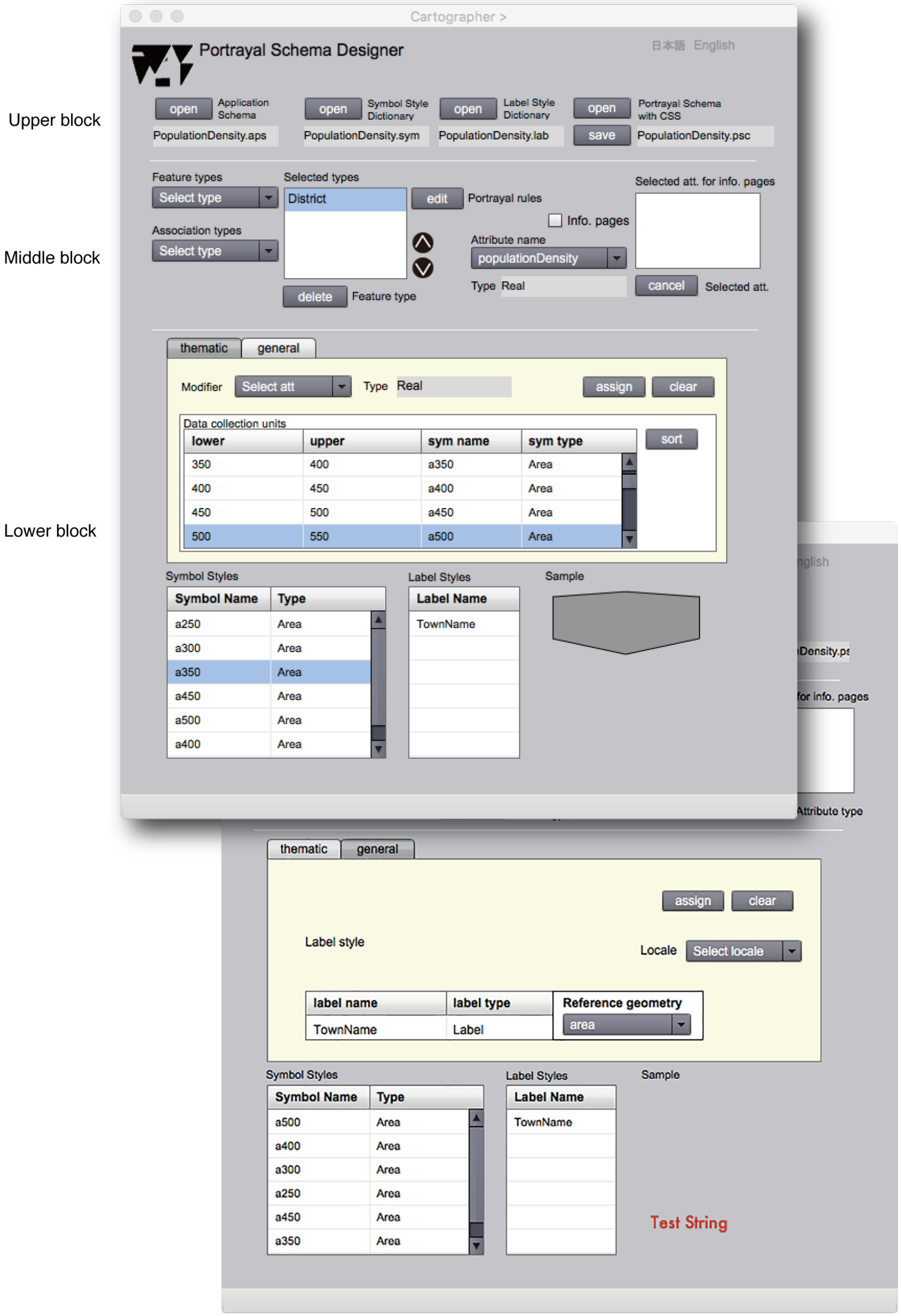


Figure 8.8-5. GUI of Portrayal schema designer. In the figure, ‘thematic’ tab shows the assigned classification of population density. While, ‘general’ tab shows the label style description. The red character string ‘Test String’ is a sample of the label.

information pages. The lower block enables the selections of symbol/label style for target attribute, and attribute value classification for the choropleth mapping.

Figure 8.8-5 illustrates the portrayal schema for the choropleth map to show the population density. A name of each census district is portrayed by the label style 'TownName'. Its position is located at the center of the geometric attribute 'area'. Population density is classified into intervals of 50 persons/km^2, and each interval is portrayed by the corresponding area symbol. The classification is shown at the lower block in the thematic tab.

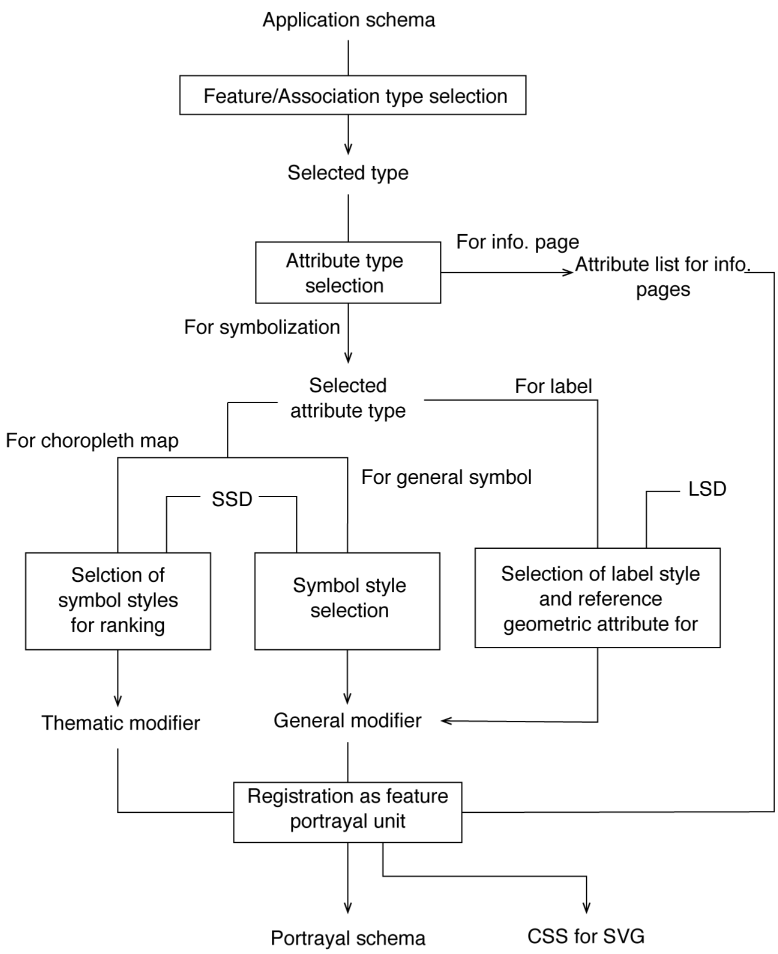


Figure 8.8-6 Portrayal schematization flow

Figure 8.8-6 illustrates the flow of portrayal schematization. Each feature is represented by one symbol, one label and information pages for multifaceted attributes. Outputs of this flow are a portrayal schema and a Cascading Style Sheet (CSS), which is used to display SVG file on the Web browser.

Map editor

Map editor is used for designing a map automatically by transferring the geospatial objects, and by the layout of marginalia by hand. It also is used as a map viewer. A map consists of map face and marginalia in accordance with the map instance model as introduced in the section 4.4.1. Map editor has three functions to provide cartographic products. They are designing of a map face, alignment of

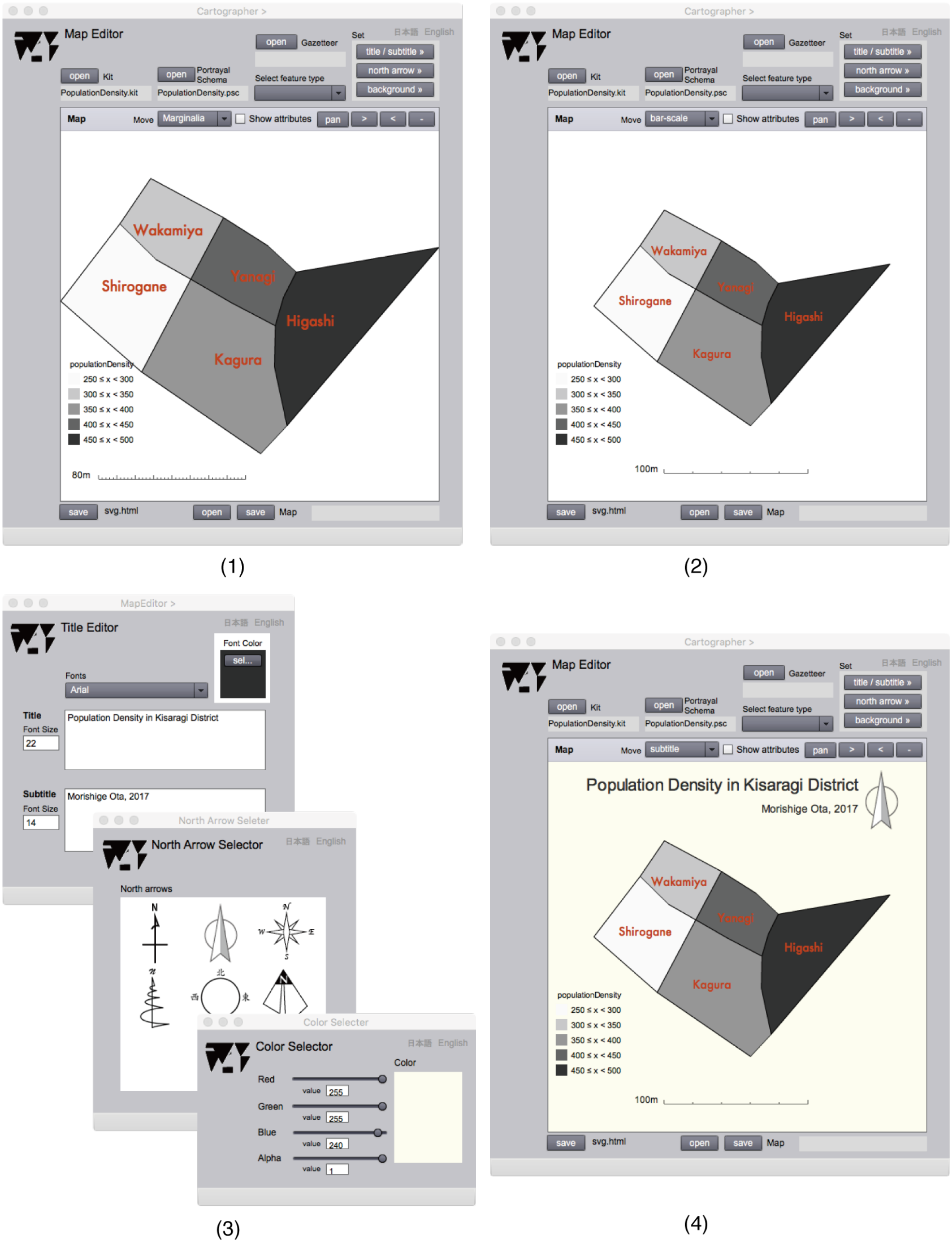


Figure 8.8-7. Editing procedure for static map

a marginalia, and providing of files for distribution.

As shown in Figure 8.8-7, the typical procedure of map designing is,

(1) Automatic configuration of a map face,

(2) Position adjustment for labels, legend and bar-scale by hand,

(3) Title setting, north arrow selection, and background color setting if required,

(4) Verification of the map.

Figure 8.8-8. Interactive map for display of multimedia feature attributes on information pages. (https://www.kagurazaka-bishamonten.com/about accessed 2018-11-03)

The map face is automatically designed in compliance with the portrayal schema. Marginalia consists of title setting, subtitle setting, bar-scale location, legend location and north-arrow selection and location. Those are done manually. Currently, the implementation of generalization assistance function is out of scope for gittok. It will be a future issue. If a portrayal schema is defined for the interactive map, the operation to see information pages will be possible on the Map editor by selecting the feature and attribute type (Figure 8.8-8).

As already mentioned at 8.8.1, the Map editor allows to select three different formats to produce a map file for distribution. They are the internal format, HTML with SVG, and GeoJSON. A set of cartographic objects with the internal format can be restored in gittok, and the user can edit the map again. While the files with other formats cannot be no longer restored in gittok, because they are graphic files and no longer follow the map instance model and application schema.

Map representation on the web

HTML file provided by gittok is a map using <svg> element following HTML V.5 for distribution to the public. Figure 8.8-9 shows a sample of the map displayed on the Web browser. The reasons to choose SVG are as follows (Eisenberg and Bellamy-Royds 2014).

i. SVG is an open standard developed by the World Wide Web Consortium (W3C) since 1999 (https://www.w3.org/TR/SVG11/ accessed 2018-01-04).
ii. SVG is resolution independent as it is provided for the vector graphics.
iii. SVG supports the event handlers enabling the coding of interactive maps.

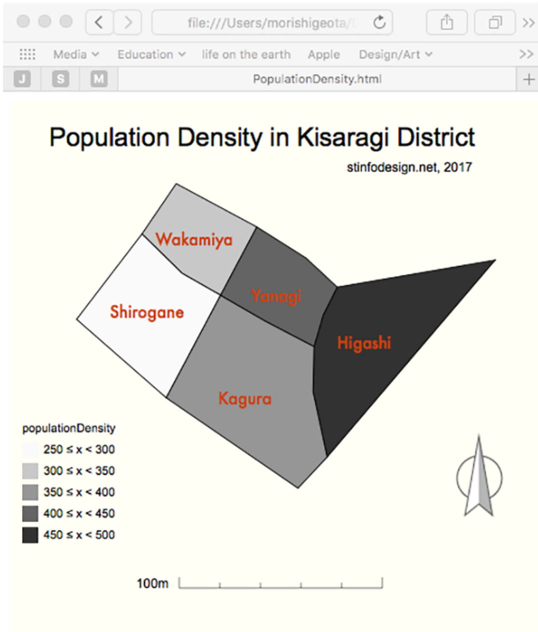


Figure 8.8-9. SVG map displayed on the Web

GeoJSON file

Gittok can output data following GeoJSON. It is a standard provided by the Internet Engineering Task Force (IETF) commonly used in web-mapping software and GI systems (RFC 7946: 2016, https://datatracker.ietf.org/doc/rfc7946/ accessed 2018-05-23).

By the way, TopoJSON as an extension of GeoJSON is also used today, because it enables to avoid the redundancy of GeoJSON by sharing of boundaries between adjacent polygons. GeoJSON files provided by gittok can be converted to TopoJSON as gittok ensures topological consistency.

The output function of GeoJSON file may seem to be a part of Exchanger as it is a function to convert a kit to a file with the system independent format. However, as already mentioned, GeoJSON file is independent from cartographic objects with the map instance model. Moreover, before encoding, we should select physical features and their attributes including geometry for mapping. We also select attributes for information pages in accordance with the portrayal schema. Therefore, GeoJSON file is called a cartographic product in gittok.

Gazetteer designer

A gazetteer is created for each feature type. The user can declare the creation of new gazetteer by the selection of an appropriate feature type from the dropdown list in the Gazetteer designer (Figure 8.8-10), and by the selection of a geographic name as an index and a geometry attribute to show the position of a feature.

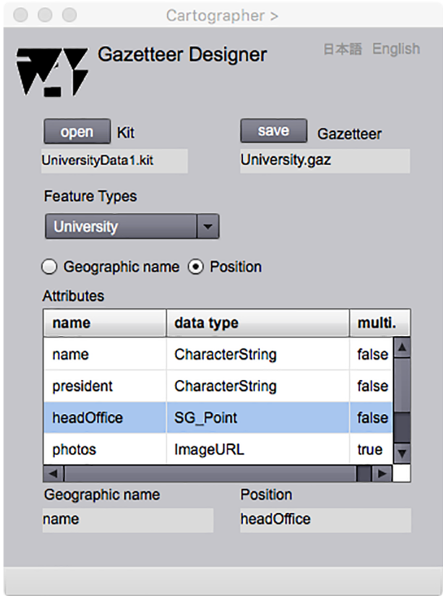
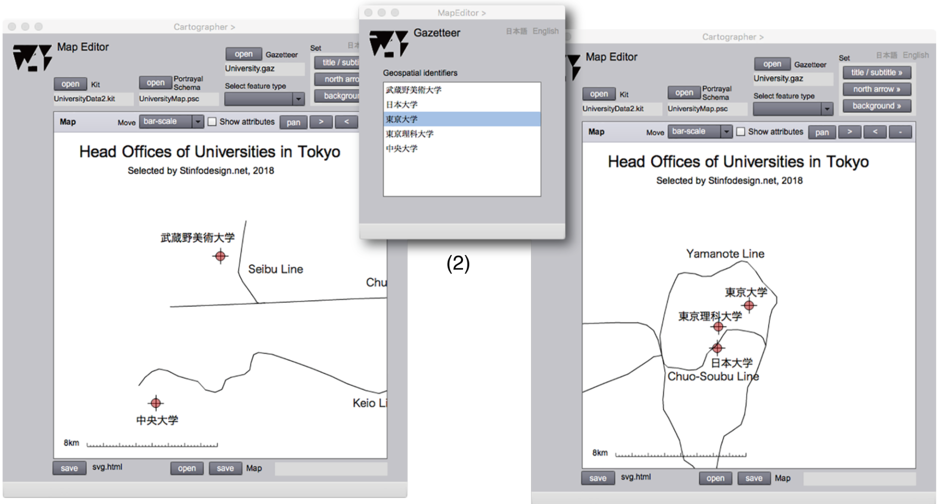


Figure 8.8-10. Gazetteer designer

Gazetteer representation

A gazetteer has a function for translating a feature to the center of the screen by selecting a geographic identifier (Figure 8.8-11).



(1)

(3)

Figure 8.8-11.   Example of gazetteer. (1) Initial screen. (2) '東京大学' is selected in the gazetteer. (3) '東京大学' is moved to the center of the screen.

8.8.3. Listing tools

Introduction

Listing tools enables a list compilation by selecting virtual feature instances. It consists of two components: the List schema designer and the List viewer. The List schema designer enables to create a rule for listing by the selection of feature type and attribute types. The user may obtain information about features in accordance with the list schema on the list viewer.

List schema designer

A list schema defines the representation rules for virtual features. Proxy attribute is the mandatory item as it is used as the key element. Other attributes representing in the list are chosen as 'selected attributes' in the list schema designer (Figure 8.8-12). The operation order for the list schema design is as follows.

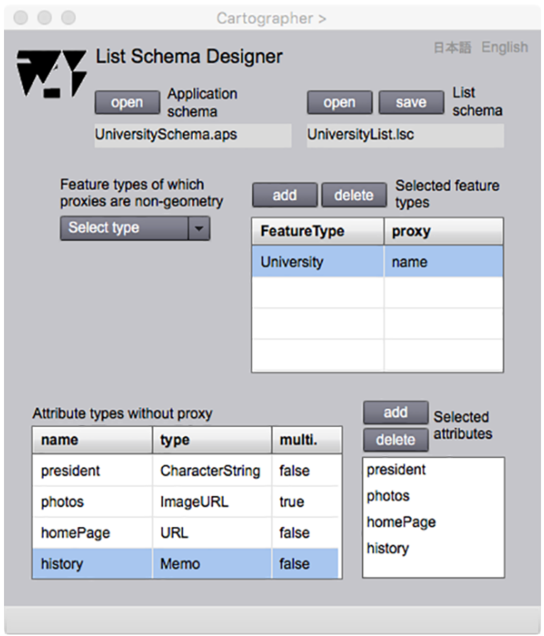


Figure 8.8-12. List schema designer

i. Select feature type for listing. It is added in the table for 'Selected feature types'.

ii. Select the feature from the table for 'Selected feature types'.

iii. Select attributes for display in the list from the table for 'Attribute types without proxy'.

iv. Repeat ii to iii, until finishing the attribute selection for every feature type in the table for 'Selected feature types'.

v. Save the list schema in the storage folder.

List viewer

A list works as an index to get feature attributes. For example, 'Proxy value list' at the left side of Figure 8.8-13 is a list face. By selecting a proxy value and an attribute type, the user can see the multifaceted attributes such as simple data, image, web site, memo, video and audio. There are no video and audio in Figure 8.8-13 because it happened to have no such attributes in this feature. List viewer also has a function to provide the html file for the distribution to the public by pressing 'save' button at the bottom of the List viewer.

Figure 8.8-13. List viewer. It represents character string, photos, home pages, and memos. (https://www.u-tokyo.ac.jp/ja/index.html accessed 2018-08-25)

Figure 8.8-14. HTML list and attribute pages of a feature displayed on the web browser (http://www.musabi.ac.jp accessed 2018-08-25)

List representation on the web

List viewer may provide the list with HTML format for the distribution in the Internet environment. The HTML file enables to access not only simple data such as string, number, memo and address, but also multifaceted attributes such as image, video, audio, and home page. Figure 8.8-14 is a part of HTML list, which shows the content of Musashino Art University. A photo of campus and a home page are information pages hyperlinked by the homepage of this list.

We should keep in mind that listing is primarily prepared for the representation of virtual features. However, a place-names dictionary can be provided by changing proxy attribute from geometric to non-geometric. Currently, GISs having such a capability seems to be few because most of existing GISs does not assume the management of virtual features.

8.9. Exchanger

8.9.1. Introduction

Exchanger can transform schemata and objects from/to internal format called Action Message Format (AMF) to/from XML as shown in Figure 8.9-1. AMF is a binary format provided by Adobe Systems used to serialize objects. Gittok uses AMF as it is developed by using ActionScript V3.0.

8.9.2. Exchanger

Exchanger enables to encode and decode geospatial data and cartographic data in accordance with their instance models for the description of XML documents. The types of data that can be exchanged are shown in Table 8.9-1.

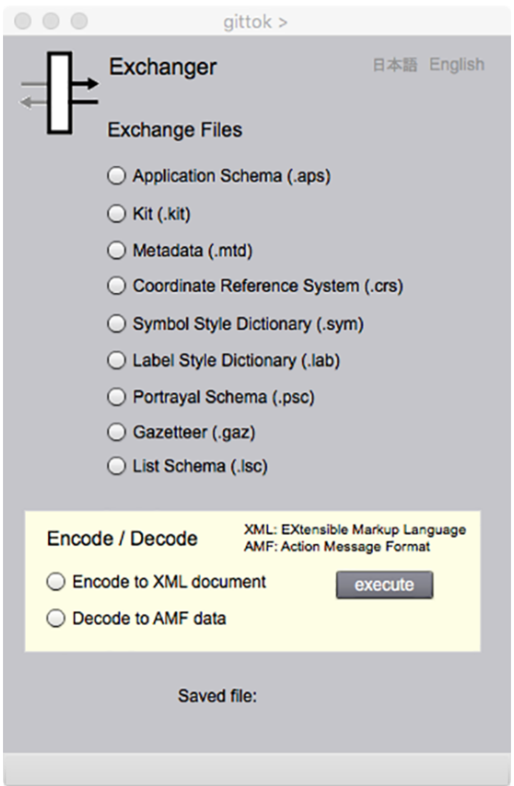


Figure 8.9-1. GUI of Exchanger

Table 8.9-1. Data for exchange as XML document.

Types	Exchangeable data	file name extensions
Geospatial data	Application schema	.aps
	Kit	.kit
	Metadata	.mtd
	Coordinate reference system	.crs
Cartographic data	Symbol style dictionary	.sym
	Label style dictionary	.lbl
	Portrayal schema	.psc
	Gazetteer	.gaz
	List schema	.lsc

The following text is an example of the XML document representing the CRS. The document structure is in accordance with the schema for CRS shown in Figure 3.4-1 illustrated at the section 3.4.

```
<CRS id="JGD2011, TP / 9(X, Y), H">
  <datum>
    <GeodeticDatum id="JGD2011" name="Japanese Geodetic datum 2011">
      <ellips>
        <Ellipsoid name="GRS8.0" semiMajorAxis="6378.137" inverseFlattening="298.257222101"/>
      </ellips>
      <pm
        <PrimeMeridian greenwichLongitude="0"/>
      </pm>
    </GeodeticDatum>
  </datum>
  <vDatum>
    <VerticalDatum id="TP" name="Mean See Level of Tokyo Bay"/>
  </vDatum>
  <cs>
    <CoordinateSystem id="9(X, Y)" name="Plane Rectangular Coordinate System 9" dimension="3">
      <axis>
        <Axis name="X" direction="north" uom="m"/>
        <Axis name="Y" direction="east" uom="m"/>
        <Axis name="H" direction="vertical" uom="m"/>
      </axis>
      <ps>
        <GaussKrugerProjection scaleFactor="0.9999" falseEasting="0" falseNorthing="0">
          <origin>
            <Coordinate component="36,139.8333333" dimension="2"/>
          </origin>
        </GaussKrugerProjection>
      </ps>
    </CoordinateSystem>
  </cs>
</CRS>
```

8.10. Functional suitability

8.10.1. Introduction

As long as gittok aims to be a learning assistance tool for the introductory education about GIT, it should be confirmed whether the implemented functions are suitable for its purpose or not. The purpose of the introductory education is that students will understand the fundamental image of GIT as a preparation for learning deeper knowledge. The fundamental image of GIT is described as the concept map and six knowledge areas. Gittok assists the students so that they can get the fundamental image through exercises.

In this section, we confirm that gittok was developed so as to meet the requirements, in terms of functional suitability. Functional suitability, in other words, functionality, is defined as a degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions. This definition is provided by 'ISO/IEC 25010 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models' (ISO/IEC 2011). The functional suitability is measured as three characteristics: completeness, correctness and appropriateness. Functional completeness is defined as a degree to which the set of functions covers all the specified tasks and user objectives. Functional correctness is defined as a degree to which a product or system provides the correct results with the needed degree of precision. And functional appropriateness is defined as a degree to which the functions facilitate the accomplishment of specified tasks and objectives.

8.10.2. Functional completeness

In the case of the introductory course using gittok, the topics that are described in Table 8.10-1 are the specified tasks on the viewpoint of knowledge areas. And functions indicated in Table 8.10-2 cover all topics taken up in the education. There is no inconsistency between two tables. Therefore, we may conclude that gittok fulfills the functional completeness.

Table 8.10-1. Topics taken up in the class (Ota 2017)

Knowledge areas	Topics
Modeling	- Modeling is a process to create an application schema. - GFM is the metamodel to show the rules for application schema. - Application schema is the rules for the geographic objects description in accordance with the universe of discourse. - Geospatial objects are a set of features and feature relationships - A feature may have spatial, place and thematic attributes (Gittok does not equip temporal attribute types at this moment). - Operations such as a spatial analysis are declared in a feature type. - A feature type associates with other feature types or itself.
Acquisition	- Acquisition is a process to get geographic objects following application schema. - Geodetic coordinate is defined by the coordinate reference system which is comprised of geodetic datum and projection methods.
Management	- Management is a process dealing with geographic objects as resources for applications. - Metadata as information about resource includes, at least, title and keywords (what), geographic extent (where), publishing date (when), overview (why and how), and responsible party (who). - Geographic objects registered in geo-library are retrieved by assignment of metadata elements. - Geo-library is used as a main part of the spatial data infrastructure.
Analysis	- Analysis is a process to detect an essence of geographic phenomena by operations. - Execution of an operation is possible by defining its syntax as a property of feature/association type. - The result of operation is embedded in a feature/association as a derived attribute.
Exchange	- Exchange is a process to encode data with the system independent format by a sender and to decode the data by a receiver. - Application schema is a common basis for data understanding between senders and receivers. - An instance model and a markup language such as XML is used for the implementation of data for exchange.
Representation	- Representation is a design process of maps and lists in accordance with the portrayal schema and the list schema. - GPM is the metamodel to provide the rules for portrayal schema and list schema. - The portrayal schema with SSD and LSD or the list schema shall be designed to meet the purpose of representation such as general-purpose map, choropleth map, interactive map, gazetteer and geographic name dictionary. - Editing of maps such as re-positioning of labels and marginalia are required for map editing - Geospatial information are distributed by the format conversion from the internal format to the system independent formats such as HTML, SVG and GeoJSON.

Table 8.10-2. Software components and their functions corresponding to knowledge areas (Ota 2017)

Gittok componentss	Functions and sub-modules
Modeler	*Application schema modeling* Feature Type designer, Association Type designer *Property type declarations* Attribute Type designer, Operation Type designer
Editor	*Instance editing* Feature editor, Association editor *Attribute editing* Geometry, Address, Image, Sound, URL, Video, Text, Memo, and List *Coordinate reference system declaration* CRS Editor
Manager	*Metadata creation* Metadata editor *Metadata and geospatial objects (kit) searching* Geo-library
Analyst	*Geospatial analysis* Feature analyst, Association analyst
Exchanger	*Encoding and decoding* Application schema, kit, metadata, CRS, Symbol style dictionary, Label style dictionary, Portrayal schema, Gazetteer, List schema
Cartographer	*Mapping* Symbol Style Designer, Label Style Designer, Portrayal Schema *Designer* Gazetteer Designer, Map Editor *Listing* List Schema Designer, List Viewer *Representation on the web* Map Editor, List Viewer

8.10.3. Functional correctness

Functional correctness is defined as a degree to which a product or system provides the correct results with the needed degree of precision. As gittok is an interactive system, software correctness can be confirmed relatively easier than other type of software because the response to the user's request is represented visually. And most of the produced information can be confirmed by the encoded XML document. Therefore, correctness of feature operations is confirmed by using data for inspection during their developments. Today, the conspicuous bugs are not found by the practices of the educations. However, it is difficult to confirm the correctness about all cases of procedures. Reports from users will be highly appreciated.

8.10.4. Functional appropriateness

Functional appropriateness is defined as a degree to which the functions facilitate the accomplishment of specified tasks and objectives. For example, appropriateness is measured as that a user is only presented with the necessary steps to complete a task, excluding any unnecessary steps. In the case of gittok, usually the user can practice exercises appropriately without wavering, because software is classified hierarchically into modules that correspond with knowledge areas, and topics in the module are stratified as sub-modules. The user can choose the next task from the executable sub-modules. However, the user should think which sub-module is appropriate, because gittok is an education assistance tool and it does not always indicate the answer automatically. However, if gittok did not accept user's action, the user will see an alert on the screen.

As a consequence, we can conclude that gittok reaches the enough level of functional suitability as a learning assistance tool of GIT. However, off course, we always need to improve gittok to realize the better functionality.

9. Education practices

9.1. Introduction

Teaching experiences on GIT by the author has been progressing in three phases. The first phase is the semester-long course for undergraduate students at The University of Tokyo three times from 2008 to 2010. The second phase is the semester-long course for undergraduate students at Chuo University two times from 2014 to 2015. The third phase is the three-day course for beginners and engineers in the GIS related industry held by the GIS Certification Association (GISCA) organized by GIS Association of Japan (GISA) since 2016. These experiences are reported in this chapter before discussing the validation of the introductory course held by GISCA.

9.2. First phase (The University of Tokyo)

The author carried out the subjects so that the students will gain the literacy on geospatial modeling and GI standards, because they are useful for research and development on GIS&T. As a matter of fact, the theme of more than 50% of classwork was conceptual modeling as shown in Table 9.2-1. We did not use any software tool, as there were no suitable tools at that period. However, the merit of this teaching method was that we could spend longer time to discuss the theory behind of GIT.

Table 9.2-1. Syllabus for the subjects about literacy on GI modeling used in 2010

No.	Subjects	Knowledge areas
1	Prologue	
2	Spatial thinking and conceptual modeling	Modeling
3	Formal description of conceptual models by UML	Modeling
4	General feature model	Modeling
5	Application schema	Modeling
6	Spatial schema 1	Modeling
7	Spatial schema 2	Modeling
8	Temporal schema	Modeling
9	Spatial reference systems	Acquisition
10	Coverage	Modeling
11	Spatial analysis	Analysis
12	Map representation	Representation
13	Geospatial data acquisition	Acquisition
14	Geospatial information technology (GIT)	All
15	Final test	

The teaching materials such as slides were prepared based on the texts made for the seminar previously performed. It was the GI standards studio held by the Spatial Information Technology SIG (SIT-SIG) in GISA since 2003 till 2007. The author was a secretary of the organizing group. This seminar was awarded by the GISA in 2006. The GI standards studio was the experience in the prehistoric phase before the experience at The University of Tokyo.

The first phase triggered the development of gittok. Because, it was found empirically that it is difficult to learn a whole image of GIT without exercises using a learning assistance software. The reason is that GIT contains a lot of 'procedural' knowledges.

9.3. Second phase (Chuo University)

9.3.1. Introduction

The author carried out the introductory course on GIT at Chuo University from 2014 to 2015 for total 14 final grade undergraduate students. All students were members in the information science faculty and they did not have any knowledge about GIT. The early version of gittok was used experimentally each time. It was a good opportunity to improve functionality of gittok, and to solve problems and bugs found during exercises in the class room.

Table 9.3-1. Syllabus for the subjects about GIT used at Chuo University in 2015

No.	Subjects	Knowledge areas
1	Prologue	
2	Let's get started with gittok	Acquisition, Representation
3	Execution of the operation defined in a feature type	Analysis
4	Object oriented modeling and application schema	Modeling
5	Exercise to get plot ratio	Acquisition
6	Scratch building of application schema for store siting	Modeling
7	Data acquisition for convenience store siting as the group work	Acquisition
8	Review of the siting exercise	Modeling, acquisition
9	Coordinate reference systems	Acquisition
10	Metadata and geo-library	Management
11	Introduction to XML	Exchange
12	Map design 1	Representation
13	Map design 2	Representation
14	Spatial schema	Modeling
-	Final exercise as homework	All

9.3.2. Syllabus

The syllabus created for the course in 2015 is shown in Table 9.3-1. The syllabus should be designed so that students can overview six knowledge areas in GIT based on GI standards. However, without the fundamental knowledge of UML, it is difficult to understand the conceptual structures of GIT. And, without the knowledge of XML, it is difficult to understand the system independent data description for exchange between different systems. Therefore, the introductions of UML and XML are mandatory in the syllabus as prerequisites of the course. Moreover, the simple exercise for 'getting started with gittok' was prepared at the starting of the class, as it is effective for the students familiarize with gittok. The purpose of exercises is to ensure that the students understand the concepts in GIT rather than merely learning how to use gittok.

9.3.3. Application schema modeling

The modeling of an application schema is one of the important exercise in this class. The students build the application schema from scratch for store siting, in the subject No 6. Several different but valid schemata are proposed. The students find that some of schemata are transformable but the reliability and difficulty of data compilation are different. Figure 9.3-1 shows examples the students proposed. The diagram (a) is the simplest application schema. However, it cannot ensure that the combination of building attribute values is correct. And it is unclear that storeLocation is an attribute of a trade area or not, as it should be an attribute of the store. The diagram (b) is improved so that the combination of building attributes is ensured by the definition of Building. And storeLocation becomes an attribute of Store. However, the relationship between arguments and return value of the operation locationFacility is unclear. Finally, (c) is the version to describe the operation to obtain store location by building information. However, the definition of BtoS association makes the schema a bit more complex. All of them are valid to estimate the store location. The students try consensus making before data acquisition as the group work. And they understand that process of consensus making is important to move forward with the project. By the way, the lecturer recommended (c), as it describes the relationships between buildings and store clearly, and the function of the association BtoS is clarified.

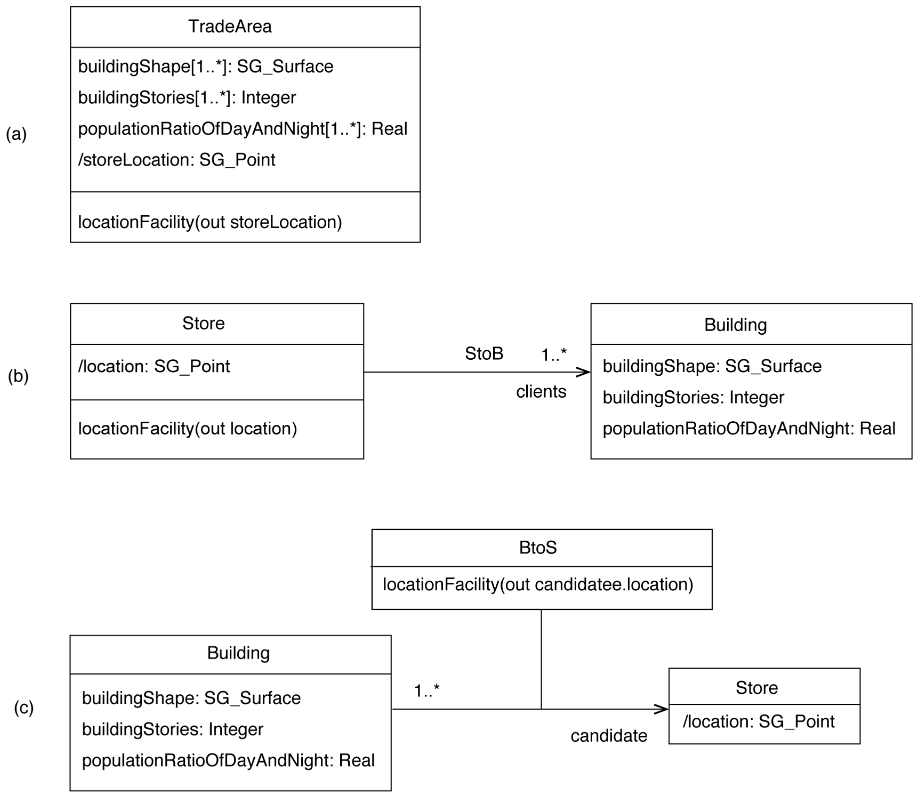


Figure 9.3-1. Alternative schemata for store siting. Arguments for syntax of locationFacility are simplified in the diagram. Actual arguments are buildingShape, buildingStrories and populationRatioOfDayAndNight.

9.3.4. Final exercise

The grades of students are evaluated based on the final exercise instead of the final examination. One example of the submitted interactive map is shown in Figure 9.3-2. The students 1) modeled the application schema, 2) investigated in the field, 3) captured feature instances with geometric and non-geometric attributes in accordance with the application schema, 4) modeled the portrayal schema for interactive map by the selection of features and attributes, 5) designed symbols, labels and map marginalia, then 6) created an interactive map in their areas of responsibility.

From the results of the final exercise, we can confirm that it is possible to provide an introduction to GIT based on GI standards in the form of a college-level course to new students.

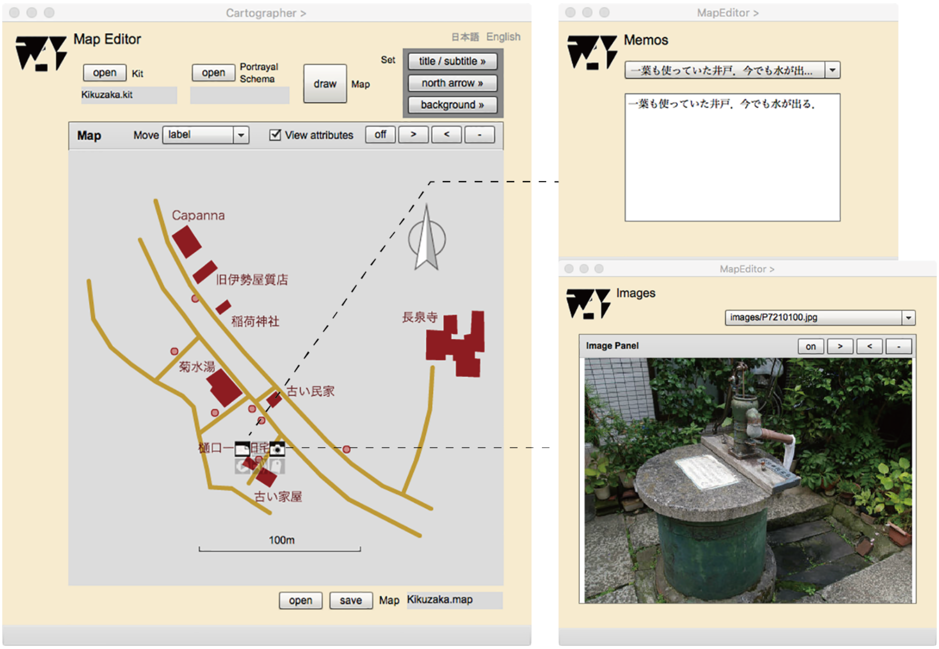


Figure 9.3-2. An example of submitted interactive map as the result of final exercise submitted by Maki Yamada in 2014. The photo of the well was taken in front of the house of Ichiyō Higuchi (1872-1896), the Japanese female writer in the Meiji Period (1868-1912).

9.4. Third phase (GIS Certification Association)

9.4.1. Introduction

GISCA organized the first open seminar on the introduction to GIT at the Shinjuku Campus of Musashino Art University, Tokyo, Japan from January 8 to 10, 2016. The title of the seminar is the Geographic Information Technology Education Course (GITEC). The length of the course is 3 days (21 hours). The duration is nearly the same as the semester-long course. Anyone can participate in this course, and the students were beginners and engineers working in GIS related industries. Twenty-one students completed the first GITEC. Since then, GITEC beginner's course has been held two times more, August 19 to 21, 2016 at Musashino Art University, and May 24 to 26, 2017 at Kokusai Kogyo Co., Ichigaya, Tokyo.

9.4.2. Syllabus

The syllabus using in practice is described in Table 9.4-1. This was made based on the syllabus used in the subjects in Chuo University.

Table 9.4-1. Syllabus for GIT introductory course held by GISCA since 2016.

No.	Subjects	Knowledge areas
1	Course guidance and introduction to GIT	
2	Let's get started with gittok	
3	Object-oriented modeling	Modeling
4	Introduction to GFM and application schema	Modeling
5	Application schema modeling (group work)	Modeling, Acquisition
6	Plot ratio (Floor area ratio) estimation	Analysis, Modeling, Acquisition
7	Store siting (group work) 1	Analysis, Modeling, Acquisition
8	Store siting (group work) 2	Analysis, Modeling, Acquisition
9	Portrayal schema designing	Representation
10	Mapping to show the result of siting	Representation
11	Spatial schema	Modeling
12	Coordinate reference systems	Acquisition, Exchange
13	Metadata and geo-library	Management
14	Geospatial data exchange	Exchange
15	Gazetteer and listing	Representation

9.4.3. Schedule

First day

In the first day, students answer the pretest after the guidance of this course. Then they perform a simple exercise 'Getting started with gittok' as same as the class at Chuo University. In the afternoon, the students learn the UML class diagram at first. Then they are split up to groups of four or five members. Each group discusses and design application schemata for the themes given by the lecturer as follows.

1) Definition of a feature type 'Building' with shape as surface, area as real value, and operation to get an area.

2) Definition of an application schema including an inheritance of the super type. The theme is the facility management of city traffic system. Given domain of discourse is as follows.

- Station is a point where passengers get in and get out, which is located between the terminal stations.
- Station is categorized to a bus station and a tram station.
- Bus station is signified by a sign pole put at its location.
- Tram station is signified by a platform on the road.

3) Definition of an application schema including an association between different feature types. The

theme is the estimation of the risk caused by tsunami. Given universe of discourse is as follows.

- The place suffered by the tsunami is called the affected area.
- A tsunami has its name and day and time of occurrence, and is associated to the earthquake that caused it.
- There are more than one affected areas suffered from one tsunami. An affected area is characterized by its name, the affected extent, the number of affected houses, and the number of affected people.
- Since the height of the tsunami depends on the affected area, the height of the tsunami is considered to be the attribute of the association between the tsunami and the affected area.
- There is a possibility that the same area is affected by different tsunamis.

Next in the afternoon of the first day, the exercise for learning a concept of feature association including an operation is performed. Each group discuss and model an application schema to estimate plot ratio. Plot ratio is also called a floor area ratio, floor space ratio, and so on depending on the country and region, and it is defined as a ratio of the total floor area to the site area of a building. Local governments restricts the plot ratio for each use district to maintain a quality of life. The plot ratio is lower in residential districts and higher in commercial districts. The students confirm that the plot ratios in the test area do not exceed the limitation imposed by the local government.

The sequence of the exercise is as follows.

i. Application schema modeling through group discussion
ii. Presentations of application schemata modeled by groups
iii. Designation of the application schema used in the exercise by the lecturer
iv. Application schema implementation by using Modeler
v. Digitizing of a geographic objects by using Editor as the group work
vi. Determination of the plot ratios by the executions of operations in the Analyst.

A 'getPlotRatio' operation is placed in association 'LtoB' between HousingLot and houses, because the plot ratio is obtained as the ratio between housing lot and total floor area of buildings (Figure 9.4-1). The arguments of 'getPlotRatio' are the shape of housing lot, the shapes of buildings, and the stories of buildings. The returned value is fed into the derived attribute 'plotRatio' kept in LtoB.

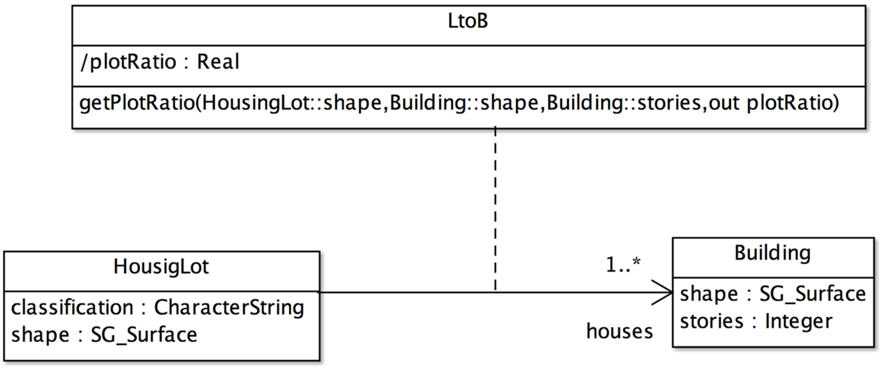


Figure 9.4-1. Application schema for estimation of plot ratio.

The students first digitize the housing lots and houses, and then define associations between housing lots and the related houses. Finally, the operation 'getPlotRatio' is executed (Figure 9.4-2). The association between the housing lots and houses is shown by the blue triangles (from) and the red triangles (to). The blue triangle appears to be put on the house, because it is automatically positioned to the center of the housing lot.

Open a kit on Analyst page. Then select Association tab and select "LtoB" in Association type list.
1. Select an instance in Association instance ID list.
You can confirm the instance by clicking "show" button.
2. Select getPlotRatio in "execute" operation list. Then click "execute" button. The plot ratio is shown in Return attribute value list.
3. Obtain two other plot ratios.
4. Save the kit with plot ratios.

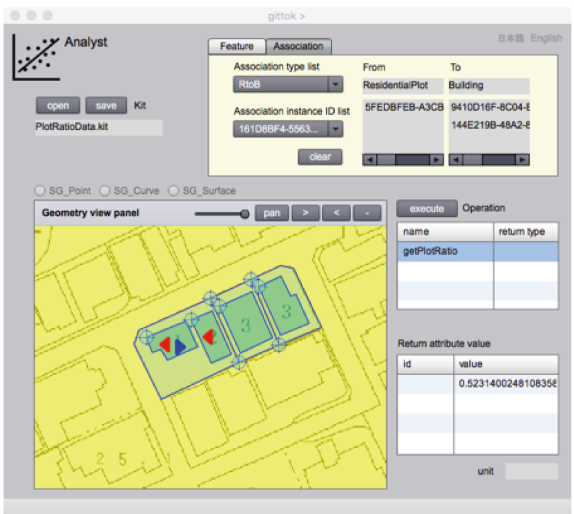


Figure 9.4-2. Execution of operation defined in application schema

Second day

The main theme of the second day is the store siting. The students try the store siting of a convenience store in the city. Students learn the functions, especially modeling, acquisition, analysis, and representation and their relationships in GIT through this exercise. At first the students discuss the modeling of the application schema as already introduced in the section 9.3.3. Next, they digitize buildings and their attributes in the estimated territory of the store as a group work. Objects digitized by each student are merged to one file, after the quality check by another member. The position of the store is estimated by the siting operation at Analyst. Then they design the map to show the candidate store location after learning of portrayal schema, symbol style dictionary and label style dictionary.

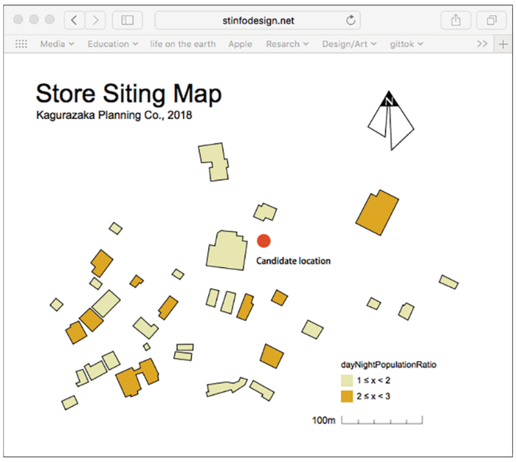


Figure 9.4-3 Map to show the candidate location

Third day

In the morning of the third day, students take lectures about spatial schema and CRS at first. Those subjects were already briefly explained during application schema modeling and object acquisition. Therefore, the lecture is done as knowledge confirmation.

Next subject is the metadata and geo-library. Gittok does not use Data Base Management System, and the geo-library is not located on the Web. However, students can describe metadata and retrieve a kit using the personal geo-library. As the background of index map to show extents of kits, gittok adopts Open Street Map provided by the Open Street Map Foundation (http://www. openstreetmap.org) and GSI Map provided by the Geographic Information Authority of Japan (http://portal. cyberjapan.jp). They enable gittok to retrieve a kit by a geographic extent (geographicExtent) as introduced in the section 8.6.

Next subject is geospatial data exchange as XML documents. The students first learn the introduction to XML, then they create XML documents by translating application schema, a kit, metadata, and so on, by using Exchanger. Students exchange documents with others and confirm the contents and structure of the XML documents.

If there is time remained, the students create a gazetteer by themselves and examine the gazetteer to find the location of a feature. Then, the students learn the listing as a mechanism to display feature attributes by using non-geometric proxy attribute. They make the list schema and create a list, then display a list on the screen.

Finally, the students take the posttest to evaluate the increasing of their knowledge. Those who get more than 70 points （perfect score is 100） will be able to participate in subsequent courses.

10. Validation of education practices

10.1. Introduction

It is nearly impossible to compare the effectivity of the education using gittok with other education courses having the same purpose. Because, at least by the Internet research, few similar courses and few similar applications were found. Therefore we apply the four level evaluation model (Kirkpatrick 1979), as it is a well-known learning evaluation approach. It has been used as a basic model for the identification and targeting of training-specific interventions in business, government, the military, and industry alike (Rouse 2011). According to this model,

i. Reaction (how learner feel about instruction),
ii. Learning (learner performance on in-class test),
iii. Behavior (learners use the acquired knowledge after the class), and
iv. Results (organizational benefit such as improvement of working efficiency)

are measured. However, the fourth level of evaluation involves in measuring organizational impact of education, and it is beyond the scope of GITEC evaluations, because the purpose of the introductory course is that students will gain the image of GIT as a preparation for learning the deeper knowledge. Therefore, we validate the education effectivity by reaction, learning and behavior.

10.2. Reaction

Unsigned questionnaire has been conducted to evaluate the reaction by students at the end of the first

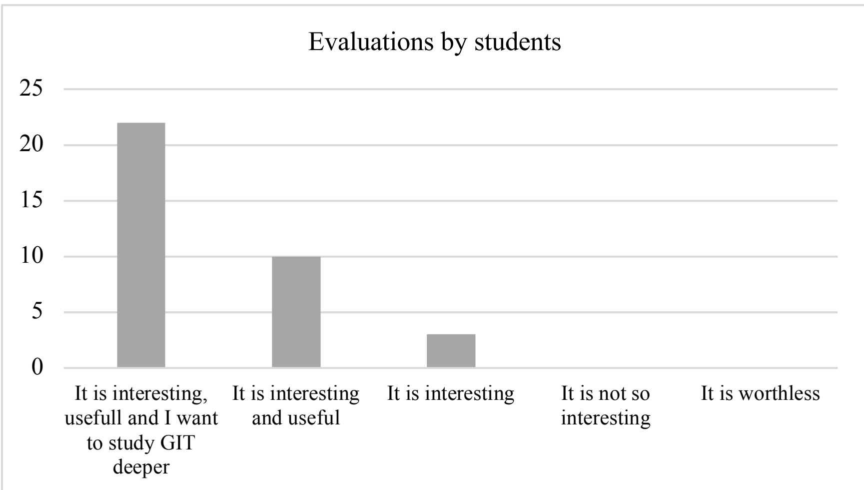


Figure 10.2-1. Result of the question “How do you evaluate GITEC?”

and the third courses (the total number of effective responses was 35).

63% of respondents answered that the education is interesting, useful and they want to learn GIT deeper. And 29% of respondents answered that the education is interesting and useful. (Figure 10.2-1). In addition, 54% of respondents were impressed that the course was easy to understand as a whole (Figure 10.2-2).

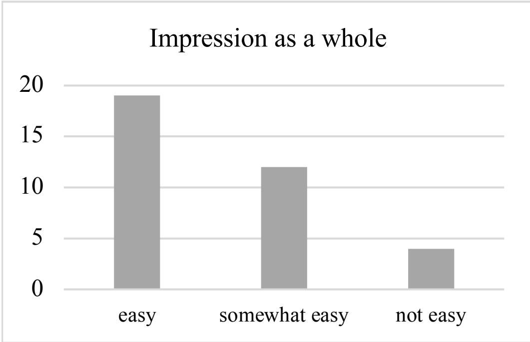


Figure 10.2-2. Impression of easiness as a whole

10.3. Learning

In order to investigate the understanding degree of the students, the pretest and the posttest were conducted every time (the total number of examinees was 48). Although the questions of the pretest and the posttest are the same, the lecturer does not give a response at the pretest. Therefore, students have to think the answers while attending. In addition, those who do not take 70 points or more for 100 points in a posttest cannot be completed students and they lose the right of participation in the

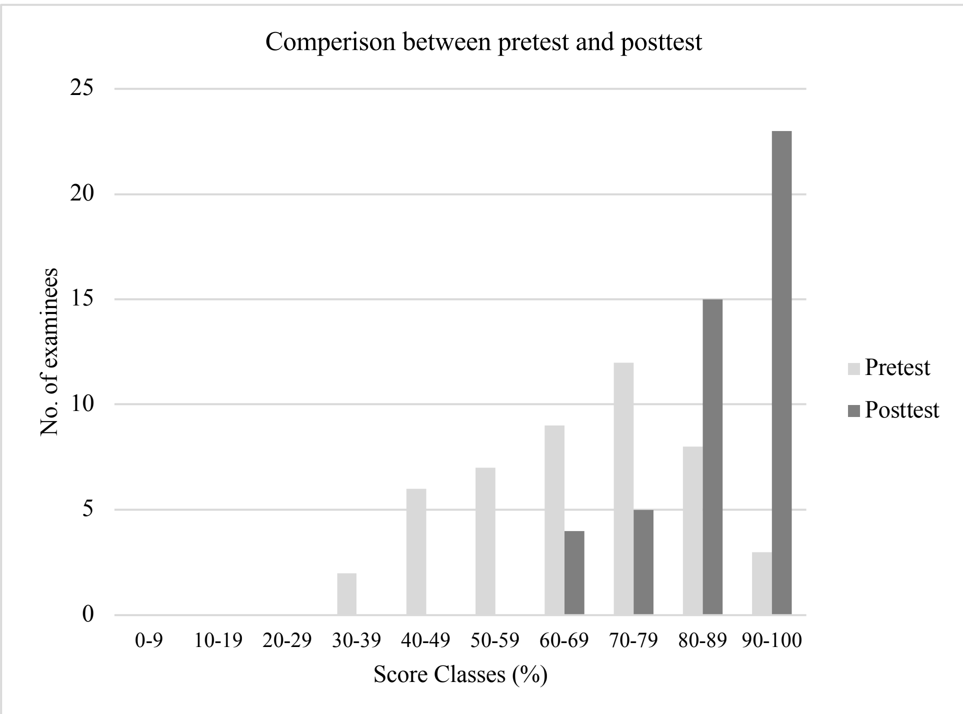


Figure 10.3-1. Change in the score from the pretest to the posttest.

subsequent courses. (Figure 10.3-1). As a result, the average score of the pretest was 67 points and the standard deviation was 16 points. The average point of the posttest was 87 points, and the standard deviation was 9 points. Since the passing score is 70 points, 24 people (50%) in the pretest stage failed, while in the posttest, 4 people (8%) failed. According to the t-test, there was a significant gain in the average of the posttest by comparing with the pretest (p-value = 0, the gain of average points is 20, 95% confidential interval is (16.6, ∞)).

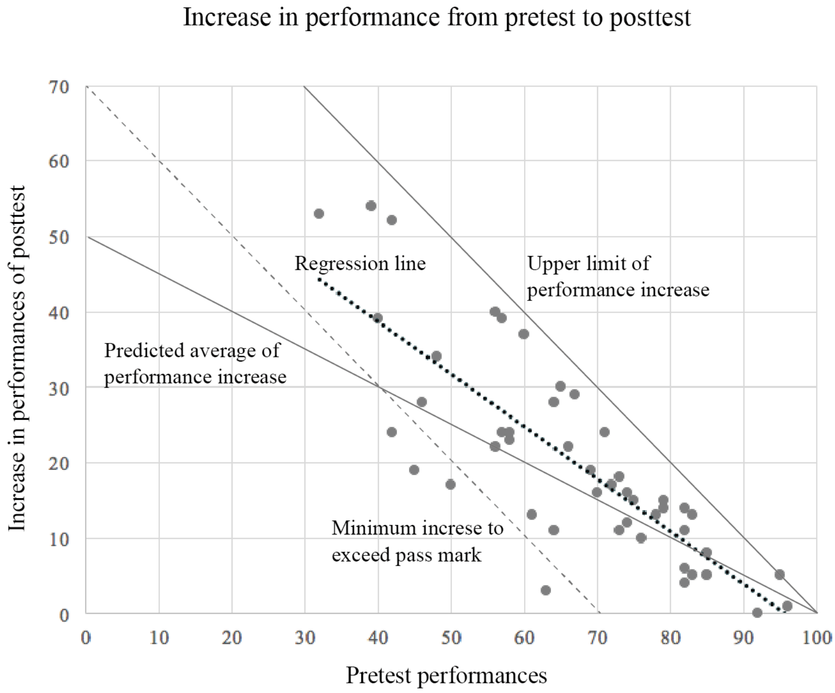


Figure 10.3-2. Increase in performances from pretest to posttest

Then, we examined the correlation between the performances of the pretest and the increase in performances of posttest (Figure 10.3-2). The correlation coefficient was -0.87. This indicates that even if the performance is low in the pretest, it tends to achieve remarkable progress in the posttest. Of course, the better scores that the students got in the pretest, the better they are able to hope for the score improvement in the posttest, even though they cannot take points beyond the full score. However, the slope of the regression line was steeper than the predicted average of performance increase when it is assumed that the score improvements distributed uniformly. The reason seems to be that the students cannot participate in subsequent courses unless they get more than or equal to 70 points in a posttest. By the way, only one person could get 100 points in the posttest.

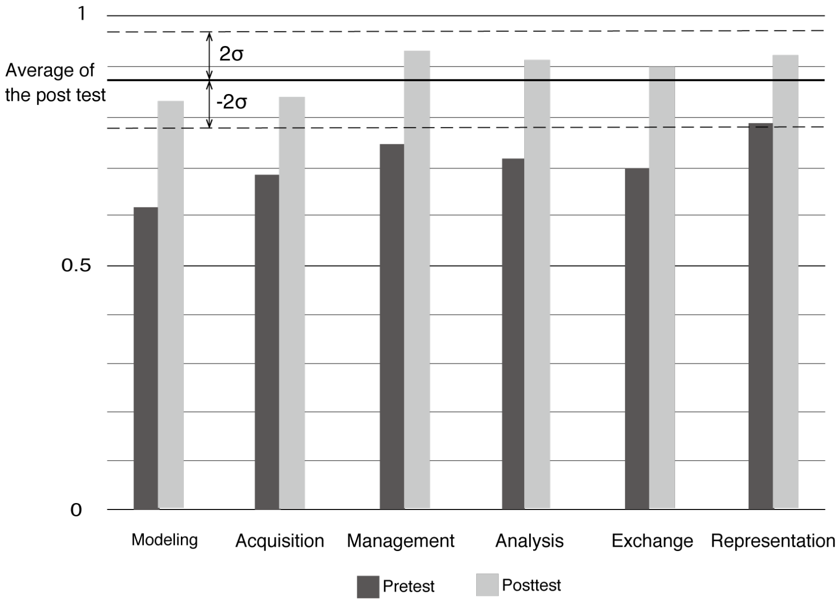


Figure 10.3-3. Performance improvement for each knowledge area from the pretest to the posttest.

As far as the difference in the correct answer rate of the pretest and the posttest is concerned (Figure 10.3-3), the result of modeling seems to be inferior to other knowledge areas. However, in the posttest, it is unlikely that there was a significant difference, since the deviation from the average value (0.89) to the result (0.83) was less than 2σ (1.3σ). Therefore, we may conclude that the levels of obtained knowledge did not have much difference among knowledge areas.

10.4. Behavior

Behavior of the students can be evaluated from three viewpoints. The first viewpoint is the number of students in the intermediate course held as the subsequent course of GITEC. 21 persons (first time: 14, second time 7) in 38 completed graduates (55%) participated in the intermediate level courses held after GITEC introductory course as shown in Figure 10.4-1. This fact indicates that more than 50% of completed graduates felt that this education is meaningful, even after four months to eleven months since participation in the introductory course.

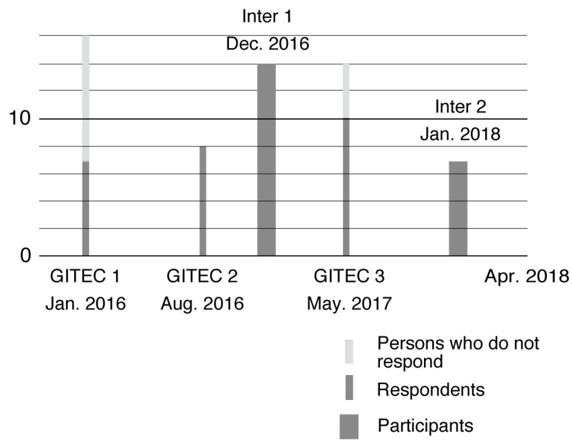


Figure 10.4-1. Respondents of follow-up questionnaire and schedule of education events. Inter means the intermediate level 2 days' course as a subsequent course of the introductory course

The second viewpoint is the response rate of the unsigned follow-up questionnaire to the completed graduates held in April 2018 (Figure 10.4-2). The total response rate was 66%. Although the rate of response might be lower as the longer days gone by since the respondent participated in GITEC, the actual rate of response were as follows.

i. 1st (2016-01, 27 months ago): Response rate 44%
ii. 2nd (2016-08, 20 months ago): Response rate 100%
iii. 3rd (2017-05, 11 months ago): Response rate 74%

Actual response rate from the first graduates was 44%. However, it should be appreciated that this ratio was achieved, despite that the first GITEC was held two years and three months before.

The third viewpoint is whether the knowledge gained by the introductory course is useful or not for their activities. As a whole, 16 persons (64%) in 25 students responded that the knowledge gained in the introductory course are useful, 8 persons (32%) responded somewhat useful. 1 person (4%) responded not so useful. And there was no person responded that the knowledge is useless (Figure 10.4-2). It means that 96% of the respondents answered positively about the usefulness. In other words, 63% (total response rate: 0.66 × positive response rate: 0.96) of the whole completed graduates made a positive evaluation.

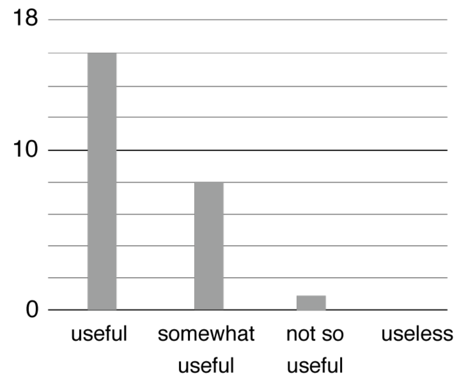


Figure 10.4-2. Usefulness of knowledge obtained by introductory course

In addition, it seems that the evaluation of effectivity is variable over time. However, there is no significant relationship between the time progression and the evaluation of the usefulness, at least by the responses of follow-up questionnaire (Figure 10.4-3).

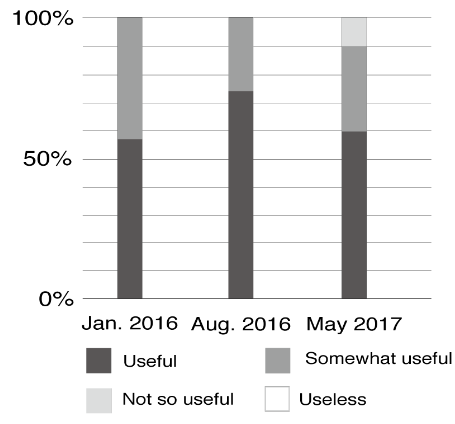


Figure 10.4-3 Relationship between time progression and evaluation of usefulness

At least from the results of evaluations reported in this chapter, we can conclude that introductory course of GITEC has reached the sufficient level for the introduction to GIT.

11. Application to Urban Engineering

11.1. Introduction

GIT is a technology for modeling phenomena occurring in the real world, analyzing models, visualizing analysis results, and leading better decision making. The GI system has been used to solve various problems caused by phenomena appearing in urban areas since the beginning of its history. For example, Howard Fisher, the founder of Harvard University's Laboratory for Computer Graphics and Spatial Analysis was originally an architect and he involved in city planning activities (Fisher 1983).

Meanwhile, in 2008, OGC released CityGML that is an open data model and XML-based format for the storage and exchange of virtual 3D city models. City information model in compliance with CityGML has used in many cities in the world for the construction of 3D city model dataset. Also in Japan, there have been some experiments applying CityGML (Otoi et al. 2014) . On the other hand, since CityGML is not created based on Japanese laws, it is not sufficient for the construction of a system that supports day-to-day urban planning tasks in local public entities. Therefore, we need to model our own application schema for urban planning GIS.

In this chapter, we will try to propose an application schema for the confirmation of frontage requirements, and an application schema to find a buildable extent restricted by the slant line regulation. In this thesis, 'frontage' means a boundary between building site and adjacent open space such as road, path, park and river. These examples indicate that we can expect application schema following GI standards may contribute for urban engineering. However, since these examples may involve bylaw that is published by each local public entity, we do not step into individual cases.

In the meantime, natural disasters in Japan such as typhoons, earthquakes, tsunamis and volcanic explosions occur more frequently than before. In addition, the problems with urban areas are becoming even more complicated in order to maintain a living environment and to keep harmonious coexistence with the natural environment, because the Japanese population is decreasing except for some large cities. Therefore, neighboring local public entities with similar troubles should collaborate toward the overcome against these issues. And, sufficient information sharing should be taken among related organizations.

To achieve these purposes, we should consider standardization of the basic architecture of the business support information system possessed by related local authorities, and we should realize transformability of information between systems managed by them. In 2005, the Ministry of Land, Infrastructure and Transport published the guidance for introducing Urban Planning GIS (UP-GIS) and urged local public entities to refer the common specifications of geospatial data including the results of urban planning basic survey (http://www.mlit.go.jp/crd/tosiko/GISguidance/index.html, in Japanese, accessed 2018-08-14). This guidance has been referred by the local public entities since it was published. However, the recent requirements such as the open data policy, the big data utilization promoted by the government, and to reflect related regulations revised after the publication are not reflected in this guidance. And although this guidance refers to GI standards, the application schema described in UML is not included. Therefore, it is meaningful to show an example that the modeling method in GIT can contribute for the revision of this guidance.

At the next section in this chapter, we will discuss the possibility of the modeling in urban engineering. Then an application schema for evaluating frontage requirements and an application schema for slant line regulation will be introduced. Finally we will discuss requirements on application schema that should be included in the guidance for UP-GIS.

11.2. Possibility of the modeling for urban engineering

According to the ISO 19100 family of standards, any phenomena in the real world should be described as an application schema by applying GFM. As already discussed in above chapters, features and their relationships can be schematized in application schema whether features are physical or virtual. And we can describe the spatial analyses as the operations of features and associations in application schema. Therefore, a model for urban planning should also be able to express as an application schema.

As a matter of fact, there has been developed the city models by applying CityGML (https://www.citygml.org/3dcities/ accessed 2018/07/02) that was described in accordance with GML (ISO/TC 211 2007c) (JISC 2007) (Ishimaru 2014). On the other hand, there are few examples of models for urban planning in accordance with the Japanese laws. In fact, it is difficult to apply the international specifications such as CityGML, because they do not care the Japanese rules. Moreover, the specification for the application system should be considered separately, as GML is the rules for geospatial data interchange.

In that case, for example in Japan, how can we describe urban planning facilities in the city model? From the view point of GIT, urban planning should be a set of operations to realize the new urban facilities in the existing city by analyzing the city model constructed in accordance with the application schema provided by the local public entities. Urban planning facilities in the city model will be urban facilities when they are realized. Most of lands and urban facilities can be considered as they relate each other. For example, a road has horizontal adjacent relationships between city blocks, buildings, parks and so on. A road also has vertical relationships between transportation facilities, underground facilities and geological structures. The stereoscopic urban planning regulation encourages the urban planning system to construct the relationships between roads and the three-dimensional mosaic of regulation districts. Moreover, we need to consider the ecological factors such as air pollution, powder dust, sound noise, solar irradiation, and temperature and humidity distributions. Meanwhile, when the autonomous mobility society is realized, the relevance between road and the automobile, the electric wheelchair, the robot, the small unmanned aerial vehicle, the sensor network, and the charging stand, for safe and efficient transportation will be also required. In the case to consider the roadscape, we need the information about connecting roads and features around the road. They will be not only features located near the road, but also features far from the road such as distant visible mountains. Moreover, to study the historical and cultural characteristics of the urban area, we should describe the relationships between the present features and the historical features already disappeared. These relationships can be represented as a network structure of features in the application schemata. And the user can see the local hierarchical structure by choosing the root feature type. In other words, by describing a network of which nodes are feature types, and which links are feature relationships, it becomes easier to grasp structures centered on individual feature types. However, significant human resource is required to construct the comprehensive city model in accordance with the relevant laws such as City Planning Act and Building Standards Act. Therefore, first of all, it is necessary to confirm such a modeling attempt is technically possible or not.

In this chapter, we will try to introduce an application schema to confirm the frontage requirements, and an application schema to find the buildable surface restricted by the slant line regulation. The first schema should specify all requirements for the frontage. The second schema should fulfill all requirements and propose the outline of algorithm to get stereoscopic surface as the restriction limit of building. Then, we will propose the requirements that should be considered when revising the guidance for UP-GIS published in 2005. Because, after publishing of this guidance, the method of basic surveys concerning city planning has changed in order to consider the urban sustainability under the situation of population decreasing in Japan.

11.3. Application schema for evaluating frontage requirements

11.3.1. Frontage requirements

A building site shall abut a road or an open space in accordance with Building Standards Act (BSA). The requirements for frontage can be sorted as shown in Table 11.3-1. This table was made by the author based on the related acts and regulations.

Table 11.3-1. Frontage requirements sorted by classes of site and abutting space

Building sites	Abutting road and other open spaces	Requirements
Site inside of city planning area or quasi-city planning area BSA 43(1)	Road defined by BSA 42	1
	Open space defined by BSAER 10(2.2.1)	2
	Open space defined by BSAER 10(2.2.2)	3
	Open space defined by BSAER 10(2.2.3)	4
Site outside of city planning area or quasi-city planning area BSA 68(9)		5

BSA: Building Standards Act BSAER: BSA Enforcement Regulation

11.3.2. Procedures for evaluation

According to Table 11.3-1, procedures for evaluation of frontage are described as follows.

Requirement 1 (R1)

Relationship between the building site inside of city planning area or quasi-city planning area and the adjacent road which is defined in BSA 42.

1) Conditions

a. Road is not a road exclusively used for automobile traffic.

b. Road comes under the criteria established by Cabinet Order as having no access of automobiles to the roadside (referred to as “specified elevated roads, etc.” in BSA 44(1.3)).

c. Road lies within areas of District Plan (limited to the areas within those of District Improvement Plan which are designated as areas to be used jointly for buildings and other structures under the provisions of the City Planning Act (CPA) 12(11)).

d. Width of a road is 4m or more, which does not lie within the district given by granted permission with the consent of the Building Review Council held in the Designated Administrative Agency.

e. Width of a road is 6 m or more, which lies within the district given by granted permission with the consent of the Building Review Council held in the Designated Administrative Agency.
f. Width of frontage is 2 m or more.
g. Relationship between site and road is in accordance with the ordinance specified by the local public entity for the restrictions on a site of special buildings, buildings having three or more stories, buildings having habitable rooms with no openings such as windows as specified by Cabinet Order, buildings whose total floor area (or the aggregate of total floor areas if there are two or more buildings on the same site) exceeds 1,000 m^2.
h. Large open space is deemed as an alternative object of a road, if the two criteria are fulfilled. 1) It lies around the site which conforms to criteria specified by Ministry of Land, Infrastructure, Transport and Tourism Order. 2) The Designated Administrative Agency has granted permission based on the consent of the Building Review Council from the viewpoint of traffic, safety, fire protection, or sanitation.

2) Evaluation

If the logical expression below described by propositions above is true, the requirement 1 will be fulfilled.

$((a \wedge b \wedge c \wedge (d \vee e) \wedge f) \wedge g) \vee ((R2 \vee R3 \vee R4) \wedge h)$

R2, R3, and R4 are specified in BSAER as the first criterion of 1) in h. It means that R2, R3, and R4 are sub-requirements of R1. Therefore, this evaluation expression can be seen as an algorithm of frontage evaluation, where the building site is located inside of city planning area or quasi-city planning area.

Requirement 2 (R2)

Relationship between the site inside of city planning area or quasi-city planning area and the open space defined by BSAER 10 (2.2.1).

1) Conditions

a. There is a park, a green belt, a public square or other large open space surrounding such a site. In this context, the large open space is designated in the Urban Parks Act.

2) Evaluation

If the condition is true, the requirement 2 will be fulfilled.

Requirement 3 (R3)

Relationship between the site inside of city planning area or quasi-city planning area and the open space defined by BSAER 10 (2.2.2).

1) Conditions

a. A site is in contact for a distance of not less than 2 m with a farm road or other road that is provided for similar common use (limited to those with a width of 4 m or more). Commonly used roads are designated in the Forest Act, the Land Improvement Act, the Ports and Harbors Act, and publicly used path ('ridou' in Japanese) without the application of laws and regulations nor without registration of private rights.

2) Evaluation

If the condition is true, the requirement 3 will be fulfilled.

Requirement 4 (R4)

Relationship between the site inside of city planning area or quasi-city planning area and the open space defined by BSAER 10 (2.2.3).

1) Conditions

a. A passageway located inside of the site that is wide enough to achieve the goals of evacuation and safe passage and which leads to a road, in accordance with the use, size, location, and construction method of the building.

2) Evaluation

If the condition is true, the requirement 4 will be fulfilled.

Requirement 5 (R5)

Relationship between the site inside of city planning area or quasi-city planning area and the open space defined by BSAER 68 (9).

1) Conditions

a. A site and a building follow the restriction which a local public entity provides to conduct proper and reasonable land use that takes account of the land use of the area concerned.

b. A site and a building follow the restriction, if there is an ordinance which is not be stricter than restrictions under the provisions of BSA 43 to BSA 45.

2) Evaluation

If the site is located outside of city planning area and quasi-city planning area and the logical expression ($a \vee b$) is true, the requirement 5 will be fulfilled.

11.3.3. Application schema for evaluating frontage requirements

In order to realize the evaluation of frontage requirements, it is necessary to describe the association between road and building site with the feature type definitions for road and building site. Figure 11.3-1 shows the application schema for evaluating the frontage requirements. Requirements 1 through 5 are representations for feature associations. Feature Type 'Road' is defined by BSA 42. Ten types of road are specified in BSA 42. According to BSA 42, a road specified in the Road Act is a subtype of a road described in BSA 42, if it is located inside of the city planning area or quasi-city planning area.

The evaluation of frontage requirements will be implemented in the operation 'frontageEvaluation' defined in the feature type 'BuildingSite'. We should examine this application schema as it is a tentative proposal.

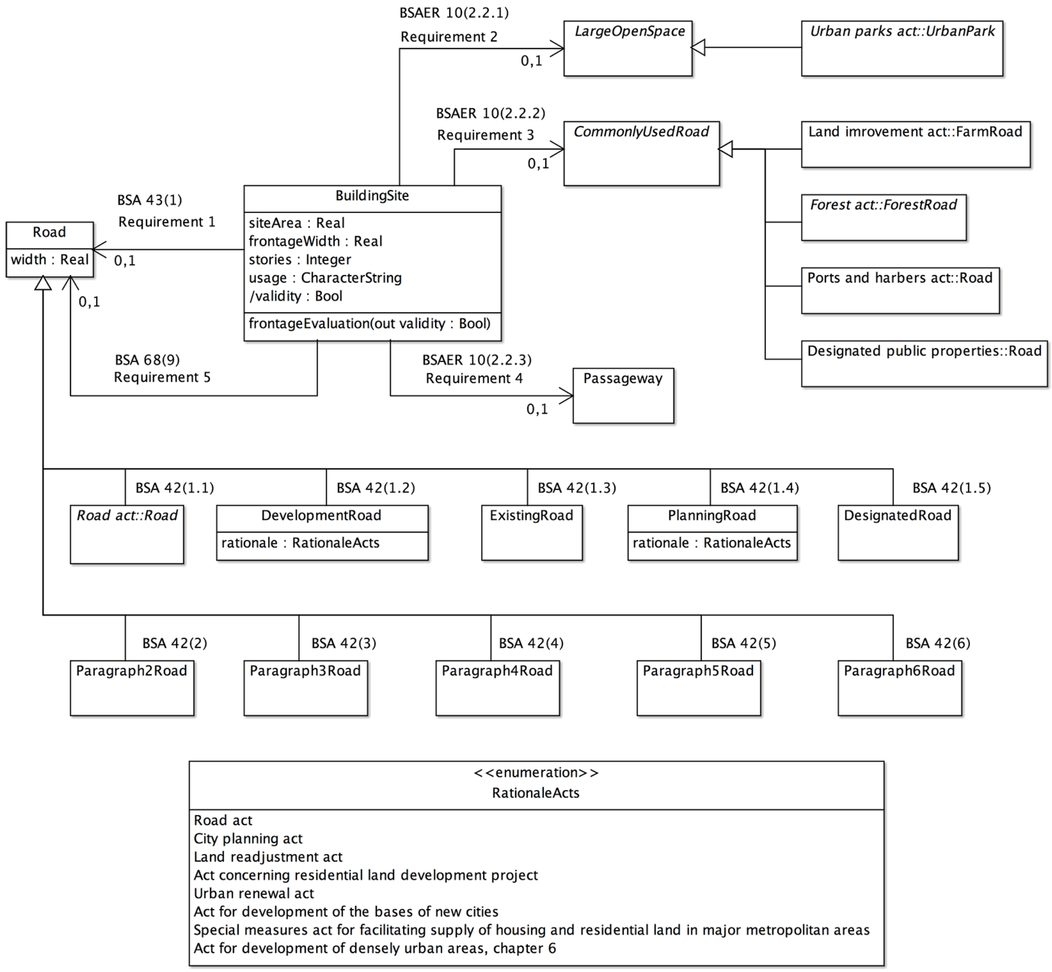


Figure 11.3-1. Application schema for evaluation of frontage requirements.

11.4. Application schema for slant line regulation

11.4.1. Slant line regulation

Height of each part of building is restricted by a slant line generated by the opposite side boundary line of a front road in accordance with BSA 56 (1.1). This restriction is called a slant line regulation in this thesis. A slant line is also determined by the opposite side boundary line of an open space if there is an open space additionally at the opposite side of a front road. Opposite side boundary may change if a front road is a planning road, if road width is wider than 12m, if a road is setbacked inside of the site, and if elevation difference happens between a road and a site. Each slant plane (formed by boundary interval, slant line gradient angle, and applied distance from the building to the boundary) is generated for each zoning, if the building site is separated by more than one zone.

A buildable surface can be generated as a smallest surface by composition of surface segments adjoining each other. Surface segments can be obtained by dividing more than one two-faced polygons (most of them may be a quadrangle) restricting the building shape, which are provided by boundary lines for slant line regulation and a boundary prism of a building site.

11.4.2. Buildable surface composition

Procedure of a building surface composition should be executed in accordance with BSA shown below.

i. Obtain plot ratio (floor area ratio) of a building site by following BSA 52.
ii. Separate the building site to site compartments, if there are more than one zoning in the site.
iii. Assign a front road segment for each compartment, and measure the road width and obtain the centerline of the segment.
iv. Obtain an opposite side boundary interval for slant line regulation by using information shown below.
    - In accordance with BSA Enforcement Order (BSAEO) 134, the open space is selected to get an opposite side boundary interval, if there is an open space in front of the site.
    - In accordance with BSA 56 (2), select the applied distance, if the building will be setbacked.
    - In accordance with BSAEO 132, opposite side boundary is separated by front roads, if there are more than one front road.
    - In accordance with BSAEO 131 (2.2), it is deemed that there is no part of the site in the road, if it is the city planning road. The opposite side boundary is same as the opposite side boundary of the city planning road.
v. In accordance with BSAEO 135 (2), raise the elevation difference between the front road and the site to the opposite side boundary,
vi. Execute the operation to get parameters for slant line regulation (slant line gradient angle, applied distance). This operation is implemented in accordance with “Restrictions on height of each part of a building in connection with front roads” described as BSA Annexed Table 3.
vii. Execute the composition of buildable surface by the operation for slant line regulation (See surfaceComposition() in the feature type ‘BuildingSite’ in Figure 11.4-1).

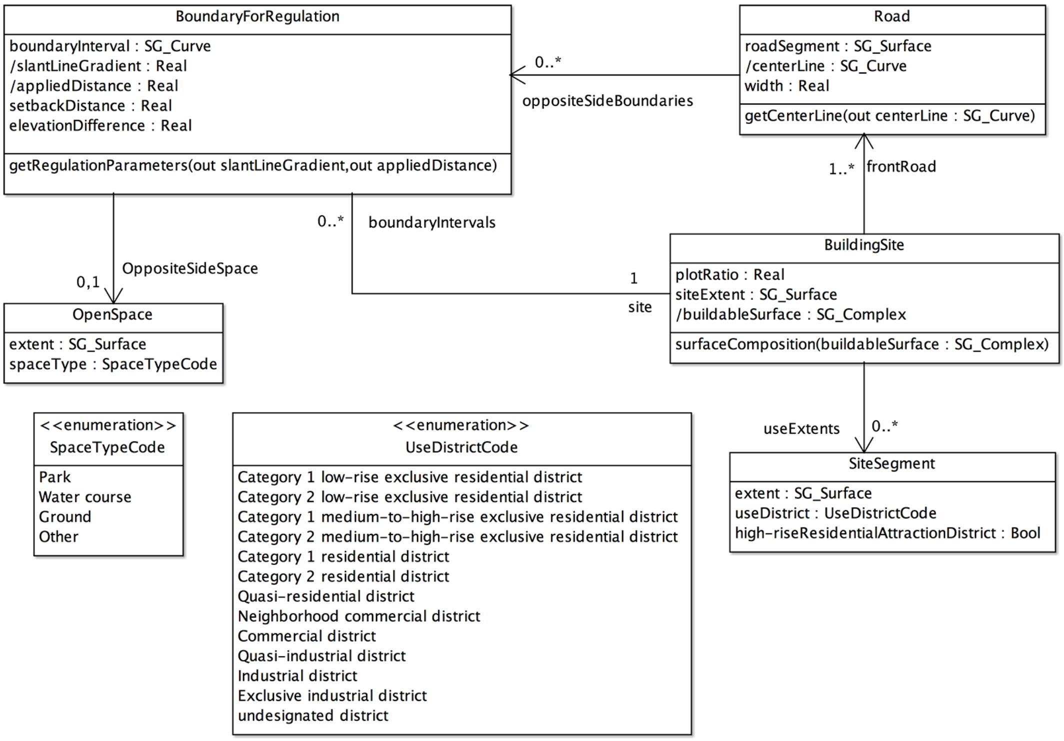


Figure 11.4-1 Application schema for slant line regulation

11.4.3. Application schema for slant line regulation

Figure 11.4-1 illustrates the application schema to get buildable surface in accordance with the slant line regulation as a result of the consideration discussed above. The building site (BuildingSite) adjoins front roads, and they define opposite side boundaries. The operation 'surfaceComposition' constructs buildable surface by applying boundary intervals which are opposite side boundaries created by front roads. If there is an open space at the opposite side of front road, it will be used to define the boundary for regulation. If the building site is separated by more than one use districts, we redefine the building site as a set of site segments. Buildable surface provided by each site segment will finally be combined as one buildable surface for the building site. We should examine this application schema as it is a tentative proposal.

11.5. Requirements on application schema for Urban Planning GIS (UP-GIS)

11.5.1. Introduction

Ministry of Land, Infrastructure, Transport and Tourism (MLIT) published the Guidance for UP-GIS to promote high level usage and dissemination of UP-GIS at local public entities in 2005. This guidance provides the specification proposal of data describing urban planning areas based on urban planning base map and basic surveys concerning city planning.

According to the surveys on UP-GIS at local public entities practiced by the Building Research Institute in 2005 and 2015, 21.3% of prefectural governments were introducing UP-GIS in 2005, and it became 42.6% in 2015 (Sakata 2017). Meanwhile, the maintenance rate of geospatial data in municipalities has been increased steadily. For example, the increasing rate was 66.1% from 2005 to 2015 at municipalities in Shiga prefecture, 51.7% in Okayama prefecture, and 51.2% in Kagawa prefecture. Therefore, the dissemination rate of UP-GIS has been increasing since 2005. However, still it is less than 50%. The Building Research Institute pointed out the reasons such as lack of professional knowledge, cost-effectiveness, and frequency of usage. Meanwhile, in recent years, many types of efforts have been made to eliminate the obstacles, for example, 1) promotion of open data policy, 2) increase in available data, 3) reduction of data acquisition cost, 4) Improvements of research on urban planning, 5) the developing of own application by the officials in municipalities, and 6) appearance of NPOs and volunteers.

Nevertheless, in 2013, the urban planning basic investigation guidance was revised to aim for intensive urban structuring, low carbon city development, central urban area revitalization, and safe and secure city planning and so on. This guidance accentuates that it is effective not only to decide and review individual urban planning but also to utilize objective data provided by the basic survey concerning city planning and its analysis and evaluation results in order to secure the appropriateness of the whole urban planning. In addition, this guidance states that UP-GIS should contribute to efficiency improvement and sophistication of administrative affairs through collaboration between urban planning and other fields. Therefore, a new model for UP-GIS is required under collaboration with various information systems already introduced by local public entities based on social demands and technology development.

11.5.2. Requirements on application schema for UP-GIS

We should consider three issues to improve the current guidance, which are described below.

i. The guidance partly follows the GI standards as tables of data definitions but not UML class diagrams. Current method is difficult to define data types with inheritance relationships clearly, and it is difficult to describe feature associations.

ii. The guidance includes few tabular information such as spreadsheets given by the basic survey. The reason seems to be that the tabular information is not a physical feature. However, it can be described as a virtual feature. Therefore, we should combine tabular information as a virtual feature in the application schema.

iii. More examples to show how to use existing data more effectively should be introduced. Frontage requirements evaluation and slant line regulation could be considered as examples. Furthermore, by adding virtual features in the application schema for UP-GIS, the scope of UP-GIS will be wider.

11.5.3. Example of application schema including virtual feature

Figure 11.5-1 describes a part of application schema for UP-GIS. The purpose of this diagram is mainly to propose association between city planning area and the result of survey on the population in administrative area. ‘Population’ is not a physical feature but it can be understood as a virtual feature associating with viewable features such as administrative district. City planning area is comprised of urban area, urbanizationControl area and non-delineated area. They inherit properties of Zone as an abstract type. Objects of administrative district are transferred from common use spatial dataset, which is a database used in the integrated GIS for municipality administration. By developing UP-GIS based on the application schema with examining the entire item of the basic survey, the latest rule can be reflected in UP-GIS. And data exchange between local public bodies adjacent to city planning areas will become easier, as consistency between geospatial objects is guaranteed by using the same specification described in the guidance.

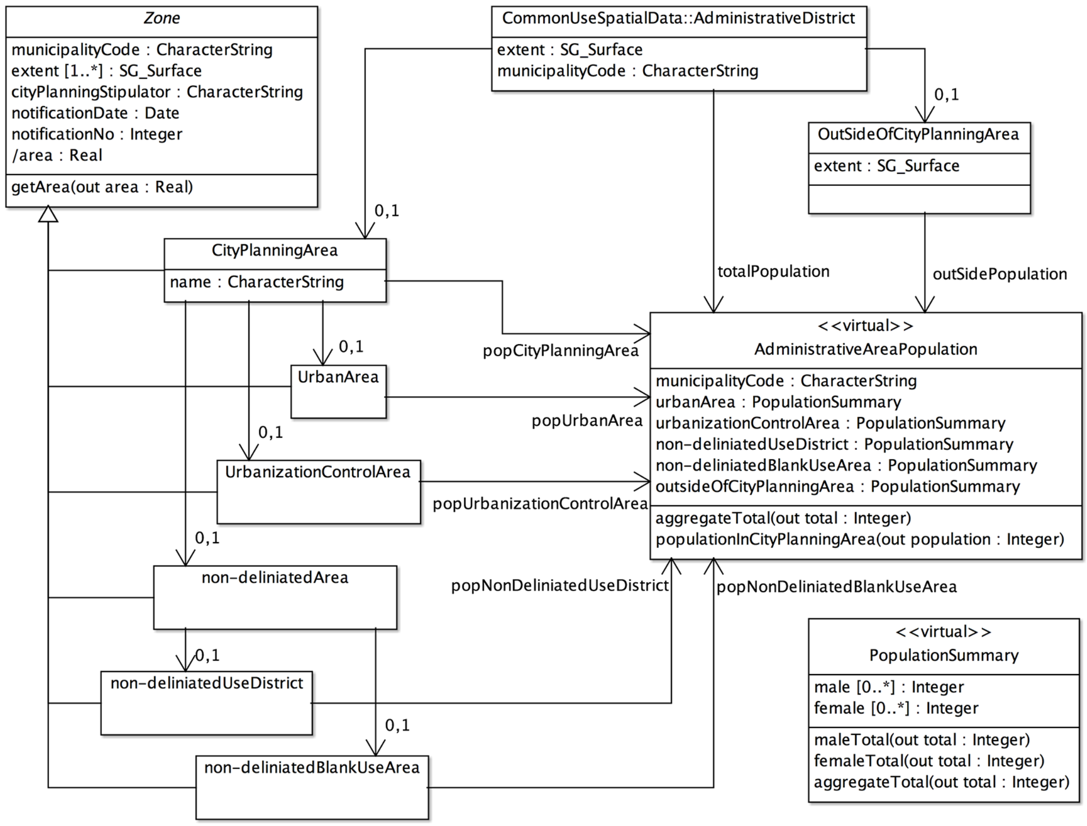


Figure 11.5-1. Example of application schema including city planning area and population in administrative district. Stereotype <<virtual>> means a virtual feature.

12. Conclusions and future works

12.1. Conclusions

In this thesis, the conceptual knowledge structure on GIT that is called 'the concept map of GIT' was proposed. It was modeled in accordance with 1) the four-level metamodel hierarchy, 2) the deep structure and the surface structure for cartographic data, 3) the management structure for geospatial objects with metadata, and 4) GI standards provided by ISO/TC 211. The family of GI standards is used as a basis to keep the consistency between elements in the concept map. The paradigm of GIT is expanded by the adoption of an object-oriented technology. Moreover, it was modeled under the harmonization between different knowledge areas (modeling, acquisition, management, analysis, exchange and representation). Therefore, consistent application development and data acquisition can be expected to realize.

This thesis also proposed the unique GFM and GPM by referring to GI standards. Gittok GFM can be thought as an extension of the geometry-centric layer-based model by introducing the concepts of physical feature type and virtual feature type. Application schema in compliance with gittok GFM may include physical feature types only, virtual feature types only, or a mixture of both. And virtual feature type fills the gap between the deep structure and the surface structure. Therefore, virtual feature type enables the possibility of application schema wider. Nevertheless, gittok GFM can be seen as a profile of ISO GFM simplified so as to be able to learn the concept of GFM in a short period. In addition, gittok GPM as an extended metamodel for GI representation provides the possibility to represent not only maps such as general-purpose map, choropleth map and interactive map, but also gazetteers and interactive lists. List can be seen as a map to represent one-dimensional nominal space as it is used to show virtual features.

Learning assistance tools for 'learning about GIT' such as gittok are quite few, despite that the society needs engineers who have capability to develop GI applications by GIT based on GI standards. The author took charge of the semester course aiming to introduce GI standards at The University of Tokyo for three years since 2008. It was before the development of gittok. However, this experience became motivation to develop the learning assistance tool. And the author conducted the experimental course using gittok at Chuo University for two years since 2014. Then, the author has been conducting the three-day introductory course using gittok that is called GITEC for engineers in the GIS related industry since 2016. These experiences are good opportunities for developing and improving gittok.

We adopted the four level evaluation model (Kirkpatrick 1979) as a validation method of GITEC, because it is a well-known learning evaluation approach especially for educations in business, government, military, and industry alike (Rouse 2011). According to this model, 1) reaction [how learners feel about instruction], 2) learning [learner performance gains on in-class test or not], 3) behavior [learners use the acquired knowledge after the class or not], and 4) results [organizational benefit such as improvement of working performance] are measured. However, the purpose of GIT education is to introduce basic knowledge and encourage students to learn GIT deeper. Therefore, we did not evaluate 'results', as it is out of scope for this education. As a result of the anonymous questionnaire immediately after the course, 63% of the students responded "GITEC is interesting, acquired knowledge is useful, and I want to learn GIT deeper". And 29% of respondents answered "GITEC is interesting and useful" [Reaction]. Next, according to compare the tests conducted before and after the course, a significant gain of knowledge was confirmed by using the t-test (20 points increase in average value, 16.6 points or more for 95% confidence interval) [Learning]. In addition, we conducted the follow-up bearer questionnaires in April 2018 for all students who completed the course since January 2016. The collection rate was 66%, and the respondents answered that the acquired knowledge is useful (64%) or somewhat useful (32%) [Behavior]. According to these results of validation surveys, we may conclude that introductory course using gittok has certain effectivity.

Finally, we proposed three tentative application schemata to introduce that GIT may contribute for urban engineering. Today, as management of urban facilities becomes complicated more, it means that the improvement of urban management systems will be required in near future. In particular, modeling technology is expected to be useful to describe facilities and phenomena in the urban area, and to attack the social problems. We proposed application schema for confirming frontage requirements, and application schema for finding a buildable extent restricted by the slant line regulation. These trials demonstrate the expressiveness of application schema for urban engineering. Then, we proposed the requirements that should be considered when improving the guidance for Urban Planning GIS (UP-GIS). This proposal will be useful for sharing information among different local authorities.

12.2. Future works

Gittok is still under development. In this section, we describe the future issues concerning the concept map of GIT, the development of gittok, the validation of educational efforts, and applications to urban engineering fields.

12.2.1. Issues concerning concept map of GIT

At present, in gittok GFM, spatial attribute included in a feature is limited to two dimensions. Although this is a result of considering the limit of introductory education, three - dimensional spatial attributes should be incorporated in order to enable more advanced education in future. In addition, there is a need to define feature types that maintain temporal attributes to describe dynamic phenomena. Next, we should consider the discussion of DBMS. It is the out of scope in the current concept map of GIT, because most of DBMSs used in GISs are Relational DBMS with the extension to use geometric attribute types defined in the simple feature model. While, gittok GFM is constructed by applying the object-oriented technology that can describe not only physical feature type, but also virtual feature type, feature inheritance and feature association. Furthermore, there are many types of DBMSs such as NoSQL are rapidly progressing today. Therefore, we should carefully think of how to teach DBMS for GIT, despite that the discussion on DBMS should be included in the advanced education.

Meanwhile, we should improve the geo-library in gittok that can maintain more diverse types of metadata such as metadata for schemata, for cartographic products, for lists, and for other geo-library services. Studies should be made to enable seamless information retrieval by metadata conforming to the general metadata model.

Although we can obtain many datasets from the open data services today, we should be careful to use geospatial objects without quality evaluation results. In this thesis, the evaluation of positional accuracy regarding map registration was introduced as an example to show what data quality is. However, we should have a mechanism to measure and evaluate various types of data quality in future.

12.2.2. Issues concerning development of gittok

Currently data interchange between gittok and other applications is limited, because gittok adopts object-oriented model that follows GFM for exchange objects. Meanwhile, gittok enables to output SVG files and GeoJSON files for mapping of physical features. It would be possible to provide the capability to convert and embed commonly used legacy data in gittok in future.

Gittok is not made for processing of large amounts of data as it is not for real projects. However in the advanced education, implementation of the data structure for high speed data retrieval will be required.

12.2.3. Issues concerning educational efforts

It is nearly impossible to compare the effectivity of the education using gittok with other education courses having the same purpose. Because, at least by the Internet research, few similar courses and few similar applications were found. This was the reason to apply the four level evaluation model. However we did not evaluate the ‘results’ such as improvement of working performance, as the level of education using gittok is the introductory level. On the other hand, the scope of education to which gittok can be applied has already exceeded the period of semester. The course design should be rearranged by, for example, separating the content which is suitable for the introductory course from that which would be better suited to an advanced course. In addition, a comprehensive curriculum covering from introductory level to advanced level on GIT should be developed in future. If engineers can learn GIT by using such a curriculum in the department of Geo-ICT, it will be able to evaluate the profit given to the organization to which they belongs. In addition, the possibility of offering a massive open online course (MOOC) and a better means for realizing active learning should be investigated given changes in social demand.

12.2.4. Issues concerning applications to urban engineering fields

Three examples of application schemata shown in this thesis are related to day-to-day administrative practice and collaboration among local authorities. In Japan, to respond to changes in the social environment accompanying the declining population and to establish a society of high resiliency to respond to natural disasters are urgent issues. To attack such problems, GIT will be an effective means. The collaboration of experts in related fields and engineers having a skill of GIT will be required to create the vision, the proposal and the system to solve those problems.

Bibliography

Ahearn, Sean C. 2008. "Object Orientation (OO)." *Encyclopedia of Geographic Information Science*. SAGE Publication.

Akeno, Kazuhiko, Shoji Okuyama, and Shinji Takazawa. 2005. "Japan: 日本メタデータプロファイル; Nihon Metadata Purofairu; Japanese Metadata Profile." In *World Spatial Metadata Standards*, edited by Harold Moellering. Oxford: Elsevier.

Andrews, J. 1996. "What Was a Map? The Lexicographers Reply." *Cartographica: The International Journal for Geographic Information and Geovisualization* 33 (4): 1–12. doi:10.3138/NJ8V-8514-871T-221K.

APA. 2005. *JIS X 7115 - Geographic Information- Metadata*. Tokyo: Japanese Standards Association. https://webdesk.jsa.or.jp/books/W11M0070/index.

Aronoff, Stan. 1989. *Geographic Information Systems: A Management Perspective*. Ottawa: WDL Publications.

Asami, Yasushi. 2011. "Message from Editor." *Procedia - Social and Behavioral Sciences* 21: 1–2. doi:10.1016/j.sbspro.2011.07.001.

Bertin, Jacques. 1983. *Semiology of Graphics: Diagrams, Networks, Maps*. The University of Wisconsin Press.

Bolstad, Paul. 2016. *GIS Fundamentals - The First Text on Geographic Information Systems*. 5th editio. XanEdu Publishing Inc.

Bruce, Kim B. 2002. *Fundations of Object-Oriented Languages: Types and Semantics*. MIT Press.

Burrough, Peter A. 1982. *Principles of Geographical Information Systems for Land Resources Assessment*. Kindle ver. Prentice Hall PTR.

Butler, H., M. Daly, A. Doyle, S. Gillies, T. Schaub, and S. Hagen. 2015. "The GeoJSON Format." *IETF*. https://tools.ietf.org/html/draft-butler-geojson-06.

CC_in_GIS, and NCGIA. 1990. "Unit 28 - Affine and Curvilinear Transformations, NCGIA GIS Core Curriculum." https://escholarship.org/uc/item/32w746gn.

Chance, Arthur, Richard Newell, and David Theriault. 1990. "An Object-Oriented GIS - Issues and Solutions." In , 1:179–88. Amsterdam: EGIS Foundation. https://www.oicrf.org/-/egis-90-first-european-conference-on-geographical-information-systems.

Chomsky, Noam. 1968. *Language and Mind*. Harcourt Brace & World, Inc.

Chrisman, Nicholas. 2002. *Exploring Geographic Information Systems*. Second edi. John Wiley & Sons.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. "Floyd-Warshall Algorithm." In *Introduction to Algorithms*, 3rd ed., 693–700. Cambridge, Massachusetts.

DGIWG. 2013. *STD-13-014-NATO Geospatial Information Model (NGIM) Edition 1.0*. DGIWG.

DiBiase, David, Tripp Corbin, Thomas Fox, Joe Francica, Kass Green, Janet Jackson, Gary Jeffress, et al. 2010. "The New Geospatial Technology Competency Model: Bringing Workforce Needs into Focus." *URISA Journal* 22 (2): 55–72. http://www.scopus.com/inward/record.url?eid=2-s2.0-78650620998&partnerID=tZOtx3y1.

DiBiase, David, Michael DeMers, Ann Johnson, Karen K. Kemp, Ann Taylor Luck, Brandon Plewe, and Elizabeth Wentz. 2006. *Geographic Information Science and Technology Body of Knowledge*. *UCGIS*. http://www.worldcat.org/title/geographic-information-science-and-technology-body-of-knowledge/oclc/75414829&referer=brief_results.

DiBiase, David, Michael DeMers, Ann Johnson, Karen K. Kemp, Ann Taylor Luck, Brandon Plewe, and ELizabeth Wentz. 2007. "Introducing the First Edition of Geographic Information Science and Technology Body of Knowledge." *Cartography and Geographic Information Science* 34 (2): 120.

Dibiase, David, Michael Demers, Ann Johnson, Karen Kemp, Ann Taylor Luck, Brandon Plewe, and Elizabeth Wentz, eds. 2006. *Geographic Information Science and Technology Body of Knowledge First Edition*. Association of American Geographers.

Duckham, Matt, Michael F. Goodchild, and Michael F. Worboys. 2003. *FOUNDATIONS OF GEOGRAPHIC INFORMATION SCIENCES*. doi:10.1016/B978-0-444-59541-6.09991-4.

Eisenberg, J. D., and A. Bellamy-Royds. 2014. *SVG Essentials: Producing Scalable Vector Graphics with XML*. O'Reilly Media, Inc.

EU. 2007. "Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 Establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)." *Official Journal of the European Union* 50 (January 2006): 1–14. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:EN:PDF.

Fisher, Howard. 1983. *Mapping Information: The Graphic Display of Quantitative Information*. Cambridge, Massachusetts: ABT Books.

Foresman, Timothy W. 1998. "GIS Early Years and the Threads of Evolution." In *The History of Geographic Information Systems: Perspectives from the Pioneers*, edited by Timothy W. Foresman, 3–17. Upper Saddle River, NJ: Prentice Hall PTR.

Goodchild, Michael F. 1992. "Geographical Information Science." *International Journal of Geographical Information Science* 6 (1): 31–45.

———. 2007. "Citizens as Sensors: The World of Volunteered Geography." *GeoJournal* 69: 211–21.

doi:10.1007/s10708-007-9111-y.

Goodchild, Michael F., and Karen K. Kemp. 1990. “NCGIA GIS Core Curriculum.” *NCGIA*. http://ibis.geog.ubc.ca/courses/klink/gis.notes/ncgia/toc.html.

Graser, Anita. 2013. *Learning QGIS*. 2.0. Olton, UK: Packt Publishing Ltd,.

Greenberg, Jane. 2004. “Metadata Extraction and Harvesting: A Comparison of Two Automatic Metadata Generation Applications.” *Journal of Internet Cataloging* 6 (4): 59–82. doi:10.1300/J141v06n04_05.

Gröger, Gerhard, Thomas H Kolbe, Claus Nagel, and Karl-Heinz Häfele. 2012. *Open Geospatial Consortium OGC City Geography Markup Language (CityGML) En- Coding Standard*. Open Geospaial Consortium. http://www.opengeospatial.org/legal/.

Hansen, Sean, Nicholas Berente, and Kalle Lyytinen. 2009. “Requirements in the 21st Century: Current Practice and Emerging Trends.” *Lecture Notes in Business Information Processing* 14 LNBIP: 44–87. doi:10.1007/978-3-540-92966-6_3.

Hatayama, Michinori, Fumitoshi Matsuno, Shigeru Kakumoto, and Hiroyuki Kameda. 1999. “時空間地理情報システムDiMSISの開発 Development of Spatial Temporal Information System DiMSIS.” *Theory and Applications of GIS* 7 (2): 25–33.

Herring, John R. 1992. “TIGRIS: A Data Model for an Object-Oriented Geographic Information System.” *Computers & Geosciences* 18 (4): 443–52. doi:10.1016/0098-3004(92)90074-2.

Heywood, I., S. Cornelius, and S Carver. 2011. *An Introduction to Geographical Information Systems*. England: Pearson Education.

ICA. 2015. “Directory 2015-2019.” http://icaci.org/files/documents/reference_docs/2015-2019_directory.pdf.

IHO. 2017. *S-100 – UNIVERSAL HYDROGRAPHIC DATA MODEL*. 3.0.0. International Hydrographic Organization.

Illert, Andreas, and Sabine Afflerbach. 2007. *Data Product Specifications*. EuroGraphics. http://www.eurogeographics.org/documents/RISE25DataProductSpecificationsV1.0.pdf.

Ishimaru, Nobuhiro. 2014. “3 次元地理空間データ CityGML/IndoorGML に関する国際標準化活動 International Activity on CityGML / IndoorGML for 3D Geospatial Data.” *Map* 52 (3): 29–36.

ISO/IEC. 2011. *ISO/IEC 25010:2011 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models*. Geneva: International Organization for Standardization and International Electrotechnical Commission.

ISO/TC 211. 2003a. *ISO 19107:2003 - Geographic Information - Spatial Schema*. Geneva: International Organization for Standardization.

———. 2003b. *ISO 19112:2003 - Geogtraphic Information - Spatial Referencing by Geographic Identifiers*. Geneva: International Hydrographic Organization.

———. 2005. *ISO 19109:2005 - Geographic Information - Rules for Application Schema*. Geneva: International Organization for Standardization.

———. 2007a. *ISO 19111:2007 - Geographic Information - Spatial Referencing by Coordinates*. Geneva: International Organization for Standardization. https://www.iso.org/standard/41126.html.

———. 2007b. *ISO 19131:2007 - Geographic Information - Data Product Specifications*. Geneva: International Organization for Standardization.

———. 2007c. *ISO 19136:2007 - Geographic Information - Geography Markup Language (GML)*. Geneva: International Organization for Standardization. https://www.iso.org/standard/32554.html.

———. 2009. *Standards Guide Iso / Tc 211 Geographic Information / Geomatics*. Geneva: International Organization for Standardization. http://www.isotc211.org/Outreach/ISO_TC_211_Standards_Guide.pdf.

———. 2011a. *ISO 14825:2011 - Intelligent Transport Systems - Geographic Data Files (GDF) - GDF5.0*. Geneva: International Organization for Standardization. https://www.iso.org/obp/ui/#iso:std:iso:14825:ed-2:v1:en.

———. 2011b. *ISO 19118:2011 - Geographic Information - Encoding*. Geneva: International Organization for Standardization.

———. 2012a. *ISO 19117:2012 - Geographic Information - Portrayal*. Geneva: International Organization for Standardization.

———. 2012b. *ISO 19155:2012 - Geographic Information - Place Identifiers (PI)*. Geneva: International Organization for Standardization. https://www.iso.org/standard/32573.html.

———. 2013. *ISO 19157:2013 - Geographic Information - Data Quality*. Geneva: International Organization for Standardization. https://www.iso.org/standard/32575.html.

———. 2014a. *ISO 19101-1:2014 - Geographic Information - Reference Model - Part 1: Fundamentals*. Geneva: International Organization for Standardization. https://www.iso.org/standard/59164.html.

———. 2014b. *ISO 19115-1:2014 - Geographic Information - Metadata*. Geneva: International Organization for Standardization.

———. 2015a. *ISO 19103:2015 - Geographic Information - Conceptual Schema Language*. Geneva: International Organization for Standardization.

———. 2015b. *ISO 19109:2015 – Geographic Information – Rules for Application Schema*. Geneva: International Organization for Standardization.

JISC. 2007. *JIS X 7136:2007 Geographic Information GML*. japanese Industrial Standards Committee.

Kameda, Takayuki, and Keiko Imai. 2003. “Map Label Placement for Points and Curves.” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E86–A (4): 835–40.

Kawabata, Mizuki, Rajesh Bahadur Thapa, Takashi Oguchi, and Ming Hsiang Tsou. 2010. “Multidisciplinary Cooperation in GIS Education: A Case Study of US Colleges and Universities.” *Journal of Geography in Higher Education* 34 (4): 493–509. doi:10.1080/03098265.2010.486896.

Kawase, Kazushige. 2011. “A More Concise Method of Calculation for the Coordinate Conversion between Geographic and Plane Rec- Tangular Coordinates on the Gauss-Krüger Projection. (in Japanese).” *Journal of the Geospatial Information Authority of Japan* 121: 109–24. http://www.gsi.go.jp/common/000061216.pdf.

Kemp, Karen K., ed. 2008. *Encyclopedia of Geographic Information Science*. SAGE Publication. https://uk.sagepub.com/en-gb/asi/encyclopedia-of-geographic-information-science/book226890.

Kirkpatrick, Donald L. 1979. “Techniques for Evaluating Training Programs.” *Training and Development Journal* June: 78–92.

Kraak, Menno-Jan, and Ferjan Ormeling. 2013. *Cartography: Visualization of Spatial Data*. Routledge.

Kresse, Wolfgang, and David M. Danko, eds. 2012. *Springer Handbook of Geographic Information*. Springer-Verlag. http://www.springer.com/gp/book/9783540726784.

Kresse, Wolfgang, David M. Danko, and Kian Fadie. 2011. “Standardization.” In *Springer Handbook of Geohaphic Information*, edited by Wolfgang Kresse and David M. Danko. Springer-Verlag.

Kubo, Sachio et al. 1993. *NCGIA GIS Core Curriculum (Japanese Edition)*. Keio University.

Kubota, Koichi. 2015. “Network Analysis.” In *Geographic Information Sciences - GIS Standard (in Japanese)*, edited by Yasushi Asami, Keiji Yano, Yukio Sadahiro, and Minori Yuda, 70–78. Tokyo: Kokon Shoin.

Ledermann, Florian, and Georg Gartner. 2015. “Mapmap.Js: A Data-Driven Web Mapping API for Thematic Cartography.” In *Proceedings for 27th International Cartographic Conference*. Rio de Janeiro: International cartographers association (ICA). http://www.icc2015.org/trabalhos/6/572/T6-572_1430403349.pdf.

Lee, Jiyeong, Ki-Joune Li, Sisi Zlatanova, Thomas H. Kolbe, Claus Nagel, and Thomas Becker. 2018. *OGC InddorGML: Corrigendum*. *Open Geospatial Consortium*. Open Geospaial Consortium. http://www.opengeospatial.org/.

Longlay, Paul A., Michael F. Goodchild, David J. Maguire, and David W. Rhind. 2015. *Geographic Information Science and Systems 4th Edition*. John Wiley & Sons.

Manso-Callejo, Miguel, Mónica Wachowicz, and Miguel Bernabé-Poveda. 2010. “The Design of an

Automated Workflow for Metadata Generation." In *Communications in Computer and Information Science*, 108 CCIS:275–87. doi:10.1007/978-3-642-16552-8_25.

Masaharu, Hiroshi. 2017. "Development in Understanding of Gauss-Krüger Projection and Its Outcomes." In *Proceedings of the 28th International Cartographic Conference*. Washington D.C. http://www.eventscribe.com//2017/ICC/assets/handouts/419529.pdf.

Masser, Ian. 2008. "Spatial Data Infrastructure." In *Encyclopedia of Geographic Information Science*. SAGE Publication.

Moellering, Harold. 1980. "Strategies of Real-Time Cartography." *The Cartographic Journal* 17 (1). Taylor & Francis: 12–15. doi:10.1179/caj.1980.17.1.12.

Moellering, Harold, H.J. Aalders, and Aaron Crane, eds. 2005. *World Spatial Metadata Standards: Scientific and Technical Descriptions, and Full Descriptions with Crosstable*. Elsevier. https://www.elsevier.com/books/world-spatial-metadata-standards/moellering/978-0-08-043949-5.

Murakami, Hiroshi. 2008. "New Legislation on NSDI in Japan: ' Basic Act on the Advancement of Utilizing Geospatial Information.'" *Bulletin of the Geographical Survey Institute* 55 (March): 1–10.

Murray, Scott. 2013. *Interactive Data Visualization for the Web*. O'Reilly Media, Inc.

Nakamura, Hideo, and Hidenori Shimizu. 2000. *測量学 Surveying*. Tokyo: Gihoudo.

Nasser, Hussein. 2014. *Learning ArcGIS Geodatabases*. Olton, UK: Packt Publishing Ltd,.

Norman, Donald A. 2011. *Living with Complexity*. Cambridge, Massachusetts: MIT Press. https://mitpress.mit.edu/books/living-complexity.

NRC. 1999. *Distributed Geolibraries – Spatial Information Resources*. Washington D.C.: National Academic Press. http://www.nap.edu/catalog/9460.html.

Nyerges, Timothy L. 1991. "Analytical Map Use." *Cartography and Geographic Information Systems* 18 (1). Taylor & Francis: 11–22. doi:10.1559/152304091783805635.

Okabe, Atsuyuki. 2006. "地理情報科学の教育と地理学 Education for Geographic Information Science and Geography." *E-Journal GEO* 1 (1): 67–74. https://www.jstage.jst.go.jp/article/ejgeo/1/1/1_1_67/_pdf.

Okabe, Atsuyuki, Yasushi Asami, Takashi Oguchi, Masatoshi Arikawa, Hiroyuki Kohsaka, Yuji Murayama, Keiji Yano, Keiichi Okunuki, Mizuki Kawabata, and Teruko Usui. 2008. *Studies on the Development of Standard Geographical Information Science Curricula and That of the Sustainable-Collaborative Web Library Systems for Serving Their Contents*. Edited by Atsuyuki Okabe. Japan Society for the Promotion of Science. https://kaken.nii.ac.jp/en/grant/KAKENHI-PROJECT-17200052/.

Olfat, Hamed, Mohsen Kalantari, Abbas Rajabifard, Hervé Senot, and Ian P. Williamson. 2013. "A

GML-Based Approach to Automate Spatial Metadata Updating." *International Journal of Geographical Information Science* 27 (2): 231–50. doi:10.1080/13658816.2012.678853.

OMG. 2003. *UML 2.0 Infrastructure Specification*.

Ormsby, T, E Napoleon, R Burke, C Groessl, and L Bowden. 2010. *Getting to Know ArcGIS Desktop*. Redlands: Esri Press.

Ota, Morishige. 2014. "地理空間情報技術の学習支援ツールの設計と開発 Design and Development of an Assistance Tool for Education on the Geospatial Information Technology." *Theory and Applications of GIS* 22 (2): 13–23. http://www.gisa-japan.org/dl/22-2PDF/22-2-13.pdf.

———. 2016. "拡張型の一般描画モデルを適用した地理情報表現 Extended General Portrayal Model for Geographic Information Representation." *"Map", Journal of the Japan Cartographers Association* 54 (2): 1–16.

———. 2017. "Conceptual Modeling and Its Implementation in an Education Assistance Software Tool for Geographic Information Technology." *International Journal of Cartography* 3 (2). Taylor & Francis: 201–25. doi:10.1080/23729333.2017.1308693.

Ota, Morishige, and Reese Plews. 2015. "Development of a Software Tool as an Introduction to Geospatial Information Technology Based on Geospatial Standards." *Cartography and Geographic Information Science* 42 (5): 419–34. doi:10.1080/15230406.2015.1031701.

Otoi, Kosei, Izumi Kamiya, Mamoru Koarai, and Takayuki Nakano. 2014. "Three Dimensional Indoor GIS Dataset Made from Design Drawings , and Its Problems." *Bulletin of the Geographical Survey Institute*, no. 125. http://www.gsi.go.jp/common/000092004.pdf.

Painho, Marco, Paula Curvelo, Ignacio Jovani, Sara I. Fabrikant, and Monica Wachowicz. 2007. "An Ontological-Based Approach to Geographic Information Science Curricula Design." In *The European Information Society*, edited by Sara I. et al Fabrikant, 1:15–34. Berlin Heidelberg: Springer-Verlag.

Pilone, Dan, and Neil Pitman. 2005. *UML 2.0 in a Nutshell: A Desktop Quick Reference*. O'Reilly.

Plessis, Heindrich Du, and Adriaan Van Niekerk. 2014. "A New GISc Framework and Competency Set for Curricula Development at South African Universities." *South African Journal of Geomatics* 3 (1): 1–12. http://www.ajol.info/index.php/sajg/article/download/106127/96098.

Portele, Clemens. 2011. "Encoding of Geographic Information." In *Springer Handbook of Geohaphic Information*, edited by Wolfgang Kresse and David M. Danko. Springer-Verg. doi:10.1007/ 978-3-540-72680-7_4.

Raper, Jonathan, and Nick Green. 1992. "Teaching the Principles of GIS: Lessons from the GISTutor Project." *International Journal of Geographical Information Systems* 6 (4): 279–90.

doi:10.1080/02693799208901912.

Reinhardt, Wolfgang. 2005. "Teaching GI Standards in Higher Education - Examples from Distance and Classroom Teaching." In *Proceedings of the 22nd International Cartographic Conference : Mapping Approaches into a Changing World*. A Coruña, Spain.

Rouse, Donald Nick. 2011. "Employing Kirkpatrick's Evaluation Framework to Determine the Effectiveness of Health Information Management Courses and Programs." *Perspectives in Health Information Management* 8 (Spring): 1–5.

Ruegg, Walter. 2003. *A History of the University in Europe: Volume 1, Universities in the Middle Ages*. Cambridge University Press.

Runco, Mark A., and Ivonne Chand. 1995. "Cognition and Creativity." *Educational Psychology Review* 7 (3): 243–67. doi:10.1007/BF02213373.

Sakabe, Kouichi. 2016. "グラフィックデザイン用語英和辞典 English-Japanese Dictionary for Graphic Design Terms." Fishtail studio. http://www.fishtailstudio.com/pages/gd.html.

Sakata, Tomohiko. 2017. "エスノグラフィー調査に基づく自治体での都市計画GISの持続的利活用に関する研究 Study on Sustainable Utilization of Urban Planning GIS in Local Governments." *NII*. https://kaken.nii.ac.jp/ja/grant/KAKENHI-PROJECT-26420638/.

Sasaki, Midori, Takashi Oguchi, Atsuyuki Okabe, and Yukio Sadahiro. 2008. "GIS Education of Geographical Departments in Japanese Universities in Relation to the Japan Standard GIS Core Curriculum." *International Research in Geographical and Environmental Education* 17 (4): 298–306. doi:10.1080/10382040802401540.

Schulze, Uwe, Detlef Kanwischer, and Christoph Reudenbach. 2013. "Essential Competences for GIS Learning in Higher Education: A Synthesis of International Curricular Documents in the GIS&T Domain." *Journal of Geography in Higher Education* 37 (2): 257–75. doi:10.1080/03098265.2012.763162.

Sui, Daniel Z. 1995. "A Pedagogic Framework to Link GIS to the Intellectual Core of Geography." *Journal of Geography* 94 (6): 578–91.

Tani, Kenji. 2009. "時系列地形図閲覧ソフト『今昔マップ2』(首都圏編・中京圏編・京阪神圏編)の開発 Development of a Time-Series Topographic Map Viewer 'Konjyaku Map 2': Capital, Chukyo and Keihanshin Area Edition." *Theory and Applications of GIS* 17 (2): 1–10.

Tobler, W. R. 1979. "A Transformational View of Cartography." *The American Cartographer* 6 (2): 101–6. doi:10.1559/152304079784023104.

Tyner, Judith A. 2010. *Principles of Map Design*. New York: Guilford Press.

U.S.Government. 1994. "Executive Order 12906: Coordinating Geographic Data Acquisition and Access:

The National Spatial Data Infrastructure." *Federal Register* 59 (71). https://www.archives.gov/files/federal-register/executive-orders/pdf/12906.pdf.

Unwin, David J., Kenneth E. Foote, Nicholas J. Tate, and David DiBiase, eds. 2012. *Teaching Geographic Information Science and Technology in Higher Education*. Wiley-Blacwell.

Vancauwenberghe, Glenn, and Danny Vandenbroucke. 2015. "LINKVIT as Part of the GIS&T Education and Training Landscape in Europe." *Iuav Giornale Dell' Università* 146 (1391): 3.

WMO. 2008. "DEVELOPING A PROPOSAL FOR A CBS POLICY ON DATA REPRESENTATION SYSTEMS." Washington D.C.

Worboys, Michael F. 1995. *GIS: A Computing Perspective*. London: Taylor & Francis.

Yuda, Minori. 2011. "Effectiveness of Digital Educational Materials for Developing Spatial Thinking of Elementary School Students." *Procedia - Social and Behavioral Sciences* 21: 116–19. doi:10.1016/j.sbspro.2011.07.045.