GEOINTERPRET: AN ONTOLOGICAL ENGINEERING METHODOLOGY FOR AUTOMATED INTERPRETATION OF GEOSPATIAL QUERIES

by

Ratchata Peachavanish

B.S., University of Washington, 1995

M.S., Syracuse University, 1999

Submitted to the Graduate Faculty of

the School of Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2004

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Ratchata Peachavanish

It was defended on

November 30, 2004

and approved by

Dr. Stephen C. Hirtle, Professor, Information Science and Telecommunications

Dr. Michael Lewis, Associate Professor, Information Science and Telecommunications

Dr. Jeen-Shang Lin, Associate Professor, Civil and Environmental Engineering

Dr. Burcu Akinci, Assistant Professor, Civil and Environmental Engineering, Carnegie Mellon University

Dr. Hassan A. Karimi, Assistant Professor, Information Science and Telecommunications
Dissertation Director

GEOINTERPRET: AN ONTOLOGICAL ENGINEERING METHODOLOGY

FOR AUTOMATED INTERPRETATION OF GEOSPATIAL QUERIES

Ratchata Peachavanish, Ph.D.

University of Pittsburgh, 2004

Despite advances in GIS technology, solving geospatial problems using current GIS platforms involves complex tasks requiring specialized skills and knowledge that are attainable through formal training and experience in implementing GIS projects. These requisite skills and knowledge include: understanding domain-specific geospatial problems; understanding GIS representation of real-world objects, concepts, and activities; knowing how to identify, locate, retrieve, and integrate geospatial data sets into GIS projects; knowing specific geoprocessing capabilities available on specific GIS platforms; and skills in utilizing geoprocessing tools in GIS with appropriate data sets to solve problems effectively and efficiently. Users interested in solving application-domain problems often lack such skills and knowledge and resort to GIS experts (this is especially true for applications dealing with diverse geospatial data sets and complex problems). Therefore, there is a gap between users' knowledge about geoprocessing and GIS tools and the GIS knowledge and skills needed to solve geospatial problems. To fill this gap, a new approach that automates the tasks involved in geospatial problem solving is needed. Of these tasks, the most important is geospatial query (usually expressed in application-specific concepts and terminologies) interpretation and mapping to geoprocessing operations implementable by GIS. The goal of this research is to develop an ontological engineering methodology, called GeoInterpret, to automate the task of geospatial query interpretation and mapping. This methodology encompasses: a conceptualization of geospatial queries; a multiple-ontology approach for representing knowledge needed to solve geospatial queries; a set of techniques for mapping elements between different ontologies; and a set of algorithms for geospatial query interpretation, mapping, and geoprocessing workflow composition. A proof of concept was developed to demonstrate the working of GeoInterpret.

TABLE OF CONTENTS

PREFACE......XI

1.0 INTRODUCTION 1

1.1 PRELIMINARIES...... 1

1.2 RESEARCH PROBLEM, CHALLENGES, AND SIGNIFICANCE 2

1.3 RESEARCH OBJECTIVES AND METHODOLOGY 6

1.4 CONTRIBUTIONS 8

1.5 OUTLINE OF THE DISSERTATION...... 8

2.0 RELATED LITERATURE...... 9

2.1 GEOSPATIAL INFORMATION SYSTEMS 9

2.1.1 Definitions 10

2.1.2 Historical Background 11

2.1.3 Components of GIS 14

2.1.4 Conceptual Models for Geospatial Objects and Spaces...... 16

2.1.5 Computer Representation Models 17

2.1.6 Coordinate Reference Systems 20

2.1.7 Geoprocessing...... 22

2.2 GIS INTEROPERABILITY AND STANDARDS...... 27

2.2.1 Information Heterogeneity...... 28

2.2.2 Interoperability Standards...... 29

2.3 ONTOLOGIES AND GIS 32

2.3.1 Ontologies...... 33

2.3.2 Types of Ontologies...... 36

2.3.3 Conceptual Framework...... 38

2.3.4 Geospatial Cognition 41

2.4 SUMMARY...... 42

3.0 GEOSPATIAL QUERIES 43

3.1 FORMULATION OF GEOSPATIAL QUERIES 43
3.1.1 Phases of Geospatial Query Formulation 43
3.1.2 Formulating Queries with GIS 45
3.2 CONCEPTUALIZATION OF GEOSPATIAL QUERIES 47
3.2.1 Query Differences Across Application Domains 47
3.2.2 Common Characteristics of Queries Across Application Domains 49
3.3 SUMMARY 54

4.0 GEOINTERPRET METHODOLOGY 55
4.1 INTRODUCTION 55
4.1.1 Conceptual Approach 55
4.1.2 Challenges 58
4.1.3 Scope 59
4.2 COMPONENTS OF GEOINTERPRET 59
4.3 GEOSPATIAL DOMAIN ONTOLOGY 61
4.3.1 Entities 62
4.3.2 Actions, Roles, and Affordance 62
4.3.3 Representation of Geospatial Queries 65
4.4 GEOSPATIAL PROCESSING ONTOLOGY 68
4.4.1 Geospatial Data Types and Geoprocessing Operators 68
4.4.2 Geospatial Data Sets in Geoprocessing 69
4.5 ONTOLOGICAL MEDIATOR 70
4.5.1 MetaEntities 70
4.5.2 MetaActions 71
4.6 SUMMARY 73

5.0 INTERPRETATION OF GEOSPATIAL QUERIES 74
5.1 Q-GET: FORMULATING GEOSPATIAL QUERIES 74
5.2 Q-ANALYZE: MAPPING QUERIES TO OPERATIONS 78
5.3 W-COMPOSE: COMPOSING GEOPROCESSING WORKFLOWS 82
5.4 SUMMARY 86

6.0 PROOF OF CONCEPT 87
6.1 INTRODUCTION 87
6.2 PROOF OF CONCEPT 87

6.2.1 Software Tools ... 87
6.2.2 Using Proof of Concept ... 89
6.3 CASE STUDY I: GENERIC GEOSPATIAL QUERIES ... 91
6.3.1 Queries ... 91
6.3.2 Representation of Queries ... 92
6.3.3 Results ... 96
6.4 CASE STUDY II: APPLICATION-SPECIFIC GEOSPATIAL QUERIES ... 99
6.4.1 Background ... 99
6.4.2 Application Domain Specialization ... 100
6.4.3 Query and Result ... 102
6.5 SUMMARY ... 106

7.0 CONCLUSIONS AND FUTURE RESEARCH ... 107
7.1 SUMMARY OF THE RESEARCH ... 107
7.2 CONCLUSIONS ... 108
7.3 FUTURE RESEARCH ... 108

APPENDIX: SAMPLE GEOSPATIAL QUERIES ... 110

BIBLIOGRAPHY ... 115

LIST OF TABLES

Table 2.1 Evolution of GIS (after [74]) 13
Table 2.2 Major application areas for GIS 15
Table 2.3 Geoprocessing operations for spatial analysis 27
Table 2.4 Results when the "touch" operator is applied to two polygons 29
Table 3.1 Phases in formuating geospatial queries 45
Table 3.2 Characteristics of geospatial queries in different application domains 49
Table 3.3 Samples of transformation subqueries in environmental health (after [9]) 52
Table 3.4 Types of subqueries organized based on their input/output objects 53
Table 4.1 Actions in geospatial queries 64
Table 4.2 MetaActions mappings 71
Table 4.3 Role conversion mappings in MetaActions 72
Table 6.1 Comparisons between CAD and GIS 100
Table 6.2 Roles and entities in AEC/FM 103
Table 6.3 Entity-role-data model mappings 103
Table 6.4 Action and roles mapped to operators 103

LIST OF FIGURES

Figure 1.1 Geospatial problem solving using GIS 2
Figure 1.2 Current approach to geospatial problem solving using GIS 4
Figure 1.3 GeoInterpret's approach to geospatial problem solving 6
Figure 1.4 Research objectives and components 7
Figure 2.1 A Triangulated Irregular Network 18
Figure 2.2 A layered map containing both vector objects and raster backdrop 19
Figure 2.3 Graticule of Earth 21
Figure 2.4 Map Projections 22
Figure 2.5 Geoprocessing data flow (after [44]) 23
Figure 2.6 Point, line, and polygon buffers 25
Figure 2.7 Classification ranking (after [74]) 26
Figure 2.8 Heterogeneity in information systems (after [104]) 28
Figure 2.9 The Meaning Triangle (after [78]) 34
Figure 2.10 The Five Universe Paradigm (after [32]) 40
Figure 3.1 Query solving through GIS: past, present, and future 46
Figure 3.2 Common components and types of a geospatial query 50
Figure 4.1 GeoInterpret approach to the Five-Universe Paradigm 57
Figure 4.2 Components of GeoInterpret 60
Figure 4.3 Knowledge in GDO 61
Figure 4.4 Specialization of common geospatial ontology 62
Figure 4.5 Entity and action 63
Figure 4.6 Entity, role, and action 63
Figure 4.7 Roles and entities 64
Figure 4.8 Entities, roles, and action in a modified CG representation 65
Figure 4.9 Selection subquery as modified CG 66
Figure 4.10 Chained subqueries as modified CG 67
Figure 4.11 MetaEntities mappings 70
Figure 5.1 Flowchart of the Q-GET algorithm 75
Figure 5.2 Modified CG representation for the FindShortestRoute action 76

Figure 5.3 A query to find the shortest route from building to park 77
Figure 5.4 A query to select only land parcels that cost less than $100,000........ 77
Figure 5.5 Flowchart of the Q-ANALYZE algorithm 78
Figure 5.6 A scenario in modified CG form 80
Figure 5.7 Flowchart of the W-COMPOSE algorithm 83
Figure 5.8 Geoprocessing workflow for the scenario 85
Figure 6.1 Knowledge base in Protege 88
Figure 6.2 MetaEntities slots in Protege 88
Figure 6.3 List of actions in the proof of concept 89
Figure 6.4 Binding entities to roles in the proof of concept........ 89
Figure 6.5 List of subqueries in the proof of concept 90
Figure 6.6 Resulting workflow generated by the proof of concept........ 91
Figure 6.7 Modified CG for Query #1 93
Figure 6.8 Modified CG for Query #2 94
Figure 6.9 Modified CG for Query #3 95
Figure 6.10 Geoprocessing workflow for Query #1 96
Figure 6.11 Geoprocessing workflow for Query #2 97
Figure 6.12 Geoprocessing workflow for Query #3 98
Figure 6.13 IFC architecture (after [54]) 101
Figure 6.14 IFC knowledge integration to GeoInterpret........ 102
Figure 6.15 Modified CG for the AEC/FM query 104
Figure 6.16 Workflow for the AEC/FM query 105

PREFACE

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Hassan Karimi, for all the invaluable advices and insights that made this dissertation possible. I also would like to thank the dissertation committee members, as well as the Department of Information Science and Telecommunications, for their supports. Lastly, I would like to thank the government of Thailand and Thammasat University for providing the financial support for my graduate studies in the United States.

1.0 INTRODUCTION

This dissertation research contributes a methodology that provides a set of techniques and algorithms utilizing *ontologies* as knowledge base to perform automatic interpretation of *user-level* geospatial queries and translate them to *geoprocessing* workflows for solving queries. The result of interpreting a query is a sequence of operations that when implemented by a *Geospatial Information System* (GIS) will result in the solution to the query. An example of a user-level geospatial query is "*Locate all hospitals that are within 5 miles from 135 N. Bellefield Ave.*" Interpreting this query would result in such geoprocessing operations as: geocoding, database query, distance computation, and selection. These operations, when implemented, will result in a solution, presented as a map or otherwise, that answers the query.

1.1 PRELIMINARIES

As most human activities are confined to the surface of the Earth, there often is a need to know not only the nature of the activities, but also the location associated with them as well. Problems that involve an aspect of location, either in the information used to solve them or in the solutions themselves, are considered to be geographic or geospatial problems [75]. Geospatial problem solving is part of a decision-making process involving objects and phenomena near or on the Earth's surface where information about them, including their Earth-referenced locations, are collected and stored as *geospatial data* (Figure 1.1). GIS are computer-based tools that allow for the transformation of geospatial data into information that can be used to make decisions [22]. They are also traditionally known as *Geographic* or *Geographical Information Systems* in the literature. Computations performed on geospatial data are generally referred to as g*eoprocessing* operations. In this research, we use the term *geospatial queries* to mean geospatial problems that are solved through GIS.

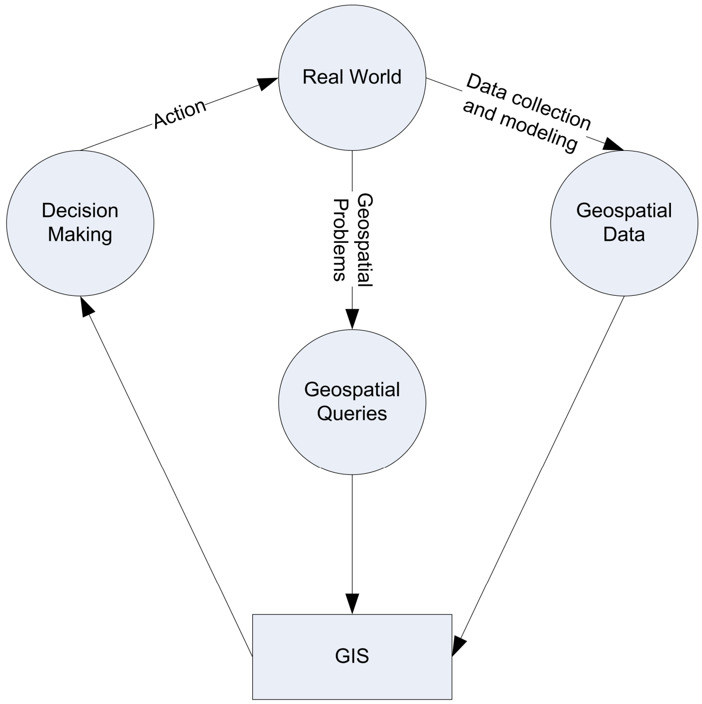


Figure 1.1 Geospatial problem solving using GIS

The extent of usefulness of GIS has been demonstrated across many diverse application domains over the period of several decades. They have long been used traditionally by large governmental organizations for land management and military applications, among others. However, recently GIS have become enormously popular and widely used in diverse applications by many different user communities. Applications in public health, transportation, business, environmental and others have benefited greatly from GIS. Their proliferation was due to the improvement in computing technologies and their lower cost, which also resulted in migration of GIS from mainframe computing environments to personal computers (PC), the World Wide Web, and mobile devices. GIS technology and its history are discussed in more details in Chapter 2.

1.2 RESEARCH PROBLEM, CHALLENGES, AND SIGNIFICANCE

Although many mainstream software packages today integrate some features of GIS as one of their core components, they are designed and targeted to solve specific kinds of geospatial queries and have a limited scope and functionalities. For example, a software package within an embedded in-car navigation system is designed and marketed for one specific purpose: real-time street navigation in a vehicle. Similarly, a PC-based mapping software package that is only capable of displaying a location of a place, given its address, on a map is generally not considered a full-featured GIS. On the other hand, software packages generally known as full-featured, general-purpose GIS have the characteristic of providing "toolbox" geoprocessing tools which can be applied to solve a wide variety of geospatial queries in many application domains; an example of this type of GIS is *ArcGIS* from *ESRI* Inc. [27]. These geoprocessing tools are generic in nature and provide users with the flexibility to customize and combine them to solve specific and specialized problems. However, despite such flexibility and power, a major drawback of current generic GIS is that effective and efficient utilization of the software, even for simple geospatial queries, requires that the users must become "GIS experts" by learning about many facets of geospatial query solving. This requisite knowledge is in addition to the knowledge of application domains and includes:

- Knowledge on how concepts, objects, and activities in real-world application domains are modeled as GIS objects (e.g., points, lines, polygons).
- Knowledge on the types of geospatial data sets needed to solve the queries and how they are obtained and integrated into GIS. This includes knowledge of sources and locations of data sets, method for correctly identifying data sets (e.g., searching and understanding metadata), and method for retrieving and integrating data sets into GIS, including data schema and format conversions.
- Knowledge on how to use specific GIS software packages, such as understanding command sets, available geoprocessing operations and their behaviors.
- Knowledge on the types of geoprocessing operations needed to solve problems and how to apply them with appropriate data sets.

As an example, to solve a simple geospatial query "*identify the location where a railroad crosses a county boundary*", the user would first need to know that, in the context of this particular query, railroads are represented as lines and county boundaries are represented as polygons in GIS. The user would then need to locate, identify, and retrieve the data sets representing railroads and counties in the region of interest. Once the data sets are retrieved, they often need to be converted, schematically and syntactically, such that they can be integrated in a single GIS database. Next, the user would also need to know that in order to solve the query (i.e., identify the location of crossing), a geoprocessing operation that tests for geometric intersection must be performed on the data sets, and that the desired railroad and county must be identified through some kind of selection operation. Furthermore, all these tasks must be performed using a chosen GIS software package which may have its own idiosyncrasies that the user must deal with.

These requisite skills and knowledge needed to utilize GIS are typically gained through extensive GIS training and experience in implementing GIS projects. However, since usually many users of GIS come from application domains (e.g., environmental, ecology, transportation, business, public health, etc.)

and do not have such skills and knowledge, the widespread use of GIS is hindered. The most important phase in using these skills and knowledge is in interpreting application domain concepts presented in queries and translate them to concepts in the geoprocessing domain. Currently, query interpretation is done in an *ad hoc* fashion; namely, if the query originator is not a GIS expert, then he or she must communicate the queries with a GIS expert who will utilize a GIS to solve them (Figure 1.2). This is the main reason why many application domain experts have to become GIS experts themselves, even though the time and cost of becoming a GIS expert is high.

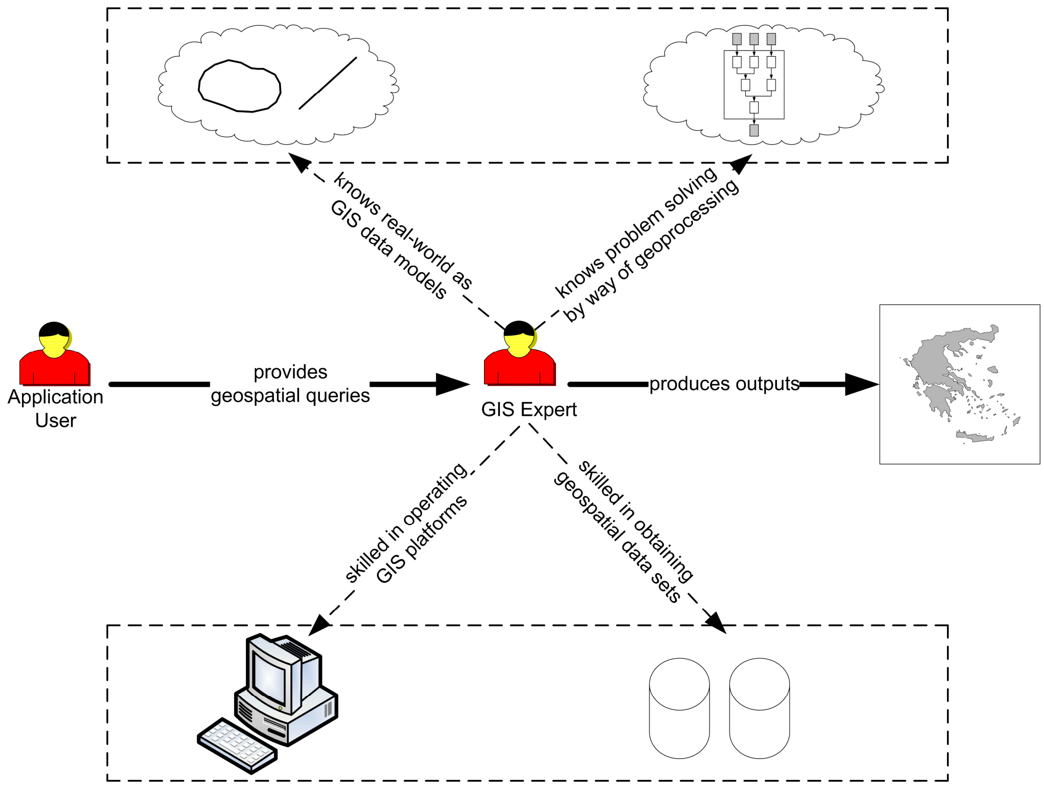


Figure 1.2 Current approach to geospatial problem solving using GIS

In summary, geospatial query solving using current, generic GIS is currently an activity that belongs to GIS experts who possess a set of specialized knowledge. Those wishing to use GIS to solve problems in their respective application domains need to obtain these knowledge and skills through training and/or practice. Acquisition of such knowledge and skills imposes a heavy burden, both cost and time, on users due to the complexity and the extent of the knowledge required. In addition, because GIS software packages often do not interoperate well among each other [64], perceived as difficult to learn and use, and that GIS technology changes rapidly, the burden on users increase if they wish to keep themselves knowledgeable on various GIS platforms and up-to-date on technology.

Due to the reasons described above, we argue that the difficulties in using current GIS are the main obstacle that prevents its widespread use by non-experts and limits its potential benefits for many application domains and user communities. Although advances in computing and information technologies have enhanced the capabilities of GIS, the usability of GIS by non-experts is currently accomplished by embedding GIS features underneath non-GIS software packages specialized for specific applications. In order for GIS to be generally usable by all users (experts and non-experts) to solve geospatial problems in various application domains, a new approach that bridges the gap between users' knowledge of geoprocessing and GIS tools and the GIS knowledge and skills required to solve them is needed.

To fill this gap and alleviate the difficulties in using GIS, a methodology that automates the process of interpreting geospatial queries is proposed. The development of such a mechanism poses several challenges. First, as previously mentioned, GIS are generic tools that are used widely by different communities to solve diverse types of geospatial queries. This indicates that the conceptualization of queries and, as a consequence, the GIS representation of them vary widely from one application domain to another. For example, objects in transportation-related geospatial queries are generally conceptualized as discrete objects with topological information explicitly represented among them. In addition, geospatial data sets for transportation analysis are typically gathered in vector form by means of such advanced technologies as *Global Positioning System* (GPS) and have the geographic scale spanning an area of a city. On the other hand, geospatial objects in ecology-related geospatial queries are typically conceptualized as activities contained inside a continuous field over a much wider region of interest (e.g., county- or state-wide). Many geospatial data sets used in ecological analyses are often obtained through remote sensing (e.g., satellite imagery) and the resulting data are simply images (i.e., in raster form). Such differences in conceptualization, scale, data, and representation among different application domains imply that any overarching methodology must take into account a wide variety of views that exist in geospatial query solving.

In addition, due to historical and practical reasons, different GIS were developed independently with little regard to interoperability [46]. As GIS move from stand-alone systems to distributed heterogeneous networked environments, problems related to interoperability will only worsen. In terms of automation, interoperability is crucial in that it allows different GIS to work together in concert to solve geospatial queries. In recent years, the *Open Geospatial Consortium* (OGC) [92] has begun to address this issue by publishing interoperability standards specifically for GIS in the Internet environment. However, the use of these standards is not yet widespread at the time of this writing.

1.3 RESEARCH OBJECTIVES AND METHODOLOGY

In this research, we focus on GIS automation where users' geospatial queries, which are expressed in application-domain contexts, are interpreted and mapped to geoprocessing workflows. Toward this goal, we propose a methodology, called *GeoInterpret*, that uses ontologies to store GIS expertise and enable automated interpretation of user-level geospatial queries [99]. Query interpretation is the most burdensome activity in geospatial query solving in terms of the effort users need to invest in order to effectively and efficiently utilize GIS. The benefit of GeoInterpret is to alleviate users' burden in gaining knowledge and skills related to geoprocessing and GIS for solving geospatial queries (Figure 1.3).

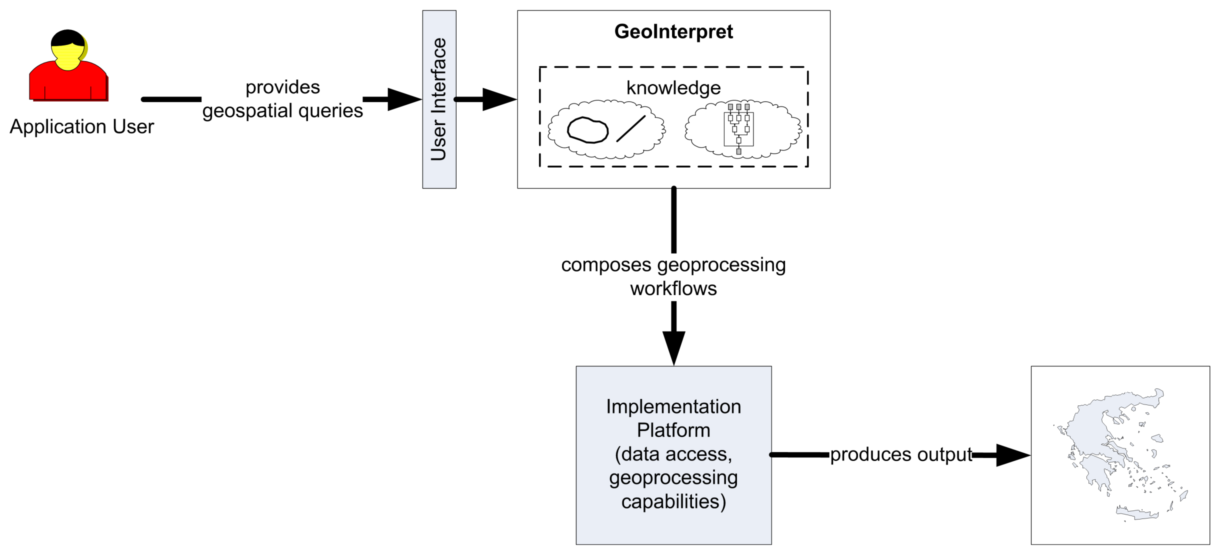


Figure 1.3 GeoInterpret's approach to geospatial problem solving

In this research, we take an *ontological engineering* approach in the development of the proposed methodology. Ontology in the context of this research is an explicit formalization that describes and captures real-world objects, concepts, and activities. The crucial difference between an ontology and a database schema is that the ontology is closer to a human cognitive model of the domain [33]. As a consequence, using ontology to bridge the gap between users and computerized information systems is a promising approach and is currently an active research area not only in the geospatial community but also in the larger computer science research community (e.g., the *Semantic Web* [129] community).

Ontological engineering refers to the set of activities that concern the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suites and language that support them [41]. However, this research does not aim to *build* geospatial ontologies, rather it aims to provide a set of methods for *utilizing* ontologies for the purpose of interpreting geospatial queries and translate them to geoprocessing operations. Building an ontology for a domain is a long-term and ongoing task which requires continuous involvement of the community at-large [81]. Once an ontology is realized, utilizing it in a useful manner would require a novel methodology. This research is meant to provide a methodology for using ontology for the purpose of bridging the gap between users and GIS, making GIS simpler for solving geospatial queries.

GeoInterpret approaches the problem of geospatial query interpretation by using the characteristics of real-world geospatial queries as its basis. While other research efforts focus on implementation aspects of geospatial query solving [7, 59, 66], this research is focused on a general methodology that is independent from any implementations or application domains. The proposed research objectives are as follows:

- To develop a methodology that uses multiple ontologies and techniques for mapping elements between ontologies.
- To determine common structure, conceptualization, and to develop a representation of geospatial queries.
- To develop algorithms for query interpretation, mapping, and composition of geoprocessing workflows.
- To develop a proof of concept and use it in case studies involving generic geospatial queries and application-specific geospatial queries.

Figure 1.4Figure 1.4 summarizes the research objectives and major components.

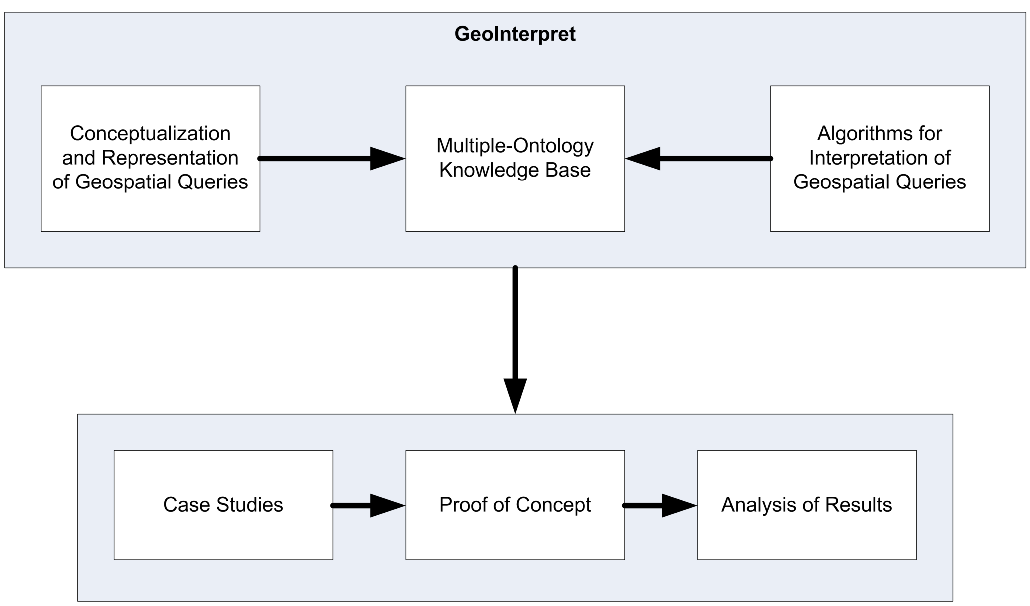


Figure 1.4 Research objectives and components

1.4 CONTRIBUTIONS

The main contribution of this research is the GeoInterpret methodology, which consists of:

- General conceptualization and representation of geospatial queries, including graphical-based representation of queries based on Conceptual Graph (CG) [114, 115].
- A multiple-ontology architecture for interpretation of geospatial queries. The architecture defines the various components and how they interact among each other.
- Techniques for mapping elements in different ontologies representing different levels of geospatial concepts.
- A set of algorithms that performs geospatial query interpretation, mapping, and geoprocessing workflow composition.

These contributions provide the groundwork for the automation of GIS by alleviating one of the users' burdens in using GIS, namely, the interpretation of geospatial queries and the mapping of queries to geoprocessing operations. Due to the diversity of application domains that use GIS, the characteristics of each domain, and the fact that different domains require domain-specific expertise, we do not directly address issues specific to a given application domain in this research. In addition, because GeoInterpret is an implementation-independent methodology, we do not address specific implementation strategies, such as user interface, interoperability, and the discovery and invokation of geoprocessing services. Interested readers may refer to [7, 12, 59, 66, 100] for those topics.

1.5 OUTLINE OF THE DISSERTATION

The remainder of the dissertation is organized according to the research objectives outlined in the previous section. Chapter 2 provides an additional background on GIS and related work on ontologies in GIS. Chapter 3 provides a discussion and analysis of geospatial queries and proposes a conceptualization for them. Chapter 4 discusses GeoInterpret, knowledge representation for geospatial queries, and techniques for inter-ontology concept mapping. Chapter 5 discusses the set of algorithms that uses GeoInterpret for query interpretation and mapping to geoprocessing operations. Chapter 6 discusses the proof of concept and the case studies for demonstrating the working of GeoInterpret. Chapter 7 concludes the dissertation and provides future research suggestions.

2.0 RELATED LITERATURE

This chapter provides a general background on GIS and related concepts and techniques relevant to this research. In addition, issues related to interoperability are discussed including relevant standards. Lastly, ontologies and research efforts related to ontologies in the context of GIS are discussed.

2.1 GEOSPATIAL INFORMATION SYSTEMS

GIS are a class of information systems that use computers to deal specifically with geospatial data to solve geospatial problems [74]. An information system is a set of interrelated components that retrieve, process, store, and distributed information to support decision making [73]. The word "geospatial" or "geographic" in GIS implies that data in the system are pertinent to features and resources at or near the Earth's surface, including human activities associated with them [75]. It also implies that problem solving in GIS deals with geography, which includes location, distribution, pattern, and relationship within a specific geospatial reference framework. Computation on geospatial data to solve geospatial problems makes GIS unique among information systems.

GIS can be applied to solve many problems in many disciplines. For example, in business applications, GIS can be used to perform demographic analysis based on location of key places to improve customer service and marketing effort. In national defense, GIS have always been an essential and integral part of military and intelligence where geography plays a crucial role. In government, GIS are extensively used in urban planning and management of land resources, including the planning and management of transportation networks, zoning, addresses, and housing. GIS also play an important role in agriculture, environmental management, forestry, mining, and other areas related to natural resources that use location information. In addition to these traditional GIS applications, new applications and services are emerging as a result of advancements in various technologies. The emergence of world-wide information and communication infrastructures, such as the Internet, has opened up new opportunities, created new problems, and challenged for new applications of GIS to be conceived.

The emergence of wireless telecommunication infrastructures has created a new environment where users have become geographically mobile. This mobility created new requirements, problems, possibilities, and challenges to the use of the infrastructures. As users become mobile, their locations can now be used to improve existing services, or to deliver new services. This *Location-Based Service* (LBS) paradigm has become an important research area encompassing the fields of computer science, engineering, telecommunications, and geospatial information science. Essentially, LBS use location of users to determine *what* information to deliver and *when* to deliver them. An in-car navigation system that tells the driver of the location of the nearest gas station is an LBS application because it uses the location of the car to determine the location of the gas station, and it delivers the information to the driver in a timely manner. GIS technology plays a significant role in LBS because of its location information processing capability.

The World Wide Web has become the world-wide communication medium and is the foundation for future information infrastructures. In addition, the development of the next generation web, also known as the *Semantic Web*, has the potential to improve the capability of the existing information infrastructures to deal with a very large amount of information in distributed, heterogeneous networked environments. The Web has expanded the use of GIS from the traditional setting of stand-alone applications on a single computer, to the large-scale setting of networked environments consisting of diverse types of hardware and software. This change has let to the creation of new types of applications, such as web-based *ad hoc* mapping and routing services (e.g., http://www.mapquest.com). In addition, the networked nature of the Internet and the Web has enabled distribution of data and processing, allowing more data to be accessed and more processing to be performed across multiple computers [1]. Grid Computing (GC) [14, 36, 37] is an emerging environment where such distribution occurs. GC has originally been defined to refer to a distributed, scientific and engineering computing environment. However, its context and scope have been expanded to include web services and peer-to-peer computing. As certain GIS applications are both computation- and data-intensive, GIS can significantly benefit from the GC paradigm.

2.1.1 Definitions

GIS can be looked at from different perspectives. From a database point of view, a GIS can be defined as *"a database system in which most of the data are spatially indexed, and upon which a set of procedures operated in order to answer queries about spatial entities in the database"*[111]. Others have defined GIS from an organizational perspective that emphasizes the role of institutions and people using the systems. One such definition states that a GIS is *"an automated set of functions that provides professionals with advanced capabilities for the storage, retrieval, manipulation and display of geographically located data"* [98]. From a computer system perspective, the *United States Geological Survey* (USGS) defines a GIS as *"a computer system capable of assembling, storing, manipulating, and* *displaying geographically referenced information, i.e., data identified according to their locations"* [124]. A simple working definition of GIS given by [74] states that a GIS is "*a set of computer-based systems for managing geographic data and using these data to solve spatial problems.*"

2.1.2 Historical Background

The origin of GIS can be traced back a few decades to the research and development efforts of early computer systems for handling graphical data and *Database Management Systems* (DBMS). The system generally recognized as the first GIS was developed in Canada, called the Canada GIS (CGIS), which became operational in 1971 [121]. The system was designed to address the needs of land and resource information management of the federal government of Canada. In 1973, the USGS started the development of the *Geographical Information Retrieval and Analysis System* (GIRAS) to handle and analyze land-use and land-cover data [85]. Other systems were also developed in Europe mostly for land-use applications by government agencies. In addition, there was a growing interest in computer-based mapping and a number of systems in universities were developed to provide map data processing capabilities. The early years of GIS, from 1960s to 1980s, can be characterized by the development of many GIS software packages to handle and analyze application-specific geographic data. These packages were run on mainframe computers in the batch mode as stand-alone software and used proprietary programs and data structures. Due to the high cost and limited computing power available at the time, only large governmental organizations and universities utilized such GIS.

Between the early 1980s and mid 1990s, the lower cost of computing and the increased computing power allowed GIS to move from large governmental organizations and universities to commercial sectors. In 1982, ESRI Inc. released *ArcInfo*, a software package based on minicomputers that was one of the first vector-based GIS to combine topological and relational data structures [86]. The development of GIS was greatly accelerated by further advances in computing technology and lower cost of computing power. Advances in operating systems, graphical user interfaces (GUIs), and database systems allowed GIS to become multi-platform applications running on different types of computer systems. In addition, GIS applications became increasingly diversified and more sophisticated. No longer limited to land-use type of applications, GIS became an important tool in engineering, social sciences, physical sciences, business, facility management, and other fields which needed to deal with information related to location.

Since the mid 1990s, the focus of computing has shifted from stand-alone and locally networked environments to wide-scale, distributed, heterogeneous computing infrastructures. The exponential growth in the use of the Internet has enabled and compelled new ways of using GIS. A wide range of GIS applications on the Internet are now widely available for users from anywhere in the world to access them. In addition, the proliferation of wireless and mobile computing technologies, such as cellular phones and Personal Digital Assistants (PDAs), has provided new platforms and paved the way for the emergence of new GIS applications. Because of the advances in computing, GIS applications are now designed, implemented, and applied very differently from their predecessors. GIS software of the future are anticipated to be used in multi-tier, heterogeneous network environments, where computers of different platforms co-exist and geoprocessing tasks are performed in a distributed manner.

Furthermore, GIS are becoming an integral component of other types of information systems by being tightly coupled with application software in other application domains, such as those available in the business and engineering sectors. Examples of GIS software integration include integrated mapping tools for displaying and analyzing demographic information for marketing and large-scale telecommunication infrastructure planning. Unlike generic GIS such as ArcGIS or ArcInfo, these GIS software packages are designed and implemented for a specific purpose and provide little or no flexibility in applying to other applications. Table 2.1 summarizes the history of GIS.

Table 2.1 Evolution of GIS (after [74])

	1960 - 1980	1980 – mid 1990s	Mid 1990s – present	Future
Technology	Mainframe computers Application-specific proprietary software and data formats Mainly raster-based	Mainframe and minicomputers Spatial and relational data structures Vector data Generalized standard software packages	Workstations and PCs Networked environments Web-based applications Open systems design Data and application integration, multimedia Object-oriented data model	Ubiquitous mobile devices Wireless network environment Interoperable Transparent ad hoc applications Semantically-aware
Users	Government Universities Military	Government Universities Military Utilities Business	Government Universities and Schools Military Utilities Business General public	Government Universities and Schools Military Utilities Business General public
Applications	Land and resource management Census Surveying and mapping	Land and resource management Census Surveying and mapping Facilities management Market analysis	Land and resource management Census Surveying and mapping Facilities management Market analysis Geographic data browsing	Land and resource management Census Surveying and mapping Facilities management Market analysis Geographic data browsing Mobile ad hoc applications

2.1.3 Components of GIS

One way to look at a GIS is to divide it into four main components: *data*, *technology*, *application*, and *people* [74].

The data component of GIS is generally referred to as *geospatial* data. The term *spatial* refers to any space whereas the term *geo* refers to the Earth's surface. Since many concepts and techniques in GIS can be applied to non-geographic domain (e.g., spatial analysis on genome data), the term geospatial implies a subset of spatial applied specifically to the Earth's surface [75]. Geospatial data consist of geometric and attribute components. Geometric component contains coordinates of entities within a frame of reference. For example, a particular location is identified along with its latitude/longitude coordinates, which are grounded to a spherical reference system of the Earth. The attribute component contains characteristic information of a geographic feature that can be linked to the geometric data. This characteristic information are typically described in text and numbers, but images, sounds, and other types of data can be used as attributes as well. Attributes are typically stored in tabular format and are linked back to geometric data through unique identifiers.

The technology component of GIS simply consists of computing hardware and software. The hardware of GIS is made up of components that are used for acquisition, storage, analysis, and display of geospatial information. Today, network infrastructure is also considered a hardware component as many GIS are now developed and applied in networked environments, typically through the client/server architecture. For software, GIS are conventionally developed using a hybrid approach of utilizing a separate module for handling geometric data, and a separate DBMS for handling attributes and descriptive data. Typically, the module for handling geometric data is proprietary to a GIS software vendor, while the DBMS is usually taken from an outside commercial vendor. Recently, however, object-oriented techniques have been used in the development of GIS that allow both geometric and descriptive data to be stored in a single database [74].

The application component is the context in which GIS are applied to solve problems. Application areas in GIS have quickly grown in a relatively short amount of time. When GIS was first developed, it had a narrow focus typically in the area of land-use and natural resource management. GIS today are used in nearly all sectors of the economy and their applications have become more diversified. Table 2.2 shows some major application areas of GIS.

Table 2.2 Major application areas for GIS

Sector	Application Areas
Academic	Research in social science, engineering, health, etc. Teaching tools
Business	Real estate planning, management, sales Retail and market analysis Goods distribution and delivery Facilities management
Government	Environmental Land management Census Weather services Public safety
Industry	Transportation Utilities Telecommunications Mining
Military	Training Command and control Intelligence

The people component includes users and developers of GIS. In the past, users were not normally recognized as a component of GIS due to the fact that the users of GIS in the early days were specialist computer programmers who were using GIS in the batch mode with little interaction with the systems. The growth of GIS in the past decade has resulted in a dramatic increase in the number of users along with more diverse applications and their sophisticated requirements [38]. This led to a growing interest in human factor in GIS applications. As GIS applications are typically visual in nature, some studies and research efforts have been concerned with the cognitive characteristics of GIS, as well as issues relating to human-computer interaction [12, 79, 122].

Lo and Yeung [74] classify GIS users into three categories: *viewers*, *general users*, and *GIS specialists*. Viewers are the public at large whose only need is to occasionally browse a geographic database for referential information. The primary requirements for the viewer are the ease of use and accessibility of the system. General users are those who use GIS for conducting business and they include managers, engineers, scientists, and lawyers. These users are considered more active as they use GIS regularly to satisfy their specific information needs. Due to the diversity of the general users, their requirements of GIS may vary considerably in terms of human-computer interactions and types of analysis performed on geospatial data. GIS specialists are those who implement GIS and they include GIS managers, database administrators, application specialists, systems analysts, and programmers. They are responsible for GIS design, development, implementation, and maintenance.

In this research, we distinguish GIS users into two types: experts and non-experts. We refer to GIS expert as the one who knows the techniques and procedures in using general-purpose GIS software packages to solve geospatial queries. We refer to non-expert as the one who needs to solve geospatial queries but does not have the requisite knowledge in GIS and must rely on a GIS expert to solve geospatial queries.

2.1.4 Conceptual Models for Geospatial Objects and Spaces

Geography deals with the physical features and phenomena of the Earth's surface or near surface, which are highly complex and rich in variety. GIS are a class of information systems that primarily use computers to perform tasks for solving problems related to geography. Therefore, real-world geographic features and phenomena must be realized and represented for use in computer systems. This section discusses the two main conceptual models for representing geographic objects: they are the *object* model and the *field* model [16].

The object model represents the world as an empty space occupied by discrete, geometrically-defined entities or objects that are both identifiable and describable by attributes. An entity may be a natural geographic phenomenon (e.g., a river, a mountain), or constructed by humans (e.g., a building, a road). Objects in geographic conceptual models are generally of zero- to two-dimensional in nature. An object occupying a region, such as a lake or a land parcel, is two-dimensional and conceptualized as an area or a polygon. A one-dimensional object, such as a road or a river, is conceptualized as a line. An individual location, such as a street intersection, is zero-dimensional and is conceptualized as a point. Though humans perceive the world in three dimensions, the ability for GIS to truly handle three-dimensional objects is very limited and fewer dimensions are used for approximation [75]. The interpretation of a geographic space containing entities also depends on the semantics associated with it [103]. For example, if we consider the geographic area of Pennsylvania, an administrative point of view would partition the area into several, smaller areas called *counties*. On the other hand, a geological point of view would organize and partition the same area very differently, depending on the needs of particular geological applications.

The other conceptual model of the geographic space is the field model, which represents geographical space in terms of continuous Cartesian coordinates in two, three, or four dimensions if time is included [11]. An attribute, such as temperature, air pressure, or elevation, by its nature usually varies smoothly and continuously over the space. In this model, each point in space is associated with one or more attribute values, defined as continuous functions. For example, the altitude above the sea level is a function defined over x and y coordinates, whose result is the value of a variable h, indicating the altitude above the sea level. The field model is useful when geographic features do not naturally fall into categories of points, lines, or areas. Fields can be distinguished by the varying features and how they vary. Fields can also be defined by continuous variation along lines, rather than across space [75]. Examples are traffic density, which can be defined on a road network, and water flow volume, which can be defined along a river path.

2.1.5 Computer Representation Models

The conceptual models of geospatial objects and spaces must be represented in computer systems in order to allow for geoprocessing. This section discusses the two generally-accepted computer representation models in GIS: *vector* and *raster*.

The object conceptual model for representing geospatial objects is typically represented as *vector* data in GIS. The vector data model consists of discrete objects defined using fundamental geometrical units of *points*, *lines*, and *polygons*. These objects are usually geospatially referenced by Cartesian coordinates in GIS. The coordinates that define the geometry of each object may have two, three, or four dimensions, where the fourth dimension can be time or another property. In some data models, geospatial objects can be defined by a mathematical function, such as *spline* and *Bezier* curves [75].

The vector data model used in GIS consists of two main components: the geometric component and the attribute component. The geometric component is made up of points, lines, and polygons representing geospatial objects. The attribute component is the information describing those objects. For example, a line representing a street has a name associate with it; the line object is the geometric component and the street name is an attribute component of that line object. The attribute component is typically stored in a DBMS and the geometric component is typically stored using proprietary schemes.

Entities encoded using the vector model are often called *features* and GIS commonly deals with two types of them: *simple* and *topologic*. Simple features are represented solely by their locations without explicit representation of the spatial relationships among simple features. This representation is sometimes called spaghetti as lines can overlap (like strands of cooked spaghetti on a plate) without any topological relationship. On the other hand, topologic features are simple features structured using topology. Topology in GIS is generally defined as the spatial relationships between features [119]. For example, in a topologically structured polygon, each polygon is defined as a collection of lines, where each line is in turn made up of an ordered list of points. When two adjacent polygons share a common line, the same ordered list of points representing that line is used for both polygons. This way, the potential problems of gaps or overlaps between adjacent polygons are avoided. Other models for describing topological relations between objects were proposed by [24, 25]. In addition, topological information can be used to test or validate the geometry of vector entities (e.g., polygons can be tested to see if they are closed; a network of lines can be checked to see if all the lines are connected).

The network data model is a topological model containing information about the connectivity of points and lines within a network. For example, points are used to represent street intersections, and lines are used to represent streets. There are many GIS applications that use the network data model, such as routing for car navigation [61], finding optimal transit bus routes [65], and managing urban resources. For example, to perform analysis on a network of city sewer pipes, rules may be defined on how water can flow through the network. In this case, the pipes are represented as lines in the network, and water valves and pipe intersections are defined as points. The water flow from residential buildings to a water treatment plant is directional and the rate of flow is modeled as impedances on the points and lines [75].

In GIS, there are often needs to represent geographic terrains as three-dimensional surfaces. For this purpose, *Triangulated Irregular Networks* (TINs) are often used. A TIN is a vector computer representation for three-dimensional data and can be generated, for example, from a *Digital Elevation Model* (DEM), which is a digital raster file consisting of a sampled array of elevations for a number of ground positions at regularly spaced intervals [125]. A TIN is constructed from adjacent non-overlapping triangles with each individual sample value corresponds to a vertex of a triangle. The value of any point not on the vertex is computed by linear interpolation of the three vertices of the triangle that contains that point (Figure 2.1). The size of triangles in a TIN varies according to the density of the sample points and can be adjusted as needed. The variable density of triangles means that a TIN is an efficient way of storing surface representation.

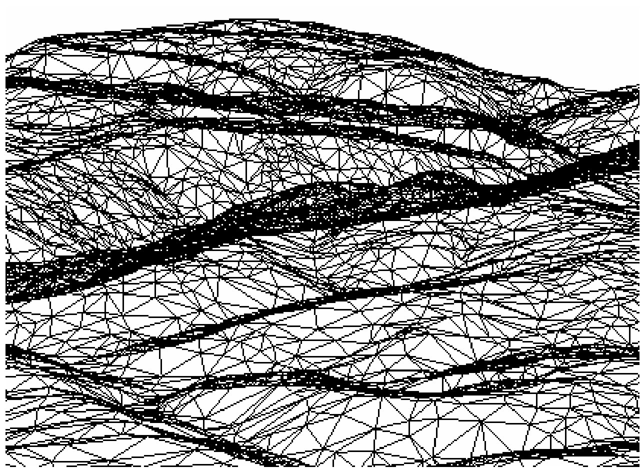


Figure 2.1 A Triangulated Irregular Network

The other type of computer representation of geospatial objects and spaces is the *raster* data model. A raster is an array of cells of equal shape and size; in most cases the cells have square shape and are called *pixels* and represented in a regular two-dimensional grid. The cells can hold attribute values typically based on categories, integers, and floating point numbers. In the simplest case, a binary value is used to indicate presence or absence of a certain property associated with the space represented by the cells (e.g., presence of vegetation). In a more complex representation, floating point values are used to represent continuous variation of an attribute over a space. An example is the continuous representation of altitude above the sea level for a particular terrain. Because of its features, raster data model is often used in GIS to represent spaces conceptualized as continuous fields. Raster data are often obtained through remote sensing (i.e., satellite imagery and aerial photography) and is especially useful as backdrops for map display. Often in geoprocessing, raster data are converted to vector in order to discretely identify objects.

When representing geographic features, it is convenient to group entities of the same geometric type together. For example, all point entities, such as buildings and landmarks, are grouped and stored together; and all line entities, such as roads and highways, are also grouped and stored collectively. A collection of entities of the same geometric type is referred to as a *layer*, which can be visualized through mapping. Figure 2.2 is an example of a map showing two layers of data. The first layer contains vector line objects representing street network. The second layer contains a raster backdrop taken from a satellite.

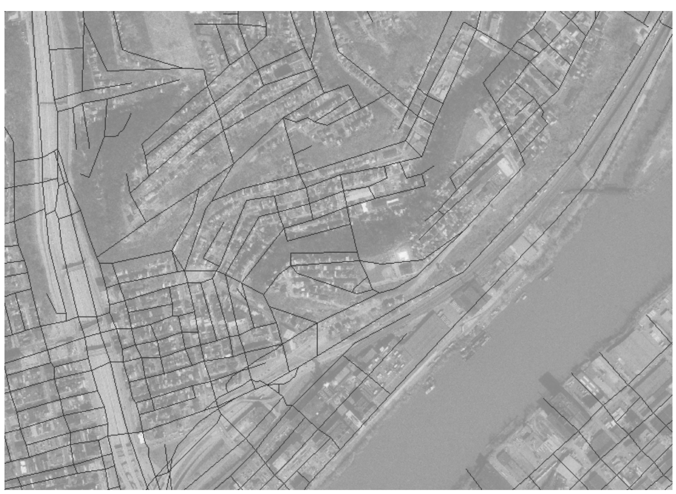


Figure 2.2 A layered map containing both vector objects and raster backdrop[1]

[1] Sources: The raster backdrop image was obtained from Pennsylvania Spatial Data Access (http://www.pasda.psu.edu) and the street vector data was obtained from the U.S. Census Bureau (http://www.census.gov/geo/www/maps). Both data sets are available to the public.

2.1.6 Coordinate Reference Systems

Geospatial data are grounded to the Earth's surface by a *Coordinate Reference System* (CRS). A coordinate is a set of numbers that determines the location of a point within a frame of reference. By using CRS, computing and data transformation methods can be consistently performed.

The spherical coordinate system, called the *Geographic Coordinate System* (GCS), is the most well known Earth-grounded CRS. It uses a network of *latitude* and *longitude*, also known as *graticule* (Figure 2.3), to fix positions on the Earth's surface. The two primary reference points in this system are the geographic North and South poles, which are the two points on the Earth's surface intersected by its axis of rotation. The imaginary line halfway between the two poles along the Earth's surface is called the Equator. The center of the Earth is regarded as the origin in this reference system. The location of a point is determined by the two angles:

- *Latitude*, which is the vertical angle measurement. It is the angle resulted from the intersection through the origin of two planes. The first plane contains the Equator, and the second plane contains the point in question. The Equator is considered to be zero-degree latitude. Points located north of the Equator are considered to have positive degree latitude, and points located south of the Equator are considered to have negative degree latitude.
- *Longitude*, which is the horizontal angle measurement. It is the angle resulted from the intersection through the origin of two planes. The first plane, called the Prime Meridian, is chosen as zero-degree longitude and contains the origin, the North pole, and the South pole (the Greenwich meridian in England is conventionally used as the Prime Meridian). The second plane contains the origin, the North pole, the South pole, and the point in question. Points located east of the Prime Meridian are considered to have positive degree longitude, and points located west of the Prime Meridian are considered to have negative degree longitude.

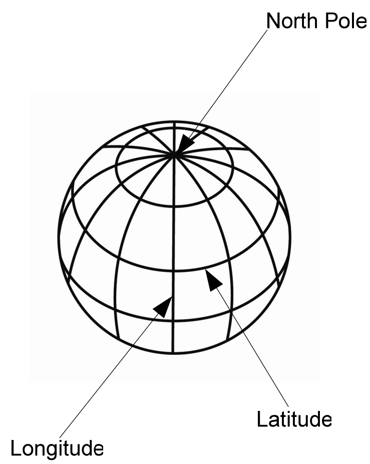


Figure 2.3 Graticule of Earth

In most GIS applications, there is a need to produce a map of the Earth on a flat, two-dimensional surface. A *map projection* is a systematic representation of all or part of the surface of a round body, especially Earth, on a plane [112]. Map projection can be *perspective* or *non-perspective*. Perspective projections use a point of origin and a surface of projection. The point of origin can be at the center of the sphere, at an infinite distance, or on the surface of the sphere. The surface of projection can be a plane, a cylinder, or a cone, each of which can be unfolded to a plane. Projections onto these three surfaces are called *azimuthal*, *cylindrical*, and *conic* respectively (Figure 2.4). This process is akin to shining a light source from the origin, through the sphere, onto the projection surface. The choice of the origin and the surface depends on the purpose and the requirements of the resulting two-dimensional maps. A non-perspective projection is obtained by mathematically modifying perspective projection such that certain properties, such as equal area, equal distance, or correct shape, can be maintained.

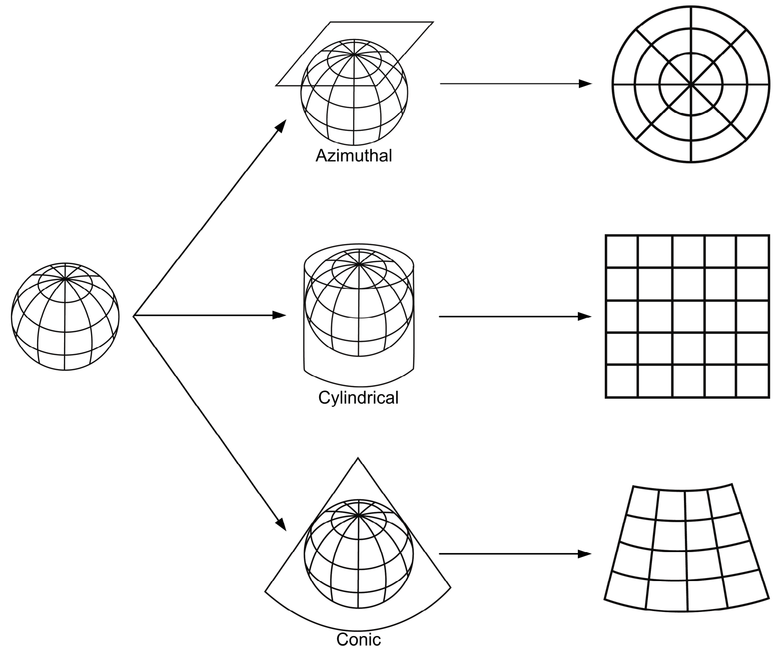


Figure 2.4 Map Projections

Additionally, *georeferencing* is defined as the representation of the location of real-world features within the spatial framework of a particular CRS. In practice, georeferencing can be seen as a series of techniques that transform measurements carried out on the Earth's surface to a two-dimensional flat surface of a map, and make it easily and readily measurable by means of a CRS. Fundamental to georeferencing is how the Earth is represented in terms of its physical shape, as it is found that the Earth is slightly flattened at the poles and is not a perfect sphere. The ellipsoid-geoid model is the commonly used mathematical surface that represents the shape of the Earth [15]. The ellipsoid is a reference surface for horizontal coordinates (e.g., latitude/longitude), and the geoid is a reference surface for vertical coordinates (elevation). The ellipsoid and the geoid, in the context of vertical and horizontal positions in georeferencing, are called datums, which are models that describe the position, direction, and scale relationships of a reference surface to positions on the surface of the Earth. Several datums have been calculated and defined, such as the North American Datum of 1927 (NAD27) and the World Geodetic System (WGS84), each based on different measurements of the physical shape of the Earth.

2.1.7 Geoprocessing

To solve geospatial queries, one or more computations are performed on geospatial data. Geoprocessing is the term that refers to such type of computation. Geoprocessing is at the heart of GIS as it is a collection of methods, procedures, and algorithms for solving geospatial queries. Geoprocessing may be carried out as part of data preparation, spatial analysis, and output generation steps [44] (Figure 2.5). Data preparation is needed because geospatial data are available from a variety of sources. As data are typically stored in different formats by different sources, there is a need to properly import, convert, and transform data into a form that can be used in a particular GIS. Spatial analysis includes a variety of geometric (e.g., distance and area) and topological operations (e.g., adjacency and connectivity) with the results used for decision making or as input for further analysis. Output generation includes the creation of presentation graphics making use of the results from the analysis stage or from a database.

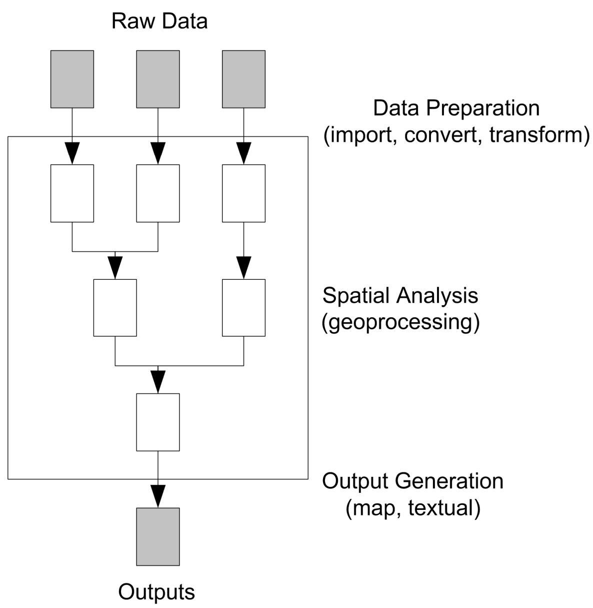


Figure 2.5 Geoprocessing data flow (after [44])

Conceptually, there are two types of geoprocessing corresponding to the computer representation models of geospatial objects and spaces: *vector-based* geoprocessing and *raster-based* geoprocessing. Modern GIS are capable of handling both types of data models and performing both types of geoprocessing, as well as converting data from one model to the other. Furthermore, many GIS applications simultaneously use both raster and vector data models and geoprocessing. A commonly used technique in cartographic production (mapmaking) involves a pictorial raster-based background overlaid by multiple layers of thematic vector-based data (e.g., a vectorized street network is overlaid on a satellite image).

Vector-based data preparation involves importation, conversion, and transformation of vector data into a form that can be used for analyses. Data importation and conversion entail syntactically translating one data format to another, while preserving the semantics of data. For the geometric component, data transformation is performed when the imported data are not in the map projection or CRS that is adopted by the particular project in which the data will be used. GIS software packages are normally capable of importing, converting, and transforming data and typically support many proprietary and interchange data formats as well as provide a large set of standards map projections and CRS. However, data transformation is a relatively complex process and requires knowledge on map projections or CRS.

The attribute component of vector data entails a three-step process [74]. The first step is to define the structure or schema of the data file. Normally, a relational database table is used and the structure is defined by specifying the characteristics of data in each column of the table. Each data item may be defined by its name, data type (e.g., character, numeric), and size. The second step is to populate the data file manually, interactively, or through an automated tool. The third and last step is to link the attribute data tables to the layers of geometric data, which is typically performed automatically by GIS.

There is a wide range of spatial analyses that can be performed on vector-based data. Basic attribute database query, where data are retrieved using a query language, such as the Structured Query Language (SQL), is often used as a means to obtain selected geographic entities according to a particular set of conditions. Computations can also be performed on data to obtain statistical information, such as frequency, sum, mean, minimum, and maximum. Calculations on geometric data are also frequently performed to attain geometric properties, such as distances between objects, areas, and perimeter of polygons.

An important vector-based geoprocessing is address *geocoding*, which is the process of assigning coordinates to locations described by addresses. For example, given the address "*135 N Bellefield Ave., Pittsburgh, PA 15213*", the resulting coordinates from the geocoding process are 40.45N latitude and -79.95W longitude. The objective of geocoding is to match given addresses to those addresses in reference databases (e.g., Topologically Integrated Geographic Encoding and Referencing system or TIGER); through interpolations and other matching techniques, precise coordinates of the addresses are computed [63].

Another frequently-used vector-based geoprocessing is *buffering*, which is the process of creating a buffer zone around vector objects (Figure 2.6). A buffer zone is a polygon and its size is specified by the user. Buffering is used primarily to evaluate the characteristics of an area surrounding a specific geometric feature. For example, real estate properties within a certain distance from a particular landmark can be identified through the use of buffering.

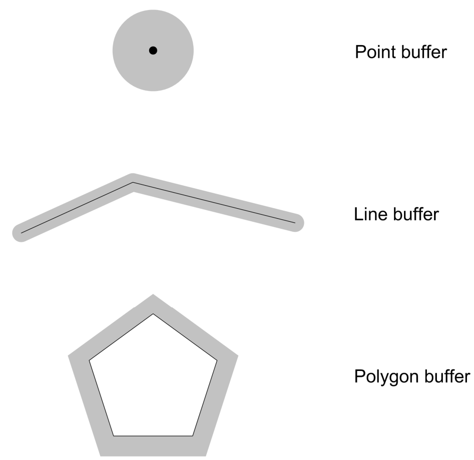


Figure 2.6 Point, line, and polygon buffers

Network analysis is another type of vector-based geoprocessing that can be employed for solving geospatial queries related to networks (e.g., roads, utilities, water pipes). *Routing* is a network analysis geoprocessing for computing paths in a network based on a particular cost function, for example, finding the shortest route from one location to another. Examples of networks in GIS that can be used in routing include streets, railways, and rivers.

Topological-based geoprocessing mainly consists of testing whether a certain relationship exists between two objects. For example, a “point-in-polygon” geoprocessing operation tests whether a given point lies within a given polygon. The result of this type of geoprocessing is a Boolean (yes/no) value stating whether the relationship exists between the two objects.

Raster-based geoprocessing in the data preparation stage often involves data format conversions and georeferencing, which is a transformation of measurements carried out on the Earth’s surface to a two-dimensional flat surface of a map, making them easily and readily measurable by means of a CRS. In some cases where the original measurement data contain distortions, errors, or are partially incomplete, geoprocessing is performed to rectify or restore the data. Raster data obtained through satellite imagery, aerial photography, and scanned maps often need this type of preparation before they can be used in spatial analyses.

Important spatial analyses for raster data include classification, which is the creation of a new raster layer by changing the attribute value of the cells of the input layer through logical or arithmetic operators [74]. For example, a classification can be obtained by applying a set of ranking criteria on a raster image (Figure 2.7). Classification is useful for simplification of raster images or categorization of geographic area, and is widely used in many applications.

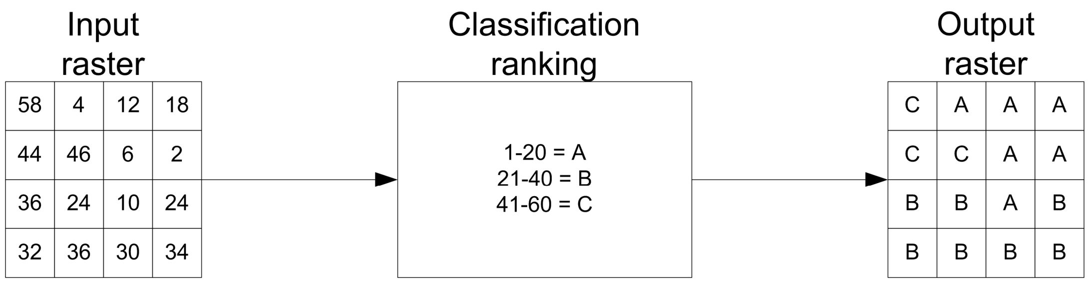


Figure 2.7 Classification ranking (after [74])

Another type of spatial analysis on raster data is where two input layers are used to generate a new layer through logical or arithmetic operators. For example, by performing arithmetic subtraction on two input layers, differences between the two layers can be detected. This operation is especially useful for detecting changes on a geographic area over time. Other raster-based spatial analyses include, but not limited to, filtering, which is a digital image-processing function for image enhancement [58], distance measurement, rotation, and translation.

The output generation of both vector- and raster-based geoprocessing deals with the production of computer display images and printer outputs. Results are generally presented as graphical maps containing multiple layers of thematic data. The scaling or zooming operation is performed by clipping out a specified area of the map and scales it to fit the display area. In the vector model, the zooming operation alters the types and instances of objects or entities displayed to the user [118]. A full-extent map of an urban area may show only major highways and important landmarks; after several zoom in operations, the map may also show local streets and buildings.

Table 2.3 summarizes examples of spatial analysis geoprocessing and describes how each one can be applied to solve geospatial queries.

Table 2.3 Geoprocessing operations for spatial analysis

Spatial Analysis	Application	Input	Output	Typical Query
Raster-Based				
Reclassification	Land-use Agricultural	Raw raster images	Codified raster images	Which areas have the most fertile soil?
Overlay analysis	Ecology Environment	Raw raster images	Overlaid raster images	Which areas of the forest have changed from last year?
Filtering	Land-use Agricultural	Raw raster images	Enhanced raster images	Highlight boundaries of farm parcels (i.e., display image with sharpened edges).
Vector-Based				
Database query	Urban management	Vector data with attribute values	A collection of identified objects	Find a list of every school in Pittsburgh.
Address geocoding	Address lookup	Reference data files, address to look up	Coordinates (may be displayed on a map)	Show a map centered at 135 N Bellefield Ave, Pittsburgh, PA 15213.
Buffering	Business	Vector data with attribute values	A collection of identified objects	Find a list of every hotel within 10 miles from the airport.
Pathfinding	Navigation	Vector data	A route (typically an ordered list of connected line elements)	Find the shortest-distance route from the airport to downtown.

2.2 GIS INTEROPERABILITY AND STANDARDS

Interoperability is a term that generally refers to the ability of two or more heterogeneous information systems to share information and procedures. A rigorous definition given by Brodie [10] is as follows:

> *Two components (or objects) X and Y can interoperate (are interoperable) if X can send requests for services (or messages) R to Y based on a mutual understanding of R by X and Y, and Y can return responses S to X based on a mutual understanding of S as (respectively) responses to R by X and Y.*

Interoperability can also be extended to include human interactions by considering that in order for a user to use the system, he/she must understand the concepts and terminology used in that particular information system or software.

2.2.1 Information Heterogeneity

The source of problems related to interoperability is *heterogeneity* in information systems [104] (Figure 2.8). In this view, platform and system heterogeneity are differences at the level of computer and communication systems and the software connection. Syntactic heterogeneity refers to the differences in formats and data types, whereas structural heterogeneity refers to differences at the level of schema, languages, and interfaces. As current information systems increasingly address information at the domain knowledge level, semantic heterogeneity – differences in the intended *meaning* – becomes increasingly important.

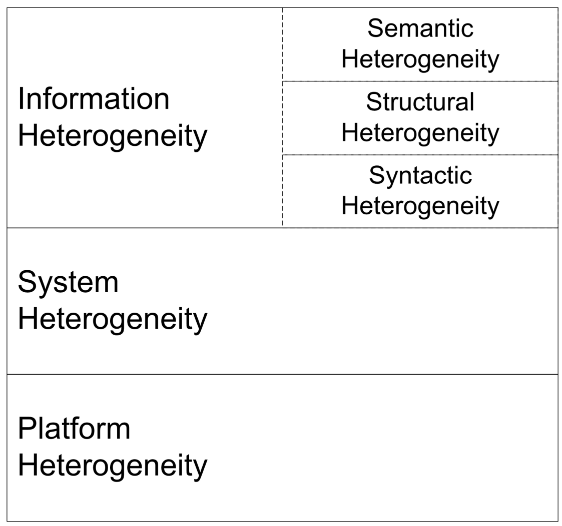


Figure 2.8 Heterogeneity in information systems (after [104])

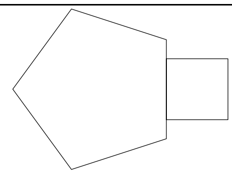
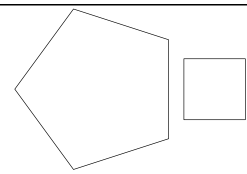
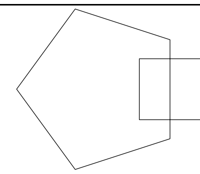
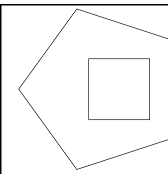
Semantic heterogeneity, a result of different conceptualizations and representations of real-world objects, is the source of semantic interoperability problems. Semantic heterogeneity exists because GIS in the past had been developed independently without taking into account the need to share and communicate between systems. Semantic heterogeneity can be distinguished into two types [8]:

- Cognitive heterogeneity, which arises when two groups of people from different disciplines conceptualize the same real-world facts differently. As an example, a geologist thinks of hill slopes as areas where soil erosion or landslides can occur, but a tourist manager may think of hill slopes as areas where skiing is [21].
- Naming heterogeneity, which arises when different names are used for identical concepts of real-world facts. Hill slope is also known as valley side, mountain flank, or simply slope.

Semantic heterogeneities manifest themselves in GIS, sometimes unexpectedly. Consider a case illustrated in [77] where different GIS platforms have different semantics for the same terminology. Table

2.4 shows the results of the “touch” operator when applied to two polygons on Oracle 9i Release 2 Spatial and Geomedia Professional GIS software packages (YES means that the two polygons satisfy the topological condition of the operator). The results demonstrate that the semantic “touch” for the one software is defined to give very different results from that of the other. These differences can lead to confusion and unexpected outcomes when performing spatial analysis.

Table 2.4 Results when the "touch" operator is applied to two polygons

				
Oracle Spatial	YES	NO	NO	NO
Geomedia Professional	YES	NO	YES	YES

Another example of semantic heterogeneity has to do with the mismatch of the level of semantic between a software system and a data source. In 1998, a car fell into a river [90] because the in-car navigation computer did not make distinction between a bridge, which is a permanent pathway, and a ferry, which is a transport carrying cars across rivers. The reason was hypothesized [102] to be that the navigation system inappropriately used data containing weak semantics (nodes, edges) when it should have used data with stronger level of semantics (roads, ferries) in computing the route and instructing the driver.

2.2.2 Interoperability Standards

A way to achieve interoperability is through a common agreement among all parties involved. This section discusses GIS-related interoperability standard-making bodies and the standards defined by them.

The Open Geospatial Consortium (OGC) is an organization consisting of companies, government agencies, and universities participating in a consensus process to develop publicly available geoprocessing specifications that result in interoperability among diverse GIS platforms. The creation of the OGC was due to the realization of several issues in interoperability among different systems, particularly when those systems are applied over the Internet. The development process of the OGC involves the creation of Abstract Specification [93] and Implementation Specifications [96]. The purpose of Abstract Specification is to create and document a conceptual model sufficient enough to allow for the creation of Implementation Specifications. Abstract Specification facilitates understanding of the real-world spatial phenomena being modeled. Each geographic feature type is classified in Abstract Specification and their descriptions include the hierarchies and relationships between objects as well as their behaviors, role names, and attributes. In contrast, Implementation Specifications are unambiguous technology platform specifications for implementation of industry-standard software Application Programming Interfaces (API). They are engineering specifications that implement part of Abstract Specification for particular distributed computing platforms.

The primary goal of Abstract Specification is to create and document a conceptual model sufficient enough to allow for the creation of Implementation Specifications. Abstract Specification consists of Essential Model and Abstract Model. Essential Model contains, in non-technical language, relevant facts consisting of real objects and events as perceived by the specification writers. It describes how the world works or should work and establishes the conceptual linkage of the software or system design to the real world. Abstract Model is a generic model of the software describing “ideal” software objects and events. It is expressed in technical language, including the use of Unified Modeling Language (UML) and describes how software should work in an implementation neutral manner. Abstract Model makes up the bulk of Abstract Specification and is divided into 17 topics with each one being worked on in parallel by a different Working Group of the OGC membership. A topic may be dependent on others.

Generally, the OGC has two central technology themes related to Abstract Specification: sharing geospatial information and providing geospatial services. The Open GIS Service Architecture, Catalog Services, Image Exploitation, and Image Coordinate Transformation Services are topics addressing the geospatial services theme of the OGC by providing detailed information on various geospatial services. The Open GIS Feature, The Coverage Type, and Earth Imagery Case are topics addressing the OGC theme of sharing geospatial information. They are fundamentally concerned with the handling and exposing of geospatial information. The remaining topics support the two OGC themes by addressing various aspects and requirements for effective sharing of geospatial information and providing geospatial services. These topics deal with coordinate reference systems, tools, and other miscellaneous items, such as data quality and application semantics.

Implementation Specifications provide programmers with specific programming rules and advice for implementing interfaces and protocols that enable interoperability between spatial processing systems. They are engineering specifications that implement part of Abstract Specification for particular distributed computing platforms.

*Si*mple Features Specification for OLE/COM is an Implementation Specification that provides GIS-specific interfaces to Microsoft’s Object Linking and Embedding (OLE) and Common Object Model (COM) technologies for accessing geospatial data. OLE/COM is a set of software models from Microsoft for accessing data from software systems. Simple Features Specification for CORBA and Simple Features

Specification for SQL are OGC standards providing comparable interfaces to the Common Object Request Broker Architecture (CORBA) and SQL, respectively. These Implementation Specifications have the same goal of integrating and leveraging accepted software standards for organizational-wide information access.

Catalog Services Implementation Specification supports the ability to publish and search collections of metadata for geospatial data, services, and related information objects. They are required to support the discovery of registered information resources. This OGC specification tailors catalog services to Abstract Specification. Coordinate Transformation Services Implementation Specification provides interfaces for general positioning, coordinate systems, and coordinate transformations in any number of dimensions.

The OGC Web Services (OWS) suite includes models for different types of information access services through the Web. The prominent one is the Web Map Service Implementation Specification (WMS), which specifies the interface for mapping service over the Web. The WMS specification defines three operations: GetCapabilities for obtaining service-level medata, GetMap for returning a map image, and GetFeatureInfo for returning information about particular features shown on a map. The specification also defines syntax for the Web Uniform Resource Locators (URLs) that invoke each of these three operations.

Geography Markup Language (GML) is an Implementation Specification designed to be a general data format language for modeling, transporting, and storing geospatial information. GML is an XML grammar written in XML Schema that provides a variety of kinds of objects for describing geography including features, coordinate reference systems, geometry, topology, time, and units of measurement. It is designed to be modular in that developers can choose to include only the schemas that are applicable to their works.

In GML, the state of a geographic feature is defined by a set of properties, where each can be thought of as [name, type, value] triple. Geometries in GML indicate the coordinate reference system in which their measurements have been made. A temporal reference system provides standard units for measuring time and duration. In GML, the Gregorian calendar with Coordinated Universal Time (UTC) is used as the default temporal reference system.

The Federal Geographic Data Committee (FGDC) is an interagency committee of the United States government composed of representatives from the Executive Office of the President, cabinet-level and independent agencies [31]. The responsibility of the FGDC includes development of the National Spatial Data Infrastructure (NSDI), defined as the technologies, policies, and people necessary to promote sharing of geospatial data throughout all levels of government, the private and non-profit sectors, and the academic community. Simply, the FGDC is responsible for the creation of policies, standards, and procedures for organizations to cooperatively produce and share geographic data.

An important standard defined by the FGDC is the Spatial Data Transfer Standard (SDTS). SDTS is a standard which specifies a way of transferring Earth-referenced spatial data between dissimilar computer systems with the potential for no information loss [126]. Compliance with SDTS is mandatory for federal agencies.

SDTS is organized into two main parts: base standard, and profiles. The base standard provides data standards at conceptual, logical, and physical or format levels [53]. It is intentionally flexible so to allow all models of geospatial data. At the conceptual level, the standard defines a catalog of spatial features and associated attributes. It includes definition of common geospatial terms (i.e., a vocabulary), such as “AIRPORT”, “PIER”, and “SHORELINE”. At the logical level, the standard explains spatial object types (e.g., point, arc, raster, and layer), data quality, conventions, and the layout of data modules within the standard. At the physical level, the standard explains the use of a general purpose file exchange standard, ISO 8211, to create SDTS file sets.

SDTS uses a profile to specify how the base standard must be implemented for a particular type of data. This allows flexibility in extending and making profile-specific changes to the base standard. Currently, there are standard profiles for topological vector data, raster data, and point data. There is also a standard application-domain specific profile for Computer Aided Drafting and Design. Other profiles, such as one for transportation applications, can also be defined in the future.

Another important standard from FGDC is the *Content Standard for Digital Geospatial Metadata* (CSDGM) which establishes the names of data elements and compound elements (groups of data elements) in a hierarchical manner for the purposes of providing a common set of terminology and definitions for the documentation of digital geospatial data [30]. CSDGM is an FGDC data standard that establishes the names of data elements and compound elements (groups of data elements) in a hierarchical manner for the purposes of providing a common set of terminology and definitions for the documentation of digital geospatial data.

2.3 ONTOLOGIES AND GIS

Semantic interoperability entails interoperations among systems as well as between users and systems in a meaningful way. To enable semantic interoperability, domain concepts need to be captured and used in an integrated manner with the underlying GIS. This section provides background and related work on ontological-based GIS.

2.3.1 Ontologies

Ontology is a concept encompassing many disciplines but has historically been confined to the area of philosophy. Recently, it has gained a specific role in branches of computer science such as artificial intelligence and database theory. Ontology is important in several research fields such as knowledge representation, database design, information integration, object-oriented analysis, and agent-based system design [49]. In information systems, ontology is becoming an important topic in the research area of interoperability; to permit information from distinct sources to be accessed, there must be agreement on the terminology in the shared area [131]. In the context of GIS, ontology can be used to specify semantic knowledge within the geospatial domain that includes a vocabulary of terms and their relationships.

In philosophy, Ontology (uncountable with capital “O”) deals with basic description of real things in the world, the description of what would be the truth. It is a branch of metaphysics concerned with the nature and relation of being [84]. On the other hand, the term ontology (countable with lowercase “o”) has two different senses assumed by the philosophy community and the Artificial Intelligence (AI) community [49]. In the philosophical sense, an ontology may refer to as a particular system of categories accounting for a certain view of the world. In this sense, it is language-independent (e.g., Aristotle’s ontology is always the same in any language). On the other hand, an ontology, as commonly used in the AI community and computer science in general, refers to an engineering artifact made up of specific vocabulary used to describe a certain reality with a specific set of assumptions on the intended meaning of the vocabulary words. A form of the first-order logic theory is usually used to represent these assumptions, with vocabulary words appear as unary or binary predicate names, respectively called concepts and relations. This type of ontology is sometimes called formal ontology.

The origin of ontology dates back to the time of Plato when a recurrent topic in philosophy was the relationship between thoughts, words, and things [78]. Plato dealt with the questions of the proper naming of things. In his view, the use of names in an “optimal world” would be to ensure that a particular expression will make everybody think of one and only one thing. However, he was doubtful that perfect names could ever be given, as things are continually changing. Aristotle later believed that to say *what* something is always requires saying *why* something is. His notion of definition was not simply the meaning of a word, but also the explanation of the “essence” of what a thing is. However, his notion was not sufficient as it did not address the limitation of communicating meaning via language, where ambiguities may be resulted from implicit exchange of different “senses” of meaning. To understand this ambiguity in meaning, Frege [39] introduced a distinction of two types of meaning: the *concept* and the *referent*. The graphical interpretation of this distinction, called the Meaning Triangle (Figure 2.9), was introduced by Ogden and Richards [97]. There are three key components in the triangle:

- *Concept*, which is an idea a person has inside his/her mind about the world.
- *Thing*, which is a thing in the real world that a person can physically sense.
- *Symbol*, which is a representation (e.g., a word) that a person makes of a concept.

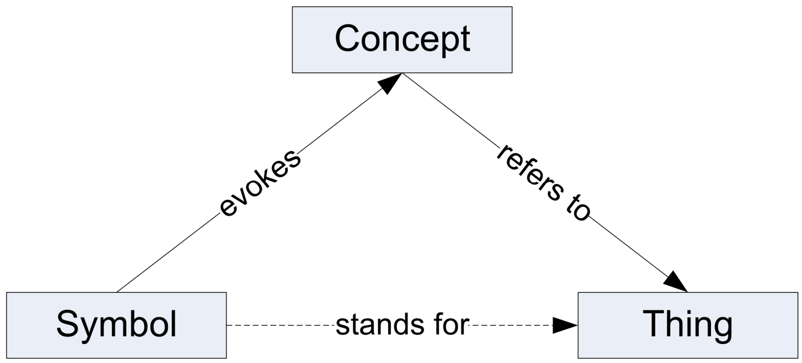


Figure 2.9 The Meaning Triangle (after [78])

The triangle illustrates that the relationship between a symbol and a thing is indirect. The link can only be completed when an interpreter processes a symbol, which invokes a corresponding concept and then links that concept to a thing in the world.

Knowledge typically makes a distinction between *intension* and *extension* [19]. Intension is knowledge and concept we have about a name. It contains knowledge about an object, a domain, a class, or information which describes and models a problem or an application. Extension is the thing a name refers to. It is the specific instantiation of a description or a model. In natural language, a description or specification of an object is an intension, whereas the actual entity for which the description is true is the extension. For example [20]:

'Venus' -> *second planet of Sun* -> *Venus*

'*Venus*' is a word that determines the intension "*second planet of Sun*", which in turns determines the extensional *Venus* object. In relational database, a schema is the intensional database, whereas the tuples of the database constitute the extensional database.

Functionally, an ontology is a specification of a conceptualization [47] that allows parties who agreed to an *ontological commitment* to communicate with one another and share knowledge. Ontological commitment is the agreement to use the vocabulary defined in an ontology as it is intended in a consistent manner. Ontologies in information systems are usually limited to a given domain. Their focus is not on knowledge and belief in general but rather on the ontological content of certain domain-specific representations [82]. In the field of information systems, work under 'ontology' was brought closer to the logic theory and became correspondingly more remote from relation to existence or reality. Some may argue that this is appropriate for a computer system as it defines the kinds of structures of objects, properties, events, processes, and relations that exist in the system. However, many are now arguing that the lack of grounding in external reality is the reason for problems involving legacy system integration [110] as older systems with different conceptual models but overlapping semantics need to refer to the common world in order to interoperate.

What constitutes an ontology can vary widely. A simple notion of catalog, in which each entity has a unique code associated with it, has been called an ontology. A more complex system could be based on natural language texts or a glossary which provides descriptions of terms and imposing some structure on them. A thesaurus is defined as "a controlled vocabulary arranged in a known order and structured so that equivalence, homographic, hierarchical, and associative relationships among terms are displayed clearly and identified by standardized relationship indicators…" [5]. A taxonomy is a classification of information entities in the form of a hierarchy, according to the presumed relationships of the real-world entities that they represent [19]. The entities in a taxonomy are related by subclass relationship with each entity distinguished by their properties or attributes. A taxonomy is usually depicted with the root entity on top as a node. Each node, including the root, has information about a real-world entity. Taxonomies are commonly used to browse or navigate information where the users only have a general idea of what they are looking for. An example of real-world taxonomies is the Dewey Decimal System [91], which is a widely used library classification system. A frame-based system provides, in additional to taxonomic structure, relations between objects and restrictions on what and how classes of objects can be related to each other. The most expressive ontologies use a set of general logical constraints to represent knowledge.

Ontologies are becoming an increasingly important research area in the field of geospatial information science [87]. For example, cognitive aspect of how humans perceived the geospatial world [113, 120] is an important research area that is related to geospatial ontologies. Other research efforts emphasizes how the real world should be modeled and formalized into ontologies [80, 82, 105-108]. Other recent efforts focus on ontological engineering methodologies for geospatial data integration [18, 28] and how those ontologies can be used for the purpose of processing by computers [13, 32, 71, 128, 130]. Most of these research efforts share a similar goal in achieving geospatial data integration at the semantic level.

The *University Consortium for Geographic Information Science* (UCGIS) stipulates research priorities related to geospatial ontology as follows [81]: a short-term (2-3 years) goal of developing and distributing an upper-level ontology for real-world geospatial phenomena that can be used as a common framework to ensure that independently developed sub-domain ontologies will be consistent and interoperable; a medium-term (3-5 years) priority that includes ontologies on vagueness and scale and works related to a better understanding of the cognition of geospatial; a long-term (10 or more years) goal of a complete formalization of the ontology of all phenomena at geographic scales.

On the other hand, efforts by the OGC focus on resolving syntactic and structural heterogeneities to provide a way for different GIS to interoperate in distributed environments. This includes standardization on geospatial schema [52], data format [17], and distributed geoprocessing architecture (e.g., [94, 95]).

Other recent efforts include the integration of semantics into GIS for the purposes of data sharing and semantic interoperability [40, 46, 51] and for solving application domain-specific problems [2, 29]. Research on distributed geoprocessing discusses the issue of semantic heterogeneity as well as the needs for mechanisms for users' query formulation and implementation of geoprocessing tasks over the Web environment [7, 23]. However, no clear-cut solutions to address these issues have yet been offered.

2.3.2 Types of Ontologies

Guarino [49] classifies ontologies according to the level of details and dependency on a particular task or point of view:

- *Top-level ontology* describes very general concepts. In the geospatial context, a top-level ontology may describe concepts related to space, time, geometry, topology, reference systems, and geoprocessing. It is reasonable in theory to have a unified top-level ontology for a large community of users.
- *Domain ontology* describes the vocabulary related to a domain. For example, a domain ontology would include concepts of land parcel and street networks, which are specializations of geometric and topological concepts in the top-level ontology.
- *Task ontology* describes tasks and activities. For example, navigation is a specialization of a geoprocessing operation in the top-level ontology.
- *Application ontology* describes concepts which depend both on a particular domain and a set of tasks. These concepts often correspond to roles played by domain entities while performing certain activities. Driving direction is a specialization of navigation specifically for driving.

An alternative division of ontologies specifically for the geographic domain was proposed by Fonseca [32]. In this approach, ontologies are divided into two types: *Phenomenological Domain Ontology* (PDO) and *Application Domain Ontology* (ADO).

PDO captures the different dimensions and internal properties of geographic phenomena. It is concerned with how geographic phenomena are captured and represented by computer systems and how algorithmic knowledge can be applied to them. PDO deals with measurement and intrinsic properties of objects and fields and operations that can be applied to them. It includes a measurement ontology describing the physical process of recording geographic phenomena and a method ontology describing algorithms and data structures that can be used on geographic phenomena. On the other hand, ADO contains domain-specific knowledge and is concerned with concepts and tasks specific to a particular application domain (e.g., ecology or geology) in which GIS can be applied to. ADO includes a subject ontology describing the vocabulary related to a specific application domain and a task ontology describing tasks and activities within that domain.

PDO and ADO are related by a semantic mediator, which performs two functions: identification and selection. Identification is the mapping between the concepts in ADO to the concepts in PDO. For example, the concept road in ADO can be identified with line vector object in PDO. Identification can also be performed in the other direction, from PDO to ADO. For example, points collected with GPS are in the realm of PDO. They need to be mapped to a concept in ADO in order for them to be meaningful (e.g., a collection of points collected from GPS represents the boundary of a university campus).

There are several reasons for separating ontologies into PDO and ADO. By separating conceptual knowledge of application domains from geographic representations, changes made on one would not affect the other. This is particularly useful as properties of objects and their relationships in real-world application domains are subject to change over time. For example, the boundary of a neighborhood in a city may be defined in one year using intersections and streets, which are points and lines; and in another year, the boundary of the same neighborhood may be defined as the circumference of an area defined by a radial distance from a landmark located at the center of the neighborhood. This redefining of the boundary changes the geometric description as well as the topological characteristics of the boundary, but still maintains the identity of the boundary and the neighborhood. Furthermore, by separating PDO from ADO, geographic representations and algorithmic knowledge can be reused across different application domains. For example, road in the transportation domain and river in the ecology domain are both mapped to line object. The geometric properties and algorithms applicable to line objects can be used equally for both road and river objects. Likewise, algorithms for spatial analysis on polygons, such as computation of area, perimeter, and point-in-polygon, can be used in the transportation domain as well as in the ecology domain.

Different ontologies can have different levels of details depending on the conceptualizations. A coarse ontology consists of a small number of axioms and is intended to be readily shareable and support core functionalities in an information system. A fine-grained ontology has a large number of axioms and is closer to specifying the intended meaning of a vocabulary. However, fine-grained ontologies are hard to develop and require an expressive language.

The level of ontology also depends on semantic granularity, which refers to the cognitive aspects involved in selection of features [35]. This distinctly differs from the concept of resolution in GIS which refers to the amount of details in a representation. Semantic granularity in GIS deals with variation in representation of geographic features across a wide range of scales. For example, a city may be represented as a single point on a map when it is perceived at a certain scale. The same city will, however, be represented with a more elaborate internal structure depicting streets, blocks, and building when it is perceived at a different scale. This is akin to the zoom operation described by Tanaka and Ichikawa [118]. High-level ontologies correspond to general information about geographic concepts, while low-level ontologies correspond to very detailed information.

Developers of information systems should create new ontologies based on existing ontologies whenever it is possible, as to make use of existing bodies of knowledge [32]. Within a specific application domain, developers can use high-level general ontologies to define their own lower-level ontologies that suit their needs. They can combine knowledge from different ontologies and create new knowledge based on the notion of *roles*. For example, “Building” in the urban ontology can be built from “Physical Object” in the high-level ontology. However, a building can also be considered a social entity and plays a role of “Organization” entity, in addition to being a physical entity. Thus, “Building” can inherit from both “Physical Object” and “Organization”.

2.3.3 Conceptual Framework

Fonseca [32, 34] provides a groundwork for ontology-driven GIS that includes a conceptual framework called the *Five-Universe Paradigm*, which is based on the *Universes Paradigm* [45]. This paradigm consists of the following five levels of abstraction (Figure 2.10):

- The first level is *Physical Universe*, which comprises actual real-world objects and phenomena. The universe is not limited to naturally-occurring objects but also artificial objects made by humans. Objects in Physical Universe are often perceived as being intertwine with the geographic space itself and are perceived as immovable, stationary geographical features [107].
- The second level is *Cognitive Universe* where images of the real-world objects are formed inside the human cognitive system, and through the process called *vision* [83], useful descriptions of those images are produced in the human mind.

- The third level, *Logical Universe*, includes organized formal definitions of the objects in Physical Universe obtained from the formalization of images and descriptions stored in Cognitive Universe. Ontologies about the geographic world reside in Logical Universe.
- At the fourth level, *Representation Universe* is where abstract and finite symbolic descriptions of the elements in the ontologies are made. Abstractions of real-world concepts at this level allow operations to be applied to them. The object and field conceptual models reside in Representation Universe.
- At the fifth and last level, *Implementation Universe* is where elements in Representation Universe are made into computer-based representations. Objects are represented as data structures (as vector or raster objects) and operations are represented as computer programs. GIS is Implementation Universe.

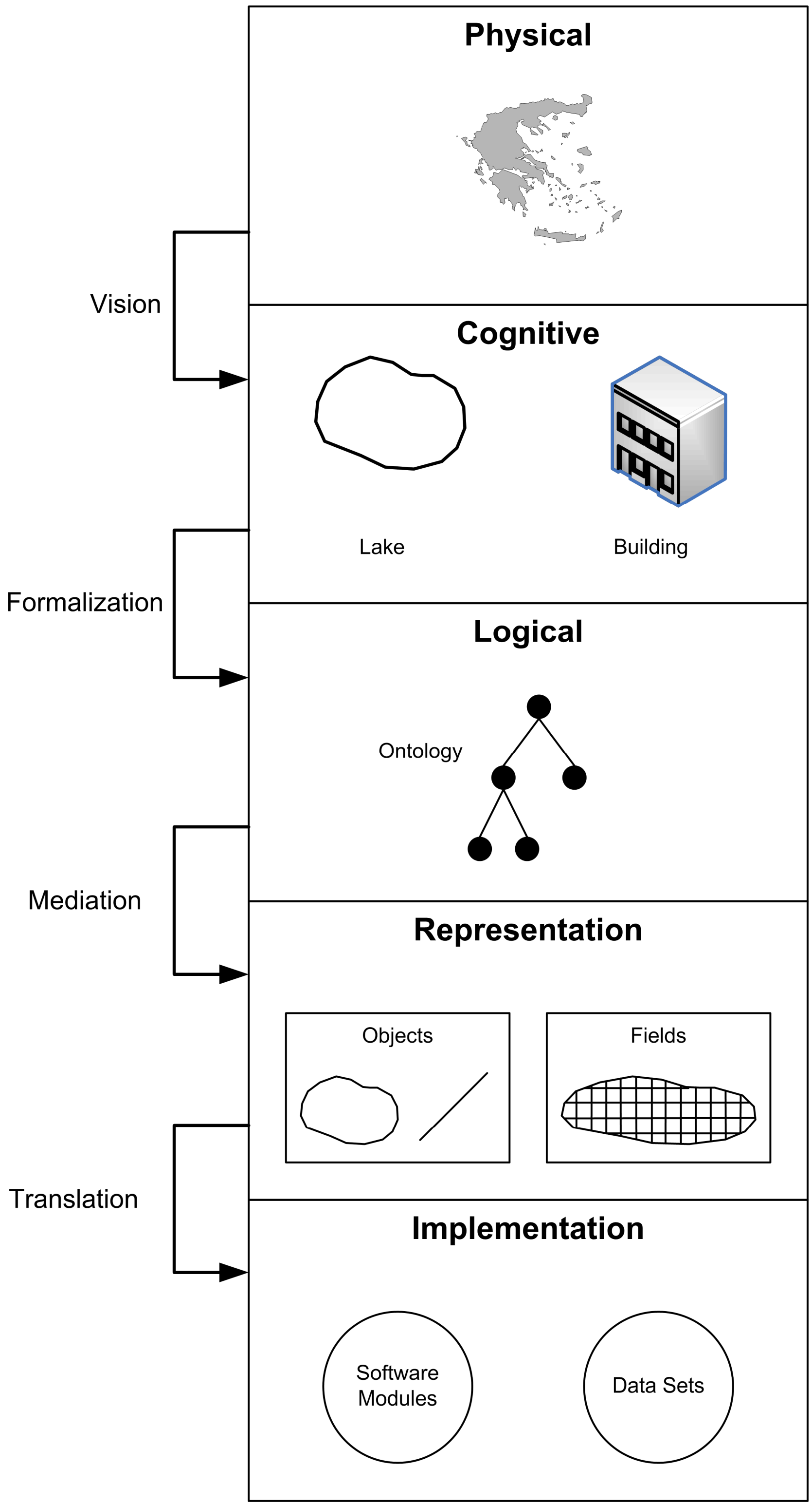


Figure 2.10 The Five Universe Paradigm (after [32])

2.3.4 Geospatial Cognition

Geographic objects are spatial objects on or near the surface of the Earth. Geographic entity types have some peculiarities that set them apart from other types of entities. This is due to the fact that geographic entities are not merely located in space, but are tied intrinsically to space in a manner that implies inheriting from space many of its structural properties [107]. Research on cognitive categories generally addresses entities in the "table-top" world. Such objects as birds, pets, toys, are of human scale and the *what* and *where* of those objects are almost always independent. In the geographic world, what the object is and where they are located are intimately intertwined. A geographic object characterized by a land mass higher up from the ground is not only identified as a mountain, but the fact that it is a mountain already connotes its location as it is conceptually a part of the geographic space itself. Categorization in the geographic world is also very often scale-dependent (e.g., is a pond merely a small ocean?).

In addition, geographic boundaries are themselves prominent phenomena for the purpose of categorization. Boundaries may be crisp or graded, and may be subjected to dispute due to ambiguity. They may be *bona fide* boundaries, which correspond to genuine physical discontinuities in the world; or they may be *fiat* boundaries, which are projected onto the geographic space independently from physical discontinuities and exist only through cognitive distinctions by human being [105, 108]. Shorelines and water bodies can readily be considered bona fide boundaries, whereas state and provincial borders are of fiat boundaries.

Common-sense conceptualization about the surrounding geographic world is referred to as *Naive Geography* by Egenhofer and Mark [26]. Naive Geography follows human intuition and establishes the link between how people think about geographic space and how to develop formal models of such reasoning that can be incorporated into GIS. Naive Geography assumes that the geographic space is two-dimensional because the horizontal and vertical dimensions are evidently decoupled in the human mind (e.g., people overestimate the steepness of slopes and the depths of canyons compared to their widths). Naive Geography also assumes that the Earth is flat since people disregard the Earth’s curvature when judging the distance between two points. This is evident by observing trans-Pacific travelers asking why their flight paths go all the way up over Alaska; the shortest path between two points across the surface of a sphere is not common-sense knowledge for most people. They also tend to base their perceptions on flat two-dimensional space of a map and have biases toward North-South and East-West directions (i.e., right angles).

Cognitive categorization of geographic objects was investigated [80] by experiments on non-expert English-speaking human subjects. These experiments were designed to elicit responses from the subjects to determine how they conceptualized the geographic world. Evidently, the term “geographical feature" only elicited natural geographic features and not artificial ones created by humans. When subjects were asked to list geographic features made by humans, each of them only listed a few and there was a low consensus, suggesting that this category lacks a clear core or essence.

Another experiment [106] revealed that there is a considerable mismatch between scientific geographers and non-expert people with regards to the assigned meaning of the term *geography*. In the experiment where the subjects were non-expert, the adjective "*geographic*" elicited almost exclusively elements of the physical environment of geographic scale or size, such as mountains, lakes, and rivers. On the other hand, the phrase "*things that can be portrayed on a map*" produced many geographic-scale, human-made artifacts, such as roads, cities, streets, and other fiat objects, which are generally considered *geographic* in scientific contexts [106]. Accordingly, *being geographic* was distinctly different from *being able to portray on a map*, as far as non-expert subjects were concerned. This evidence has implication on the usability and interoperability of GIS for non-expert users.

2.4 SUMMARY

This chapter provides relevant background information on GIS and related concepts and techniques. In addition, issues related to information heterogeneity in GIS and interoperability standards are discussed. Lastly, ontologies and their use in the context of GIS are discussed.

3.0 GEOSPATIAL QUERIES

In this research, geospatial queries are defined as those which concern concepts, objects, and activities related to some locations on the Earth's surface. These queries are formulated by users and, through some mechanism, posted to GIS for geoprocessing. Because geospatial queries deal with entities located near or on the surface of the Earth, they require Earth-referenced geospatial data sets representing those entities.

In this chapter, real-world geospatial queries are analyzed in order to obtain their general characteristics and commonalities. The objective of the analysis is to form a consistent conceptualization of geospatial queries and a representation that will allow for automated processing through algorithms.

3.1 FORMULATION OF GEOSPATIAL QUERIES

The main difficulty in analyzing geospatial queries is the lack of a general framework that describes how they are formulated and expressed by users. This is especially true when the queries are articulated in a natural (human) language with much implicitness. In the context of geospatial query solving using GIS, this implicitness does not pose a significant problem because of the existence of human GIS experts who, with much contextual knowledge, understand and correctly interpret the queries as intended by users. When ambiguities occur, GIS experts can resolve them through communications with users.

The goal of GeoInterpret is to allow users to express geospatial queries using application-level concepts and terminologies directly without needing to understand the intricacy of geoprocessing. Toward this goal, we need to determine the appropriate level of *explicitness* of user queries that would leave no ambiguity yet still maintain application-level (non-geoprocessing) context. The remainder of this section explores the process of user formulation of geospatial queries.

3.1.1 Phases of Geospatial Query Formulation

In our view, the formulation of geospatial queries is done in multiple phases, each with an increasing in the level of explicitness. Initially, the formulation of a query begins with an overall *goal* that a user intends to achieve. It is not unusual that the goal is *imprecise* even in the mind of the user at first;

however, its overall intent should remain unmistakably recognizable through the different phases of query formulation.

In the second phase, after the goal has been established, the user decides on a set of criteria that needs to be met in order for the goal to be satisfied. These criteria are in the realm of the decision-making process and are *not* within the context of GIS. Furthermore, the knowledge required for the conception of the criteria is well within the expertise of the application domain. In other words, the user needs to be well-versed in his or her domain in order to form an appropriate set of criteria. For example, in order to formulate a geospatial query about soil strength, the user must know the unit of measurement for soil strength and the sensible range that can be posed as queries.

Once the criteria have been established, they are made explicit in the third phase into unambiguous *GIS specification of requirements*. These are essentially a set of requirements of what the user wants from a GIS. If a GIS expert is involved in the process, these requirements are formulated interactively between the user and the expert. Accordingly, the knowledge needed to create the specification at this phase is both from the application domain expert (the user) and the GIS expert. The application domain knowledge is needed to articulate user's criteria as GIS requirements; while the GIS expertise is needed in order to know the capabilities and limitations of the GIS and whether the query at hand can be solved with the available resources and technology. In addition, the specification requirements at this phase must be *exact* in that there must be no ambiguity in them. For example, the statement "*Identify land parcels that are located near Lake Erie*" is not sufficiently exact because the term "near" in the statement has an ambiguous meaning and must be quantitatively clarified. On the other hand, the statement "*Identify the five nearest land parcels to Lake Erie*" is sufficiently exact because it needs no further clarification.

In the fourth and last phase, the specification of requirements is *interpreted* using GIS expert's knowledge into a sequence of geoprocessing operations. The GIS expert then carries out this geoprocessing workflow on GIS to obtain the solution to the user's query. Geoprocessing workflow produced in this phase is *independent* from the application domain as they are simply a set of operations to be invoked on GIS. Once the GIS expert understands the user's intent during the third phase of query formulation and is certain that the query can be solved using the available resources, then the user is no longer relevant in the fourth, workflow formulation, phase. The different phases of geospatial query formulation are summarized in Table 3.1.

Table 3.1 Phases in formuating geospatial queries

Phase	Context	Description	Scenario
1. Goal	Application domain	Goal to be accomplished	Find affordable lands near the School of Information Sciences that are best for building houses.
2. Criteria	Application domain	Criteria that satisfy the goal	Identify residential land parcels located within 5 miles of the School of Information Sciences building and cost less than $100,000.
3. Requirement Specification	Application domain and geoprocessing	Goal and criteria in the context of GIS	Create a map that shows land parcels located within 5 miles of the School of Information Science building, each residing entirely within a residential zone and costing less than $100,000. The map should contain the land parcels and streets.
4. Geoprocessing Workflow	Geoprocessing operations and GIS functionalities	Specific functions in GIS to solve the query	Import street network data, land parcel data, and zoning data for Pittsburgh, PA. Geocode the address *135 N. Bellefield Ave., Pittsburgh, PA* (or locate the position of the School of Information Science building). Perform a range selection of land parcel polygons located within 5 miles of the position found in the previous step. Select only those polygons having the *cost* attribute value less than $100,000. Select residential zone polygons from the zoning data. Perform polygon-in-polygon operation on selected land parcel polygons and residential zone polygons. Overlay the resulting polygons on street network. Produce a map output.

3.1.2 Formulating Queries with GIS

Having GIS platforms that are easy to use has been one of the objectives of the GIS community since its inception. In doing so, GIS researchers have focused on developing strategies to formulate geospatial queries. Figure 3.1 shows the different strategies for geospatial query formulation since the introduction of early GIS platforms. Early GIS utilized command-line interfaces and batch processing where users would have to specify geoprocessing workflows manually, corresponding to the fourth phase (geoprocessing workflow) of query formulation. Modern GIS are equipped with sophisticated *Graphical User Interfaces* (GUI), facilitating users in the formulation of geoprocessing workflows. However, although the GUI alleviate certain difficulties associated with learning the command-line interfaces, users still need to possess the GIS expertise in order to determine the geoprocessing operations needed to solve queries. Therefore, the GUI still operate at the fourth phase (geoprocessing workflow) of query formulation. In other words, solutions to geospatial queries can be obtained from GIS when the queries are formulated at the geoprocessing workflow phase. This is the current state-of-affair of general-purpose GIS software packages.

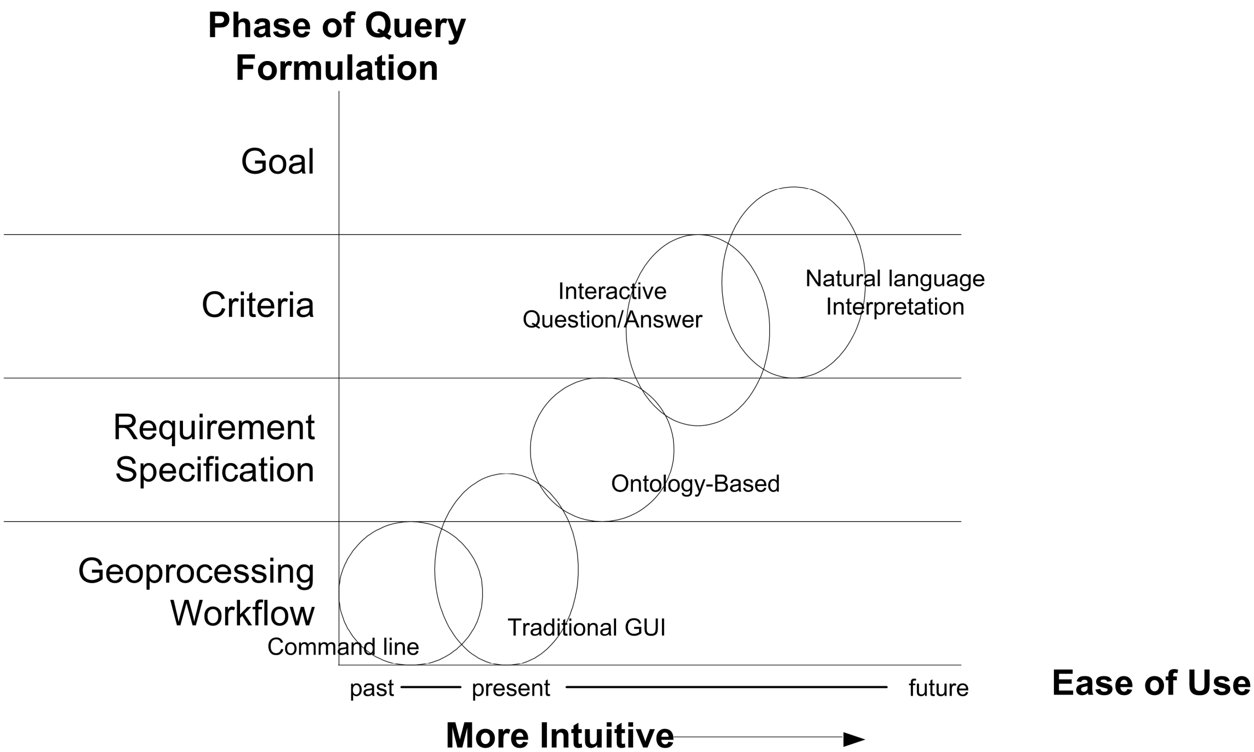


Figure 3.1 Query solving through GIS: past, present, and future

At the other end of the spectrum, natural language interpretation of geospatial queries would allow users to submit their queries in a plain natural (human) language. However, the mechanism to accomplish such an advanced interpretative capability does not yet exist and is potentially very complex and must be able to deal with the implicitness inherent in human languages. It also implies that a vast amount of contextual knowledge must be stored in the system in order to perform highly automated and intelligent interpretation. A similar, albeit less complex, mechanism in the form of questions/answers may also be used to extract the needed information from users through a series of interactions that would clarify any ambiguity that may exist in the queries. In either case, the mechanism would operate primarily at the second (criteria) phase of query formulation.

In this research, we argue that solutions to geospatial queries can be obtained automatically from GIS when the queries are formulated at the third (specification) phase, which comprises application-domain concepts and terminologies (no knowledge about geoprocessing operations is needed). Furthermore, we argue that the GeoInterpret methodology is the basis for enabling such level of automation through the use of ontologies.

3.2 CONCEPTUALIZATION OF GEOSPATIAL QUERIES

In the previous section, we identified the level of explicitness of geospatial queries that is needed for GeoInterpret. In this section, geospatial queries are further analyzed to determine their conceptualization, including their structure, components, and commonalities. A general conceptualization of geospatial queries is needed in order to conceive a computer representation for them. A computer representation for geospatial queries is needed in order for them to be processed algorithmically, thus allowing for automated interpretation and translation to geoprocessing workflows.

Sample geospatial queries from five application domains were extracted from the literature and experts in the respective fields (sample queries are summarized in the Appendix). The selected domains were *public health*, *business*, *ecology*, *construction planning*, and *transportation*. These domains were chosen based on their diversity, broad usage, and the availability and accessibility of the literature and experts in the fields. Because application domains overlap, the same query may be considered as part of more than one domain.

The extraction of geospatial queries from the literature proved to be difficult. There was markedly a lack of consistency on how queries were formulated in different publications. Queries were often written implicitly and/or verbosely. In most cases, a GIS-produced map was shown with a description of the map; however, the original query resulting in that map was not explicitly stated. Hence, the original query must be inferred from the map itself. Nevertheless, this observation was not surprising because geospatial query solving remains a relatively complex task requiring specialized expertise. The fact that the queries themselves were rarely stated concisely and explicitly confirmed the *ad hoc* nature of geospatial query solving and the lack of a framework for it.

3.2.1 Query Differences Across Application Domains

In this section, we examine common characteristics of geospatial queries within a *single* application domain. We then compare those commonalities across different application domains.

In the public health domain, geospatial queries are mainly concerned with making predictions or estimations based on past or present measurements of some observed incidences or some levels of quantities obtained through measuring instruments. The types of analyses performed on these data are generally done to uncover spatial clusters or patterns against demographic information. In addition, data analysis, through spatial interpolation and extrapolation, is often performed to estimate quantities at certain locations based on measured data. In the business domain, queries are usually formulated to aid in business operations and decision-making process based on past and current business activities. These queries are similar in nature to those in the public health domain in terms of pattern analysis and the use of demographic information. However, business queries frequently deal with selection of individual identifiable features and their attributes. In the ecology domain, the characteristic of queries that stands out from other domains is in the expectation of the use of large-scale, *raster* data obtained through satellite images. In addition, queries often concern temporal analysis or changes of features over a period of time. For construction planning, queries deal mostly with issues related to topology and attributes of discrete objects and land parcels surrounding the area of constructions. Unlike in ecology where typical geographic extents are very large (county- or state-wide), construction planning mainly deals with areas as large as a few city blocks. Lastly, geospatial queries related to transportation predominantly involve networks of pathways and require network-based geoprocessing, such as routing, block distance, and traveling salesman. Table 3.2 summarizes characteristics of queries across different application domains.

Table 3.2 Characteristics of geospatial queries in different application domains

	Public Health	**Business**	**Ecology**	**Construction Planning**	**Transportation**
General Characteristics	Predictions and estimations based on past or present measurements	Operational decision making; selections based on criteria	Scenarios and spatio-temporal analysis based on measurements	Topological analysis; scenarios	Linear-referencing analysis; scenarios
Types of Operations	Statistics; patterns and clusters analysis; demographic analysis; interpolation; extrapolation	Statistics; patterns and clusters analysis; demographic analysis; selections; attribute-based	Classifications; spatio-temporal analysis; patterns and cluster analysis	Topological relationship of points, lines, and polygons; attribute-based	Linear topological computations
Typical Data Models	Point data; Aggregates; census polygons	Point data; user-defined polygons	Raster-based analysis with point, line, and polygon data	Point, line, and polygon	Line networks with point and polygon data
Spatial Scale	City-wide; county-wide; state-wide	City-wide; county-wide; state-wide; nation-wide	County-wide; state-wide; nation-wide	City-wide; county-wide	City-wide; county-wide; state-wide
Temporal Scale	Days; months; years	Days; months; years	Months; years	Days; months	Hours; days
External Data Needs	Incidence and instrumental measurements; demographic	Business activities; demographic	Rasters of geographical areas; measurements	Urban plans; terrain; geotechnical	Street network; urban plans

3.2.2 Common Characteristics of Queries Across Application Domains

In the previous section, we examined the different characteristics of geospatial queries obtained from different application domains. In this section, we identify the *common* characteristics of all queries from all application domains. The challenge in this task lies in the diversity of geospatial queries as described in the previous section. For the purpose of this research, we identify the common *types* of geospatial queries as well as the common *components* of a geospatial query.

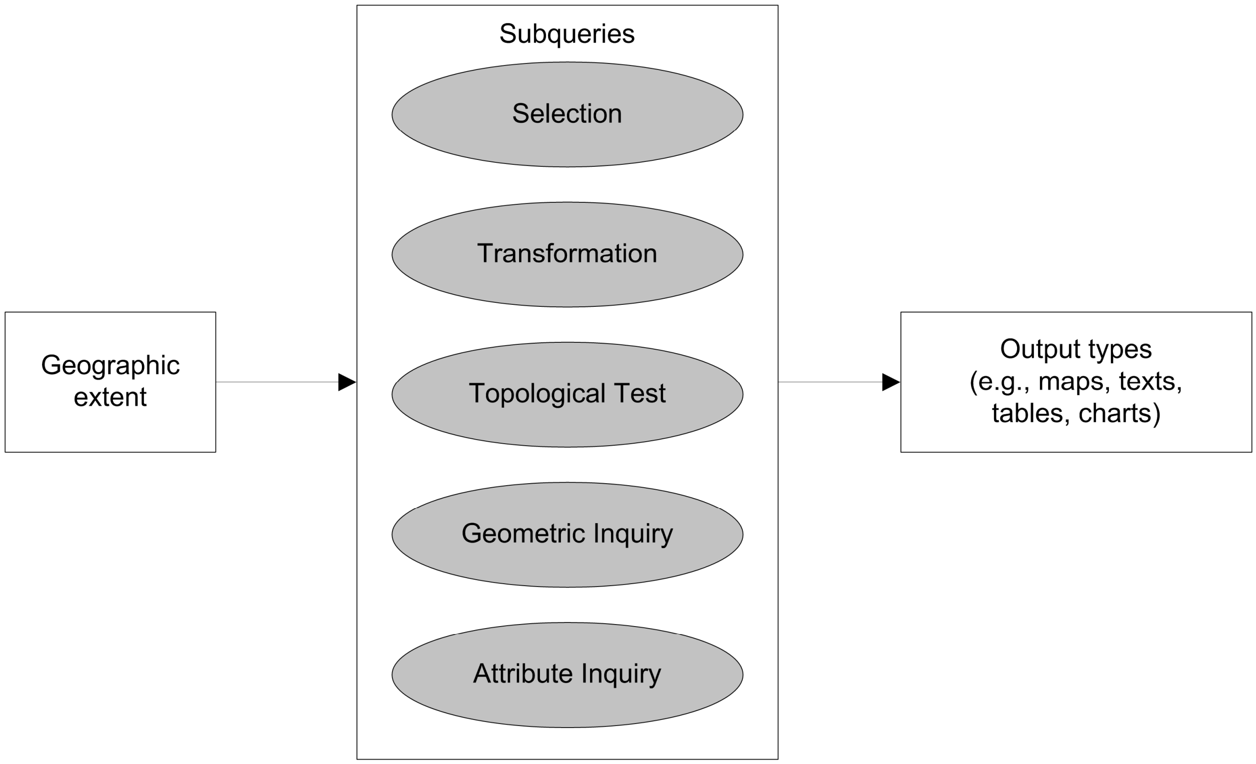


Figure 3.2 Common components and types of a geospatial query

Based on the examination of geospatial queries from various application domains, three common components of a geospatial query are identified: *geographic extent*, *subqueries*, and *output types* (Figure 3.2). The geographic extent is the region of interest of the query and is generally specified using one of the following methods:

- *Label*, which specifies the name of a place where its geographic extent can be determined through a database query (e.g., a gazetteer lookup).
- *Range*, which geometrically specifies a distance from a reference position. The reference position can be a label or another type of location identifier (e.g., street address).

In either case, the resulting geographic extent is the *boundary* of the area of interest. For example, specifying the geographic extent by label entails matching a given labeled name with a known coordinated boundary corresponding to that name (e.g., "Pittsburgh, PA" has a known geographic extent which is a two-dimensional area). Specifying the geographic extent by range entails spatial computation. For example, "*within 5 miles from 135 N. Bellefield Ave.*" is a statement that specifies the geographic extent of a query using range and requires geocoding and distance operations. In addition, geographic extent can also be defined through a graphical interface where users are allowed to draw a *geographic box* manually on a displayed map.

A geospatial query has one or more outputs that are presented to users as the outcome to the query. An output is usually a map containing geospatial features. However, users may also request an output to be presented as tables or charts containing information about the features. In many cases, multiple outputs of different types are required as the outcome of a single query.

At the heart of a geospatial query is a set of subqueries, which are the requirement specifications of the query, requiring geoprocessing. A query typically consists of multiple subqueries. Six types of subqueries are identified: *selection*, *transformation*, *topological test*, *geometric inquiry*, and *attribute inquiry*. The rationale for this categorization is based on the nature of queries from the point of view of users and not based on geoprocessing. The goal of this categorization is to conceive a general conceptualization of geospatial queries that is sensible. Each of these types of subqueries is explained in details in the following subsections.

In ontological modeling, a type of geospatial objects is referred to as a *feature class*. Within a geographic extent, a feature class may have many instances. For example, *Street* is a feature class. Within a city, there are many instances of *Street*, each with its own identifier. A feature class is not limited to physical objects, however; any concept which has an Earth-referenced location associated with it can be considered a feature class. For example, geospatial queries in the public health domain often deal with incidences of diseases that are spatially identified. Thus, incidences of a disease is considered a feature class.

A *selection* subquery specifies a set of conditions that identifies a subset of instances of a feature class within the geographic extent of a query. Conditions may be spatial or non-spatial. Instances of a feature class satisfying the conditions constitute the outcome to a selection subquery. For example, address geocoding is considered a selection subquery because it identifies an instance of a feature class (based on a street address). Instantiation of objects (direct retrieval from data sets) and filtering are considered a selection subquery.

Selection subqueries typically contain nested subqueries of some other types. For example, subquery "*Identify hospitals located within 5 miles from home*" is a selection subquery as its intent is to identify a subset of hospitals that satisfy a spatial condition. However, the condition itself ("*within 5 miles from*") is another subquery of the type *spatial inquiry*, described below.

A *transformation* subquery specifies that a new set of instances of a feature class be derived using spatial properties of existing instances within the geographic extent. The feature class of the derived instances does not need to be the same as the feature class of the base instances. Table 3.3 shows some sample transformation subqueries in the domain of environmental health.

Table 3.3 Samples of transformation subqueries in environmental health (after [9])

	To POINT	**To LINE**	**To AREA**
From POINT	Prediction of levels of air pollution at a location between two monitoring stations.	Estimation of the water quality of stream segments on the basis of data from monitoring sites.	Mapping of pollution distribution on the basis of data from selected sample locations.
From LINE	Prediction of levels of pollution at a location adjacent to a road network.	Extrapolation of water quality classes from classified segments to unclassified segments of the stream network.	Mapping of levels of pollution around a highway.
From AREA	Prediction of levels of pollution at a location within a mapped area of contamination.	Prediction of levels of pollution along a stream or road passing through an area of mapped contamination.	Interpolation of health data from one area base to another (e.g., from health districts to census districts to allow matching against population data).

A *topological test* subquery specifies that a binary test be performed on whether a particular relationship between two instances exists. For example, a query asking if a particular lake is located entirely within a county boundary is considered a topological test subquery. The outcome of a topological test is a Boolean (yes/no) value.

A *geometric inquiry* requests a computation on a geometric property of a set of instances of a feature class. The outcome to this type of subquery is a *computed* value. For example, a subquery asking for the distance between two positions is a geometric inquiry.

An *attribute inquiry* requests a retrieval and/or computation to be performed on attribute values of a set of instances of a feature class. The outcome to this type of subquery is an attribute value. For example, a subquery asking for the name of a spatially-identified building is an attribute inquiry. In addition, a subquery that does not involve instances of feature classes is also considered an attribute inquiry (e.g., computation of a statistical mean of a set of numerical values).

Table 3.4 summarizes the different types of subqueries according to their input and output. A transformation subquery requires instances of a geospatial feature class as input and produce instances of a geospatial feature class as output. A selection subquery always produces instances of a feature class as output, although they may be either an instantiation of objects or a result of filtering. Topological test, geometric inquiry, and attribute inquiry produce non-instance values (e.g., Boolean, numerical values, attribute values) based on properties of instances of a feature class.

Table 3.4 Types of subqueries organized based on their input/output objects

	Feature class instances as output	Non-instances as output
Feature class instances as input	Transformation Selection	Topological Test Geometric Inquiry Attribute Inquiry
Non-instances as input	Selection	Attribute Inquiry

To demonstrate the different types of subqueries, consider the following query:

Create a map that shows land parcels located within 5 miles of 135 N. Bellefield Ave., Pittsburgh, PA, each residing entirely within a residential zone and costing less than $100,000.

The geographic extent for this query is "Pittsburgh, PA", which defines the boundary of the city. Although the actual geographic extent is the 5-mile area around the addressed location, the area must be spatially computed as subqueries. On the other hand, "Pittsburgh, PA" gives us the general region of interest.

The output to the query is a map containing land parcels (and implied a backdrop to give a geographic context). The outermost subquery, which is "*Create a map that shows land parcels…*", is a *selection* subquery because it specifies the objects for display to be a subset of all land parcels within the geographic extent. The conditions of the selection, represented by the remainder of the query, are made up of nested subqueries of various types (re-written for clarity):

- "*located within 5 miles*" is in fact composed of two subqueries. To see how this is the case, consider a re-phrasing of the subquery: "*Is the distance from X to Y less than 5 miles?*" In this form, the first subquery is a geometric inquiry that asks for the Euclidean distance between two points; while the second subquery is an attribute subquery that compares the distance value resulting from the first subquery against a fixed quantity of "*5 miles*". The result of the second subquery is a true or false value that is used as a condition for selection (i.e., if true, select the instance).
- "*135 N. Bellefield Ave.*" identifies a location within the geographic extent based on a street address. Therefore, it is considered a selection subquery.
- "*residing entirely within zone*" is a topological test subquery because it requests that a test be performed on two instances of feature classes. This subquery is different from the first subquery ("*located within 5 miles*") because the first one specifies a fixed quantity; however,

topological computations are independent from geometric quantities (i.e., topology remains invariant even if the space is deformed [52]).

- "*residential zone*" consists of two subqueries. The first one is a selection subquery because it requests a subset of zone instances where each one is of type residential. The second subquery is an attribute inquiry because it requests for a retrieval of an attribute value to indicate the type of zone (i.e., “residential”) of each instance for use as condition for selection.
- "*costing less than $100,000*" consists of two attribute inquiries. The first one requests for a retrieval of an attribute value (cost) and the second one asks for a numerical comparison against a fixed quantity of “$100,000”.

3.3 SUMMARY

This chapter provides an analysis of geospatial queries based on existing queries in the literature. The objective of the analysis is to uncover commonalities among geospatial queries and to conceptualize them as a structure containing various components. The components of geospatial queries are identified in this chapter and they include a set of nested subqueries as a core component. In addition, different types of subqueries are identified and described.

4.0 GEOINTERPRET METHODOLOGY

4.1 INTRODUCTION

GeoInterpret consists of techniques and algorithms that make up a coherent methodology for automating the formulation, interpretation, and translation of geospatial queries to workflows consisting of geoprocessing operations. As previously discussed, this interpretation of queries is the most important phase in geospatial query solving and is currently performed by GIS experts, requiring them to possess significant specialized knowledge and skills which are generally obtained through formal training and extensive hands-on experience. Current GIS technology does not support an automated way to perform such a task, involving interpretation and translation of query elements from application-level semantics to geoprocessing-level semantics. In order to automatically interpret and translate queries, contents at different conceptual levels of geospatial query solving must be captured. In GeoInterpret, these contents are captured as formal ontologies.

4.1.1 Conceptual Approach

The conceptual framework for geospatial query solving in GIS was described by the Five-Universe Paradigm [32], which was discussed in Chapter 2. In this paradigm, an ontology in Logical Universe is the formalized knowledge of users' understandings of the real-world; and through a semantic mediator, elements contained in it are associated with abstract representations of objects in Representation Universe.

In GeoInterpret, we extend the Five-Universe Paradigm further by explicitly stating that multiple formal ontologies, each representing different conceptual levels, exist in Logical Universe. This multiple-ontology approach is based on the division of ontologies by Fonseca [32]. In GeoInterpret, one ontology is to capture the real-world, application-specific domain concepts as well as real-world geospatial concepts (generally referred to in the literature as “geospatial ontology”). The purpose of this ontology is to represent the knowledge necessary to *understand* geospatial queries (i.e., the knowledge for interpreting queries); hence, this ontology also captures the conceptualization (e.g., structure and components) of geospatial queries at the level understood and used by non-expert users. The second ontology, on the other hand, captures knowledge related to geospatial data models (e.g., vector and raster objects) and geoprocessing operations (e.g., geocoding), which are inherently independent from application-level knowledge defined in the first ontology.

In GeoInterpret, the former ontology for representing application-level domain knowledge is named *Geospatial Domain Ontology* (GDO) and the latter ontology for representing geoprocessing-level knowledge is named *Geospatial Processing Ontology* (GPO). The two ontologies are related through the *Ontological Mediator* which contains mappings between elements from GDO to GPO (Figure 4.1). In this conceptual approach, GDO represents the knowledge at the level of application-domain users who are experts in their respective fields (e.g., an ecologist), while GPO represents the knowledge at the level of GIS experts who may not possess the application-level knowledge (e.g., GIS expert who is not an ecologist). Ontological Mediator defines the mappings between elements in the two ontologies and hence, contains knowledge required and employed by both application-domain experts and GIS experts.

Following the Five-Universe paradigm, GDO is the result of the formalization of elements in Cognitive Universe. In other words, the world of an application domain, as conceptualized by users in that domain, is formalized as GDO. Similarly, GPO is the result of the formalization of elements in Representation Universe. Concepts understood by GIS experts, such as the object and field models and operations applicable to them, are formalized as GPO.

The task of solving geospatial queries posed by an application-domain expert (who is not a GIS expert) must be carried out by a GIS expert who knows the intricacies of geoprocessing operations supported by GIS. This task generally involves communications between the two parties in order for the GIS expert to understand the needs of the application expert. In GeoInterpret, this knowledge of understanding queries is resided in Ontological Mediator. Hence, by implication, the construction of Ontological Mediator must be performed with cooperation from both the application-domain experts and GIS experts.

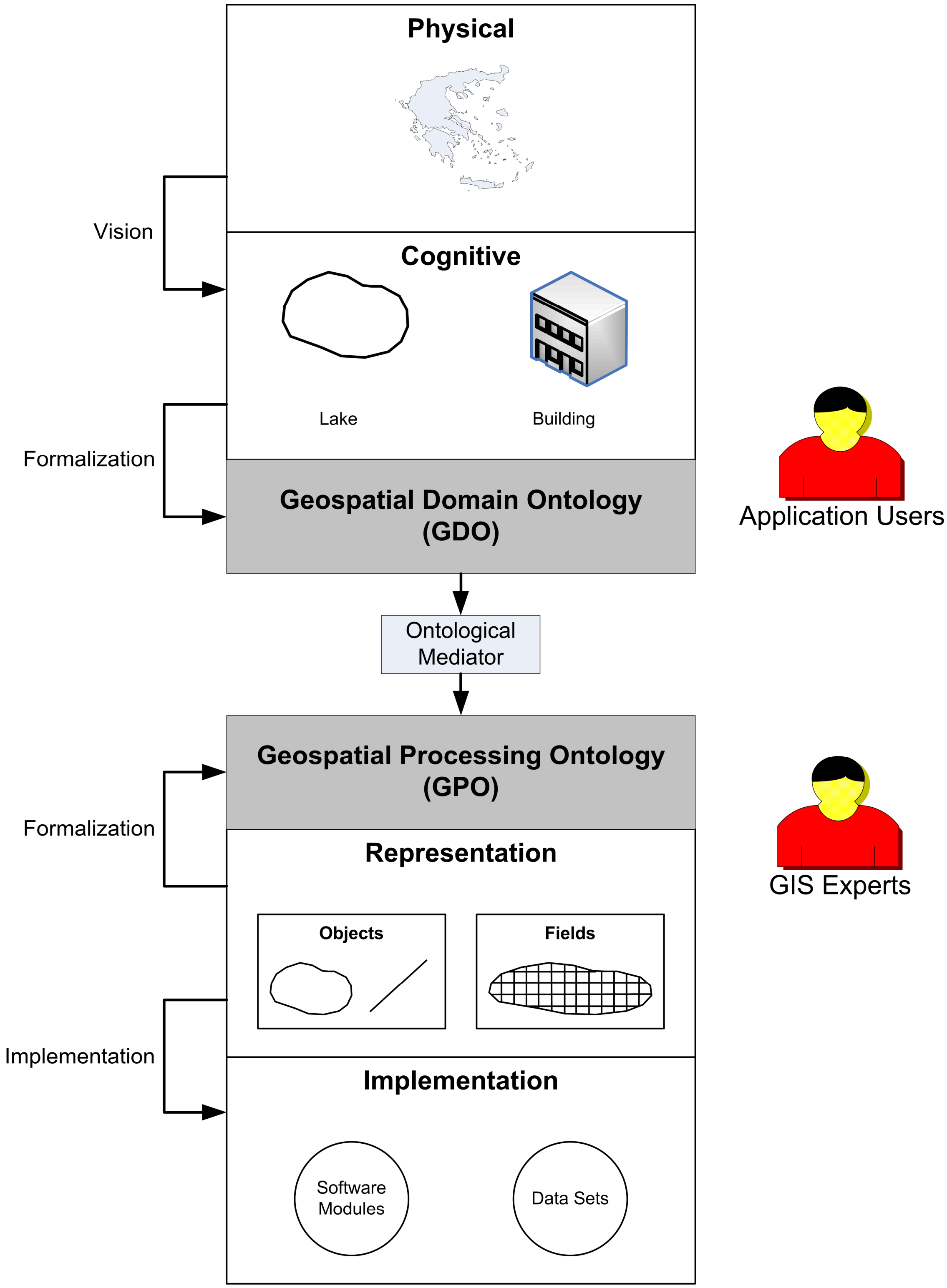


Figure 4.1 GeoInterpret approach to the Five-Universe Paradigm

The multiple-ontology approach used in GeoInterpret has several advantages. First, because the knowledge on geospatial data models and geoprocessing defined in GPO is inherently independent from application domains, it can be developed independently and be reused for many application domains. Second, the separation of application-specific domain ontology from geoprocessing ontology allows application specialists who have no expertise in GIS to develop their own conceptualizations of the world without having to deal with the complexity of geoprocessing. This approach is different from others [6, 66] in which no clear distinction is made between GIS experts and application-domain experts.

GeoInterpret provides a top-down methodology, from formulation of queries, interpretation of queries, mapping of query elements to geoprocessing tasks, to creation of geoprocessing workflows. GeoInterpret does not rely on specific implementation environments (e.g., geoprocessing services in the Web environment [7]) and thus can be adapted to different implementation strategies. In summary, GeoInterpret is an integrated methodology for utilizing ontologies as the knowledge base to perform the specific tasks of interpreting and translating geospatial queries to geoprocessing tasks.

4.1.2 Challenges

Uschold and Jasper [123] argue that "*An ontology may take a variety of forms, but it will be necessarily include a vocabulary of terms and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms*." In GeoInterpret, ontologies are used to impose a structure and constraints which make formulation of geospatial queries possible; hence, their interpretation. The ontological knowledge at the user level (GDO) is the knowledge on how geospatial queries are formulated in a user-level, application-domain context. The ontological knowledge at the geoprocessing level (GPO) is the knowledge on geometrical representations, geoprocessing operations, and constraints on how those operations can be applied to geometrical representations.

The knowledge on how application-level queries are mapped to geoprocessing tasks is defined in Ontological Mediator that bridges the two ontologies and is one of the core components of GeoInterpret. Toward the objective of designing Ontological Mediator, several challenges must be met. First, there must be an explicit and precise representation of geospatial queries in a form that will allow for computer processing. Second, there is a need for a knowledge representation scheme that makes connection between elements defined in GDO and elements defined in GPO in such a manner that their structures are not significantly affected (i.e., application domain specialists should not have to make significant compromises on their GDO in order to make use of GPO and vice versa). In other words, the complexity in semantic mapping between the two ontologies should reside in Ontological Mediator, which includes knowledge by both GIS experts and application domain experts. Lastly, a set of algorithms that employ GDO, GPO, and Ontological Mediator to perform query formulation, query interpretation, and mapping to geoprocessing operations is needed.

4.1.3 Scope

The main contribution presented in this dissertation is not a specific implementation strategy; rather, it provides a new approach that is query-based and is a general methodology that can be used as a basis for future research that may include implementation aspect. Additionally, the method for query submission (i.e., user interface), which is in the field of Human-Computer Interaction (HCI) research area, is not directly addressed by GeoInterpret. One reason for this limitation is that the interaction between users and GIS is generally dependent on the implementation context as well as the application that utilizes the GIS. For example, GIS utilized by mobile users require a different approach in user interface design from that of desktop-based GIS. However, utilizing GIS under both environments eventually boils down to the common conceptual task of solving geospatial queries.

Since geospatial analyses are predominantly vector-based (often raster data are converted to vector data prior to analysis), vector-based geoprocessing is the focus of this research. Future research related to GeoInterpret could include raster-based geoprocessing.

4.2 COMPONENTS OF GEOINTERPRET

The components of GeoInterpret comprise three main algorithms: *Q-GET*, *Q-ANALYZE*, and *W-COMPOSE*, which employ GDO, GPO, and Ontological Mediator (Figure 4.2).

GeoInterpret conceptually situates between a user interface and an implementation platform (e.g., geoprocessing tools and/or services) and thus is a platform-independent methodology. Q-GET is an algorithm for ontology-based formulation of geospatial queries that allows them to be formally represented, which is a requirement for queries to be processed by computers. Queries in formalized form are then passed on to Q-ANALYZE, which analyzes and maps elements in queries to appropriate geospatial data models and geoprocessing operations. In the last step, W-COMPOSE, the resulting geoprocessing tasks are combined to form geoprocessing workflows. The entire procedure utilizes GDO, GPO, and Ontological Mediator to determine how queries are formulated, interpreted, and mapped to geoprocessing operations and to create workflows.

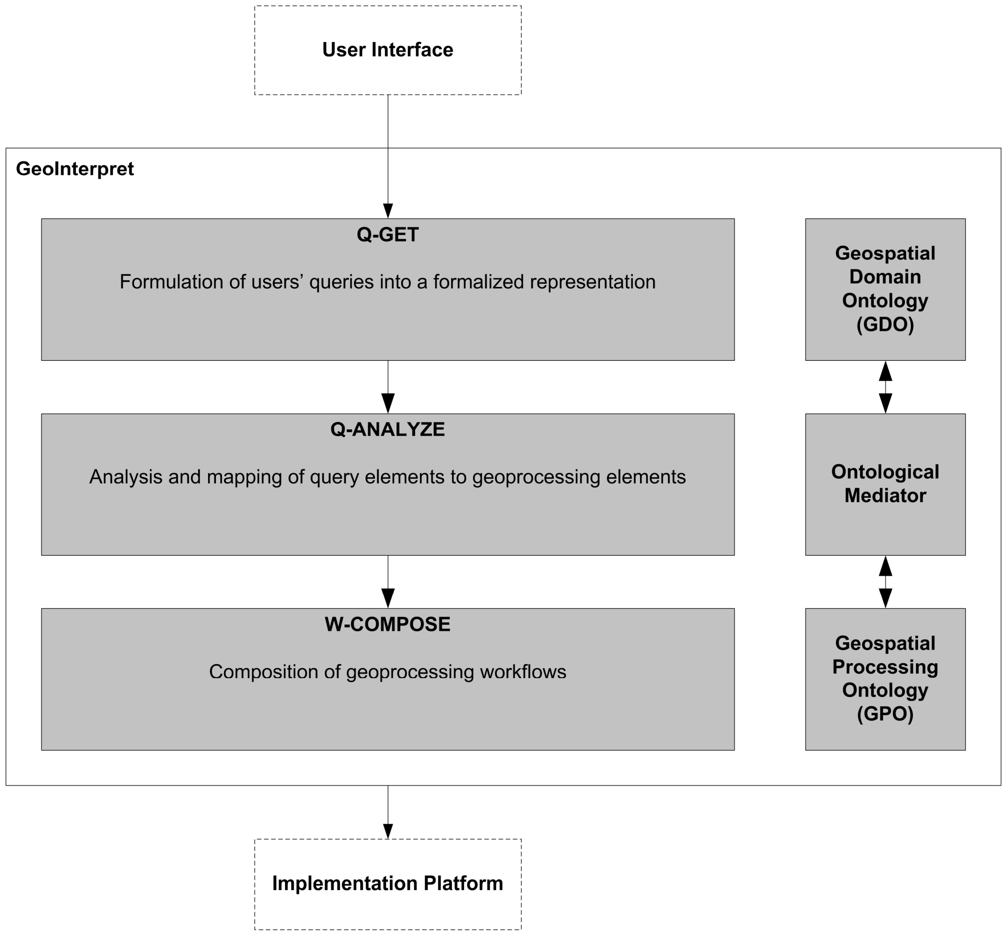


Figure 4.2 Components of GeoInterpret

Geoprocessing workflows produced by W-COMPOSE are *logical* workflows in a sense that they identify the sequences of geoprocessing operations that are needed to solve queries without addressing implementation-level issues, such as workflow optimization, geoprocessing service discovery, geoprocessing algorithms, or other platform-specific requirements. This approach provides the flexibility in using the resulting workflows in different implementation strategies. For example, workflows can be used as instructions for users to solve geospatial queries using GIS software packages. Another novel approach involves the use of workflows for creation of *implementation workflows*, which contain platform-specific instructions that also include additional operations for geoprocessing resource identification, discovery, and retrieval over heterogeneous networked environments. The Q-GET, Q-ANALYZE, and W-COMPOSE algorithms are discussed in Chapter 5.

4.3 GEOSPATIAL DOMAIN ONTOLOGY

Figure 4.3 shows GDO, the knowledge base at the level of application domains. The approach taken in building GDO in GeoInterpret is to have a single geospatial ontology that contains the general real-world geospatial knowledge that is common to all application domains. This is similar to what is commonly referred to in the literature as the “top-level ontology” [48].

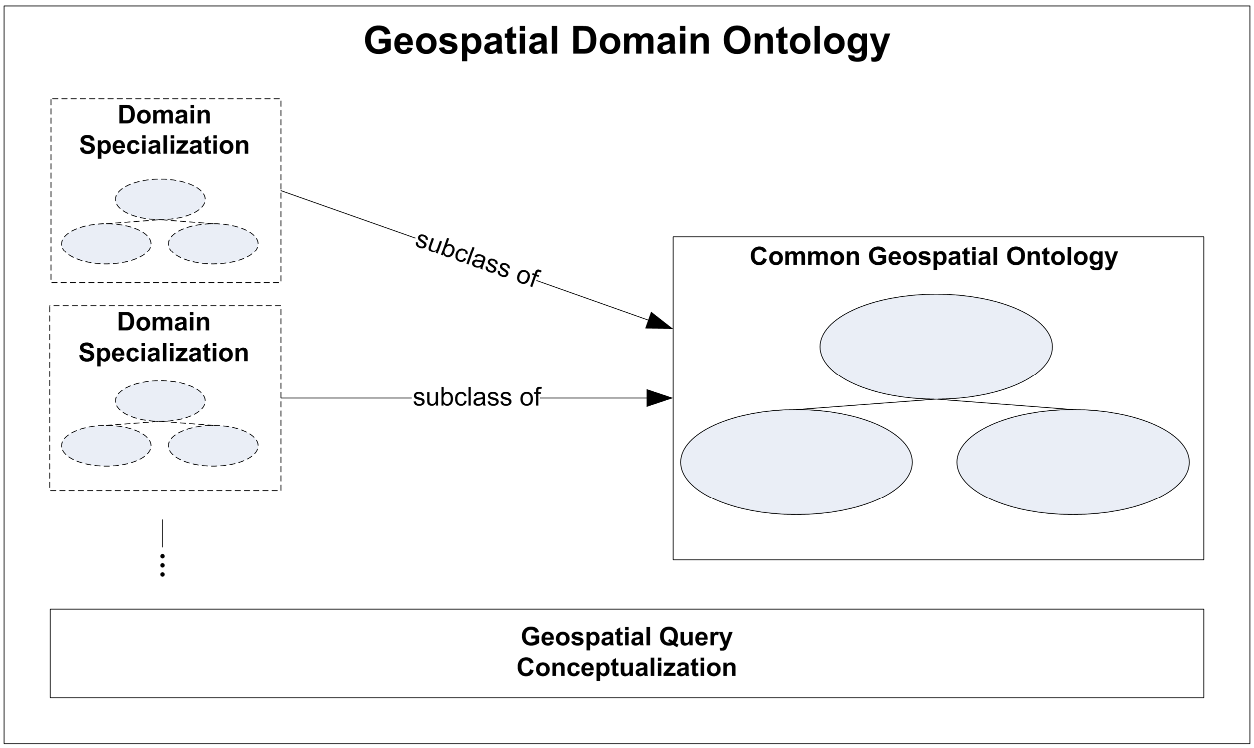


Figure 4.3 Knowledge in GDO

In order to make GDO specific to an application domain, specialization through inheritance can be used. For example, concepts and terminologies specific to the field of ecology can be specialized from the common geospatial ontology by subclassing elements from the common ontology. Other ontological engineering techniques (e.g., [18]) may also be used in the creation and management of GDO, but they are beyond the scope of this research.

Furthermore, GDO also contains the conceptualization of geospatial queries (and subqueries), which includes subquery components, types, and how they are constrained and represented. This knowledge is needed to understand and interpret queries. How queries are constructed and represented are discussed below.

4.3.1 Entities

As with traditional ontologies, GDO defines the vocabulary of a domain. This includes objects, terms, their properties, and relationships (usually hierarchical) among them. In GeoInterpret, we called these objects *entities.* The relationship between entities in the common geospatial ontology and domain-specialized ontologies is through inheritance (Figure 4.4).

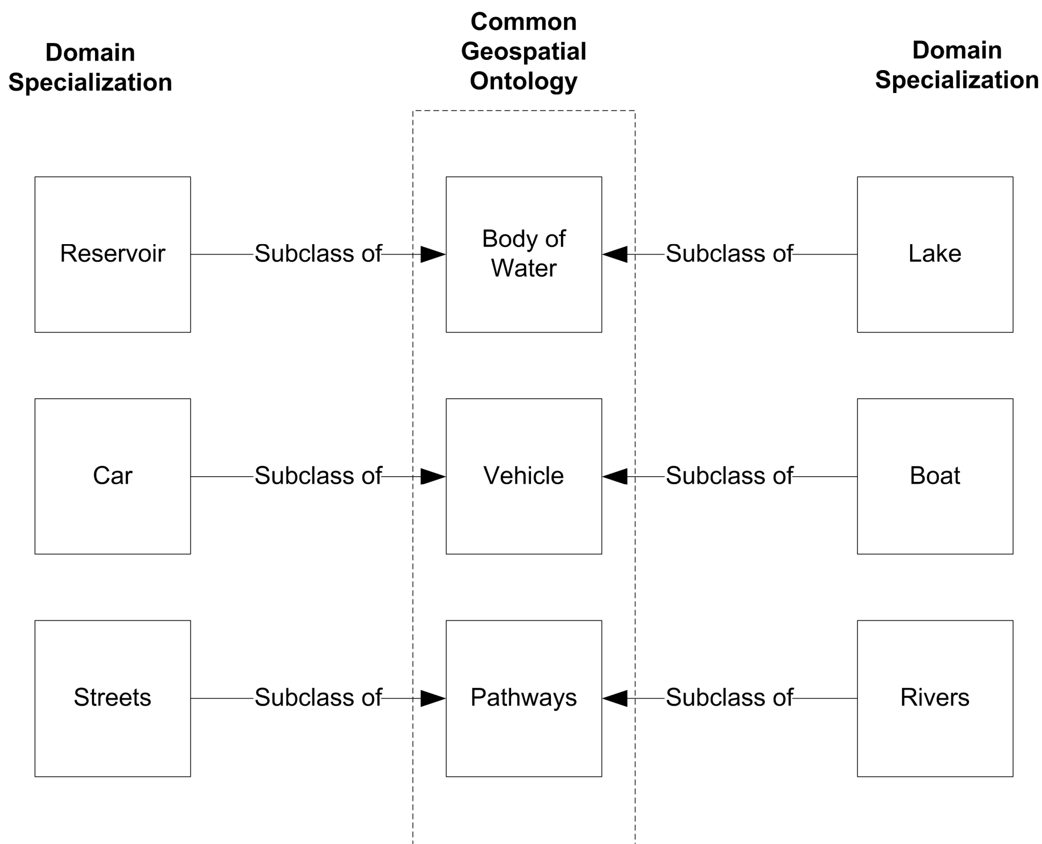


Figure 4.4 Specialization of common geospatial ontology

4.3.2 Actions, Roles, and Affordance

In the context of geospatial query solving, we also apply the notion of *action* [13, 71, 102], *role* [3, 116, 117] and *affordance* [42, 43, 89] to GeoInterpret. The use of actions in the context of ontological engineering has been proposed in the literature [13, 71] as a way to capture human activities and users' intentions. In GeoInterpret, which deals with the *domain of geospatial queries*, actions are subqueries that make up geospatial queries, i.e., different types of actions represent different types of subqueries.

The use of roles has been proposed and studied in the literature for conceptual modeling and object-oriented programming [117] as well as for ontological engineering [50] including bridging different ontologies [32]. A role can be viewed in various ways [116], e.g., a named relationship, a specialization, a generalization. On the other hand, affordance refers to the *actionable* properties between two concepts and has been discussed in the geospatial domain [60, 70]. In its simplest form, affordance connects entities to actions (e.g., "a bridge affords crossing a river") as shown in Figure 4.5.

Figure 4.5 Entity and action

In GeoInterpret, we use the notion of role to connect entities to actions. The general relationship between them is shown in Figure 4.6. The use of roles to determine the actionable properties of entities has been successfully applied to the area of computer security [88].

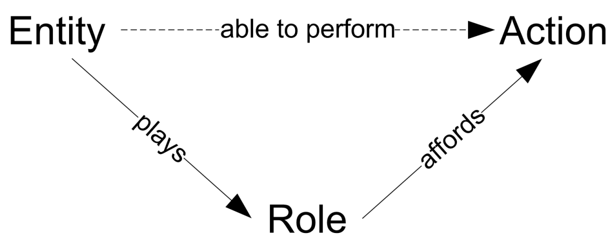


Figure 4.6 Entity, role, and action

In this conceptual model, *a role affords an action*. Thus, *in order for an entity to perform a certain action, it must play a specific role which affords the action*. This application of role as intermediary is necessary because, in the context of geospatial query solving, the participating characteristics of an entity in an action depends on the role it plays. In other words, an entity behaves differently in the same action when played under different roles.

Table 4.1 lists typical actions in geospatial queries. Each action has a corresponding type of subqueries and the roles that afford the action.

Table 4.1 Actions in geospatial queries

Action	Type of Subquery	Roles
Find the shortest route between two locations	Transformation	origin, destination, route
Determine the distance between two locations	Geometric Inquiry	origin, destination, length
Determine the length of a feature	Geometric Inquiry	linear entity, length
Determine the location of a place with a street address associated with it	Selection	street address, location
Determine the location of a place with a name associated with it	Selection	name, location
Determine if a location is located entirely within a region	Topological Test	location, region, boolean

Note that the actions shown in Figure 4.1 have only roles associated with them and not the actual entities. For example, the action “*Find the route between two locations*” requires three entities: the first one plays the role `origin`, the second one plays the role `destination`, and the third one plays the role `route`. In other words, given an origin and a destination, this particular action produces an entity that is a route. It does not matter what the entities are, as long as all the roles are fulfilled by eligible entities, the action can be carried out.

Because an entity can change its role dynamically, the relationship between entities and roles are not the same as a directly inherited (fixed hierarchical) relationship. For example, an entity of type Building can play either the role `origin` or the role `destination` (Figure 4.7), depending on the query (e.g., the query “*Find a route from Building to LandParcel*” is a different query from “*Find a route from LandParcel to Building*” because they produce different outcomes). Similarly, a single role can be played by many types of entities. The eligibility of an entity to play a role is explicitly defined in GPO.

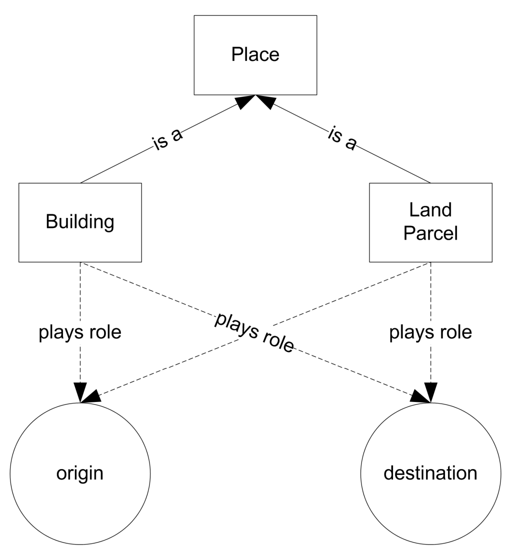


Figure 4.7 Roles and entities

4.3.3 Representation of Geospatial Queries

To semantically represent geospatial queries using entities, roles, and actions, we modified the Conceptual Graph (CG) knowledge representation [114, 115] to explicitly represent them. CG is a knowledge representation notation that is capable of representing a wide range of knowledge forms [69] and because of its visual nature, we choose it for the purpose of clarity of presentation.

Generally, a CG is a directed bipartite graph that has two types of nodes: (1) *concept nodes*, depicted as rectangles, are nodes representing concepts; and (2) *conceptual relation* nodes, depicted as circles, are nodes that represent relationships between concepts. We modified CG for GeoInterpret in order to precisely capture the notion of entities, roles, and actions as follows:

- Rectangular nodes represent entities (i.e., entity nodes).
- Circular nodes represent roles (i.e., role nodes).
- Triangular nodes represent actions (i.e., action nodes).
- An entity plays a role is indicated by an arc from the entity node to the role node.
- A role affords an action is indicated by an arc from the role node to the base of the (triangular) action node.
- An entity produced as a result of an action plays a specific role to that action (although the relationship is not of "affordance"). Their relationship is denoted by an arc from the entity node to the role node to the tip of the (triangular) action node.
- An entity node does not have any incoming arc but can have many outgoing arcs.
- A role node has exactly one incoming and one outgoing arc.
- An action node can have many incoming arcs to its base but has exactly one incoming arc to its tip (denoting the outcome of the action node).

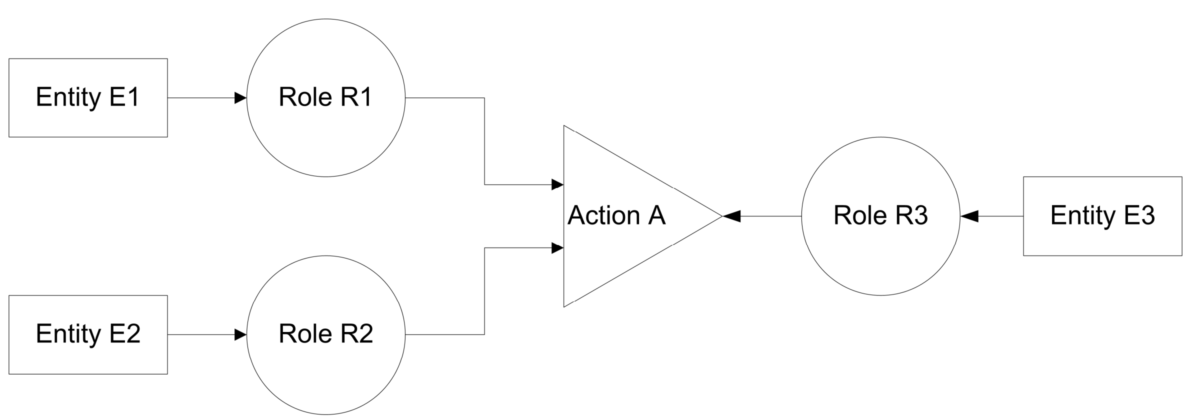


Figure 4.8 Entities, roles, and action in a modified CG representation

Figure 4.8 shows how a geospatial query can be represented as a modified CG. In this example, entity E1 plays role R1, entity E2 plays role R2. Both entities affords action A which produces entity E3 playing role R3, which also related to the action. The representation can be written as:

```
R3:E3<-[A]<-(R1:E1,R2:E2)
```

A query with one action node constitutes a subquery. Multiple subqueries make up a single, large query. Different subqueries require different sets of roles for the purpose of affordance. The roles and actions available for query formulation are defined in GDO as part of common geospatial ontology.

Selection subqueries requires two incoming roles: s*elector* and *feature* (Figure 4.9). The entity S1, which plays the role selector, determines which instances of F1 are selected as the output F2. This implies that S1 can be implemented as an array of Boolean values that corresponds to the array F1 of instances of a feature class, and that F2 is a subset of F1 where its instances are determined by the Boolean values stored in S1. A selection subquery can be written as:

```
feature:F2<-[Select]<-(feature:F1,selector:S1)
```

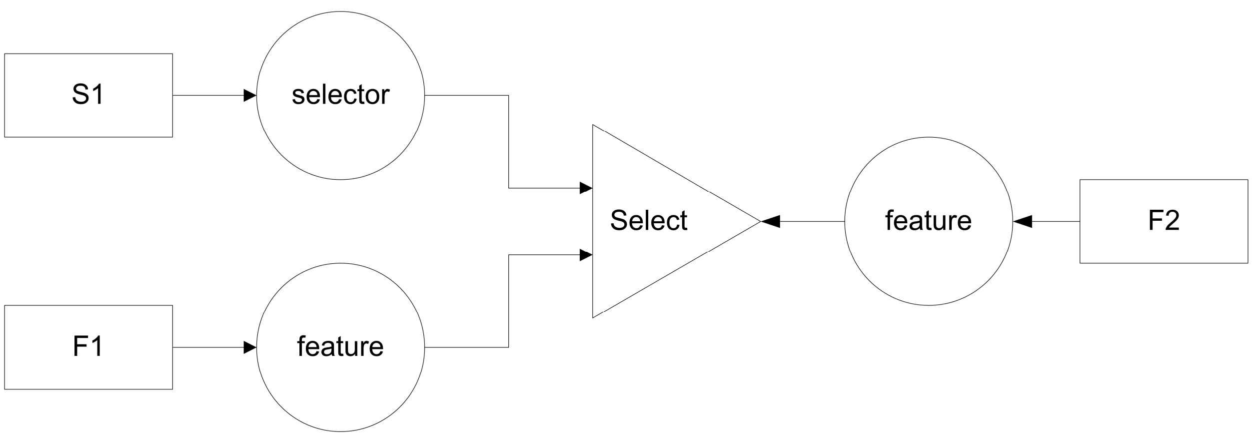


Figure 4.9 Selection subquery as modified CG

Other types of subqueries can be represented as modified CG in a straightforward manner. For example, a transformation subquery can be represented as:

```
R3:E3<-[A]<-(R1:E1,R2:E2)
```

In this form, E1 and E2 are entities of the base feature classes and E3 is the entity of the derived feature class. The roles R1, R2, and R3 associated with the query depend on the application-level semantic of the query and are chosen from those defined in GDO.

In a topological test subquery, the resulting entity plays the role `boolean`, which is functionally identical to the role `selector` used in selection subqueries (i.e. `selector` "is-a" `Boolean` as defined in ontology):

```
boolean:S1<-[TopoTest]<-(feature:F1,feature:F2)
```

Because each subquery CG is terminated at both the input and output ends with entity nodes, they can be *chained* to compose a larger query. For example, the query shown in Figure 4.10 performs a topological test on Fa and Fb and the resulting `boolean` values S1 are used as `selector` in the selection subquery. In this example, S1 plays different roles in different subqueries. It plays the role `boolean` in the first subquery and the role `selector` in the second subquery (even though they are functionally identical).

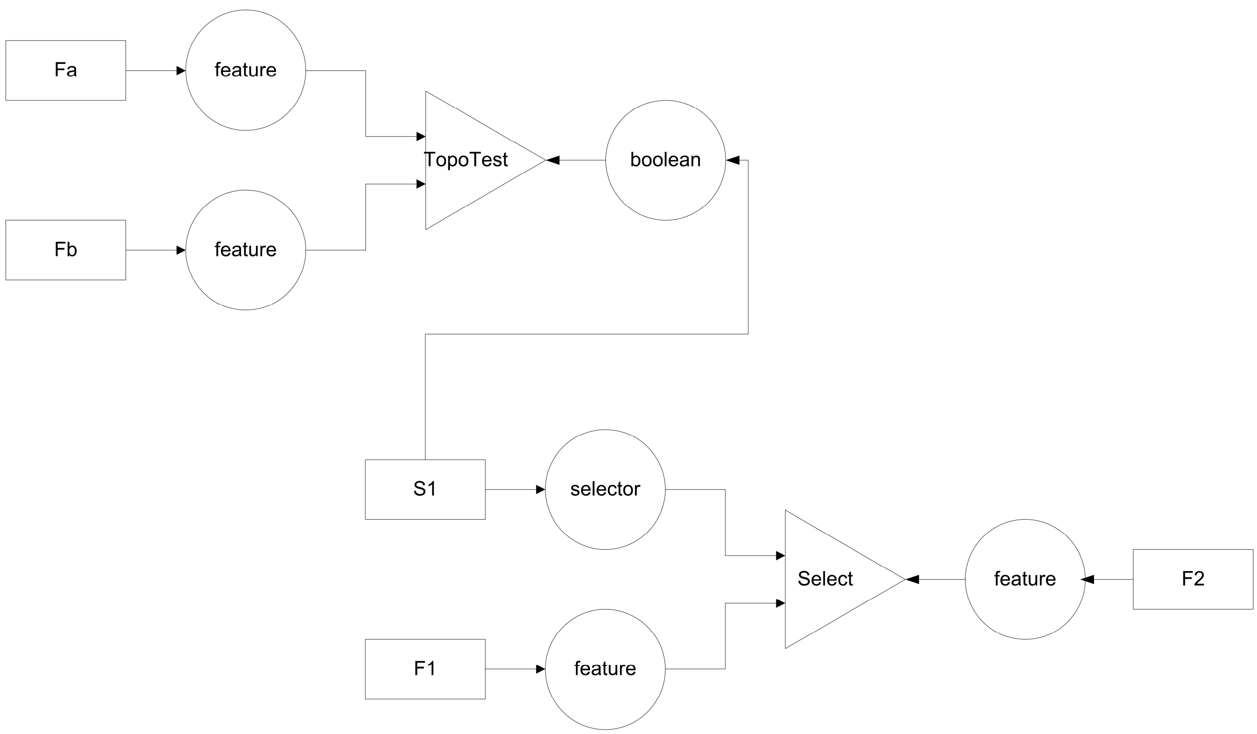


Figure 4.10 Chained subqueries as modified CG

In general, a complex geospatial query, when represented as chained subqueries, resembles a tree structure where the action nodes at the bottom (leaves) of the tree represent actions that must be completed first. The entity at the root (top) of the tree would represent the final outcome of the query.

4.4 GEOSPATIAL PROCESSING ONTOLOGY

GPO contains knowledge about geoprocessing that is independent from application-level knowledge. It comprises knowledge on geospatial data models and geoprocessing operations, and the constraints on how operations can be used on data models. In GeoInterpret, GPO is transparent to GDO and is accessed through Ontological Mediator.

4.4.1 Geospatial Data Types and Geoprocessing Operators

Commonly, basic vector-based geospatial data models include zero-dimensional objects (i.e., points), one-dimensional objects (i.e., lines or curves), and two-dimensional objects (i.e., polygons or surfaces). Other data models such as network and polyline are extensions of these basic types.

Demers [22] classifies GIS operators into selection, measurement, classification, statistical surfaces, spatial arrangement, comparison, and cartographic modeling. This classification is based on the functionality of the operators in the context of problem solving. Others [44, 74] classify them according to their computational characteristics, e.g., topological, geometrical, arithmetic and overlay. On the other hand, Verbyla [127] organizes them from a more practical perspective of GIS users as tabular, point, line, network, polygon, etc.; this categorization essentially corresponds to what GIS experts know about how GIS software represent objects. In addition, OGC through the *OGC Feature Geometry* standard [52] has defined geometrical and topological objects for the GIS community, including operators which can be applied to the objects.

To facilitate the interpretation of queries, operators in GeoInterpret are classified into two main types: *basic* and *compound*. A basic operator is defined as an operator that cannot be further broken down into smaller operators. For example, the shortest path computation is considered a basic operator in GeoInterpret, even though algorithmically it can be broken down into a sequence of smaller computations. It is considered to be a basic operator because it performs one specific *function* of finding the shortest path between two points within a line network.

For the purpose of clarity, operators in GeoInterpret are defined as *functions*. A basic operator in GeoInterpret takes one or more inputs and produces one output: `TZ=F(T1,…,TN)`, where `F` is the operator identifier, `TZ` is the object type of the outcome of the operator, and `T1,…,TN` are the object types of the operator's inputs. In geoprocessing, an operator is applied to a *layer* consisting of multiple instances of an object type.

A compound operator is made up of two or more basic operators and its purpose is to facilitate in the mapping of query elements. The rationale for having compound operators is that a user-level geospatial subquery often needs more than one basic operators to solve. Compound operators are a way to package common geoprocessing tasks into large units. As an example, consider a query asking whether a location X is within 5 miles of a location Y. This query consists of two subqueries. The first subquery asks for the Euclidean distance between two points, and the second subquery asks to compare the distance resulting from the first subquery against a fixed quantity of "*5 miles*". Users should be allowed to express the query as: "*Is X within 5 miles of Y?*" through an action defined in GDO without having to know how it is broken down into two simpler subqueries.

The knowledge of query decomposition can be defined at one of two places. The first is to define the decomposition knowledge at the *action* level in GDO by defining a *compound action* ("*Is X within Z miles of Y?*") as being made up of two *basic actions* ("*What is the distance from X to Y?*" plus "*Is the distance less than Z?*"). The other place where the decomposition knowledge can be defined is at the geoprocessing level where a compound operator: `BOOLEAN=WITHIN_DISTANCE(X,Y,Z)` is defined as two nested basic operators: `BOOLEAN=IS_LESS_THAN(DISTANCE(X,Y),Z)`. We choose the second approach in which the decomposition knowledge is defined at the geoprocessing level because of the following reasons. In the first approach, the correct decomposition knowledge depends on the correct understanding of actions at the application-domain level. In other words, the knowledge must be defined within the context of an application and thus may vary from one application to another (e.g., Does "within" mean the same thing across all application domains?). On the other hand, the second approach would produce stable decomposition knowledge because geoprocessing operators are application-independent and are precisely defined. Accordingly, a compound operator `TZ=FZ(T1,…,TN)` is mapped to one or more basic operators `F1,…,FN`.

4.4.2 Geospatial Data Sets in Geoprocessing

Geospatial problem solving, by definition, always requires at least one Earth-referenced data set. A data set consists of instances of a geospatial feature class and their attribute values. Following the *OGC Metadata* standard [56], a data set is described, through metadata, with identifying keywords which use conceptual terminology defined in an application-level ontology (e.g., GDO). The metadata also specifies the geographic extent of the data set using place names or a *geographic box* specifying the four corners of the geographic extent. In addition, the metadata also describes the CRS used in the data set.

When data sets are imported into GIS, they must be converted into a common CRS. In addition, if the data sets are of different data formats, they must also be converted into the native format used in the GIS. Because these conversions are syntactic in nature and are primarily issues related to interoperability, we omit them in the dissertation.

4.5 ONTOLOGICAL MEDIATOR

The knowledge which specifies the mappings from *entities* to *geospatial data models* and the knowledge which specifies the mappings from *actions* to *geoprocessing operations* are defined in Ontological Mediator. These two types of mappings are respectively defined in *MetaEntities* and *MetaActions*, which comprise Ontological Mediator.

4.5.1 MetaEntities

Mapping in MetaEntities involves entities, roles, and geospatial data models. *An entity, when played under a role, is mapped to a geospatial data type*. This also implies that when an entity plays a different role, it may be mapped to a different data model. For example (Figure 4.11), the entity `Lake`, when plays the role `destination` in a query, is mapped to the `POINT` geospatial data type. However, the same entity `Lake`, when plays a different role of `territory`, is mapped to `POLYGON`. This type of mapping is similar to the concept of semantic projection [72] where semantically-rich concepts are mapped to "simpler" conceptual space. This grounding of entities to geospatial data models through roles restricts the possible interpretation and mapping of queries to geoprocessing operators.

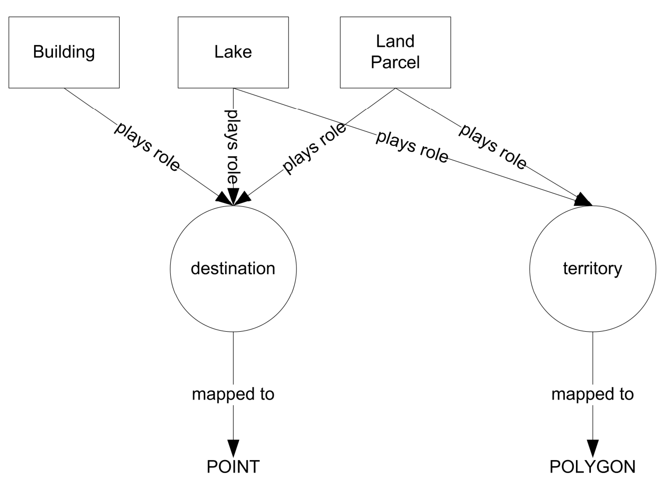


Figure 4.11 MetaEntities mappings

4.5.2 MetaActions

Mapping in MetaActions involves actions, roles, and geoprocessing operators. *An action, together with its related roles, is mapped to a geoprocessing operator.* Table 4.2 contains examples of mappings where each mapping is in the form:

> `rz:[A]<-(r1,…,rn) is mapped to TZ=F(T1,…,Tn)`; *where* `A` *is an action,* `r1,…,rn` *are input roles to* `A`, `rz` *is output role of* `A`, `F` *is the geoprocessing operator,* `T1,…,Tn` *are the input parameter to* `F`, *and* `TZ` *is the output parameter of* `F`.

An action alone without roles is insufficient for mapping because two identical action but with different associated roles do not represent the same *intent*; and hence, do not map to the same operator. For example, the action `Locate` (the first two entries in Table 4.2) is to identify the location of a place (i.e., an entity playing the role `location`). A location can be identified by either an `address` or a `name` (i.e., the two roles associated with action `Locate`). An example of an entity that can play the role `address` is "*135 N. Bellefield Ave.*", which is mapped to the `STRING` data type in a `GEOCODE` operation. On the other hand, the same `Locate` action but with a different associated role is mapped to a different operator (`Locate` by `name` is mapped to `GAZETTEERLOOKUP`).

Table 4.2 MetaActions mappings

Action-Role	**Operation**
`location:[Locate]<-(address)`	`POINT=GEOCODE(STRING)`
`location:[Locate]<-(name)`	`POINT=GAZETTEERLOOKUP(STRING)`
`pathway:[FindShortestRoute]<-(origin,destination)`	`POLYLINE=DIJKSTRA(POINT,POINT)`
`length:[MeasureDistance]<-(pathway)`	`NUMBER=LENGTH(POLYLINE)`
`length:[MeasureDistance]<-(origin,destination)`	`NUMBER=LENGTH(POINT,POINT)`
`border:[FindBoundary]<-(territory)`	`POLYGON=BOUNDARY(POLYGON)` `POLYGON=BOUNDARY(NETWORK)`
`boolean:[IsWithinDistance]<-(location,location,length)`	`BOOLEAN=WITHIN_DISTANCE(POINT,POINT,NUMBER)`

Additionally, an action can also be mapped to more than one operator. For example, the action `border:[FindBoundary]<-(territory)` is mapped to two different `BOUNDARY` operators – one returns the boundary of a `POLYGON` object while the other returns the boundary of a `NETWORK` object. This mapping is possible because `territory` is a role that can be played by many types of entities, some of which are represented simply as `POLYGON` (e.g., LandParcel), while others are represented as `NETWORK` (e.g., a road network spanning an area). To resolve this ambiguity, MetaEntities is used to determine exactly which geospatial data model is used by a given entity under a specific role.

Another type of entries in MetaActions is *role conversions*. It is in the form:

> *`r2<r1 is mapped to TZ=F(T1,…,TN)`; where `r1` is the original role, `r2` is the new role, `F` is a geoprocessing operator, `T1,…,TN` are the input parameter to `F`, and `TZ` is the output parameter of `F`.*

Recall that a geospatial query is modeled as chained subqueries where the connection between any two subqueries is an entity. In addition, the same query often uses an entity in multiple subqueries. As a consequence, the same entity often must play different roles under different subqueries. In some cases where the conversion from one role to another is not explicitly stated as an action, or that the data set representing the entity is not of the correct geospatial data model, the change in role (i.e., role conversion) may be accomplished by using a geoprocessing operation, which is also defined in MetaActions. Table 4.3 shows examples of role conversion mappings. For example, consider the query "*What is the area of the LandParcel and how far is it from the school*". In this query, there is an entity `LandParcel` that initially plays the role `territory` and uses `POLYGON` data models in the first subquery ("*What is the area*?"). However, in the second subquery ("*How far is it*?") the same `LandParcel` plays a different role `location`, which is represented as `POINT` for geoprocessing. The reason for this is because the intention of the second subquery is to find the distance from one `location` to another `location` and does not deal with the fact that the `LandParcel` can also be a `territory`. In this case, there is no explicit action within the subquery chain to change the role from `territory` to `location` that would result in data conversion from `POLYGON` to `POINT`. In this case, the role conversion can be done automatically using the `CENTROID` operator defined in MetaActions based on available data set. For example, if there exists a `POLYGON` data set for `LandParcel`, then MetaActions allows `LandParcel` to be converted to `POINT` (through the CENTROID operator) only if the conversion is done for the purpose of changing from role `territory` to role `location`.

Table 4.3 Role conversion mappings in MetaActions

Role Conversion Signature (User-Level)	Operations (Geoprocessing-Level)
`(location)<-(territory)`	`POINT=CENTROID(POLYGON)` `POINT=CENTROID(NETWORK)`
`(location)<-(boundary)`	`POINT=CENTROID(POLYGON)` `POINT=CENTROID(NETWORK)`

4.6 SUMMARY

This chapter provides the underlying methodology of GeoInterpret by discussing its components and the schemes for relating elements in different ontologies. Entities, actions, roles, affordances, and how they are applied in GDO and GPO were discussed. In addition, the knowledge representation of geospatial queries including its graphical representation is discussed in the chapter.

5.0 INTERPRETATION OF GEOSPATIAL QUERIES

This chapter discusses the algorithms that utilize the GeoInterpret knowledge base to interpret geospatial queries, map interpreted queries to geoprocessing operations, and create geoprocessing workflows that solve geospatial queries. The algorithms are: Q-GET, Q-ANALYZE, and W-COMPOSE.

5.1 Q-GET: FORMULATING GEOSPATIAL QUERIES

The purpose of Q-GET is to formulate geospatial queries into modified CG representation. Q-GET determines subqueries and obtains other components of a query, namely geographic extent and output types. The flowchart for the Q-GET algorithm is shown in Figure 5.1.

The initial step in formulation of a geospatial query is to specify the geographic extent (Step A). Next, each of the subqueries is constructed by first specifying its action (Step B). Then, possible sets of roles that can be associated with the action are retrieved from MetaActions (Step C) and one is chosen (Step D). After this step, the action intent is unambiguously identified with its associated roles. Once the roles are known, the user can then associate entities defined in Domain Ontology to the roles (Steps E and F). However, since the resulting entity of an action can be used as input to another subquery (chaining), the user may choose to associate a role with an entity already specified in one of the previous subqueries. Once all subqueries are created and chained, the resultant terminal entities will be output as outcome of the query (Steps G and H).

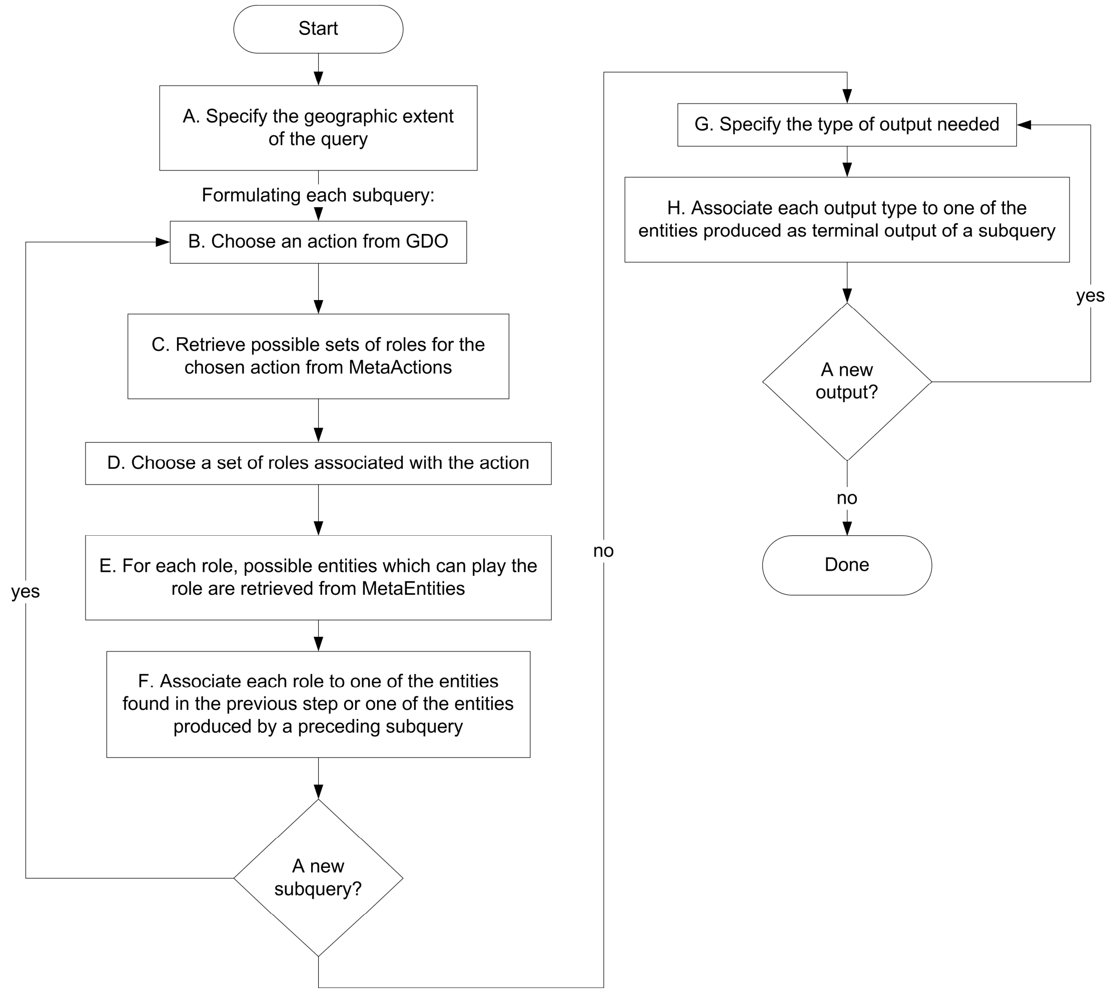


Figure 5.1 Flowchart of the Q-GET algorithm

Consider the following example:

Find the shortest route from 135 N. Bellefield Ave. to Frick Park.

This query asks for the shortest route from a location specified by a street address to another location specified by a place name. To formulate the query, first the geographic extent is specified, which is "*Pittsburgh, PA*". Next, the action `FindShortestRoute`, is chosen which only has one possible set of roles associated to it (Figure 5.2).

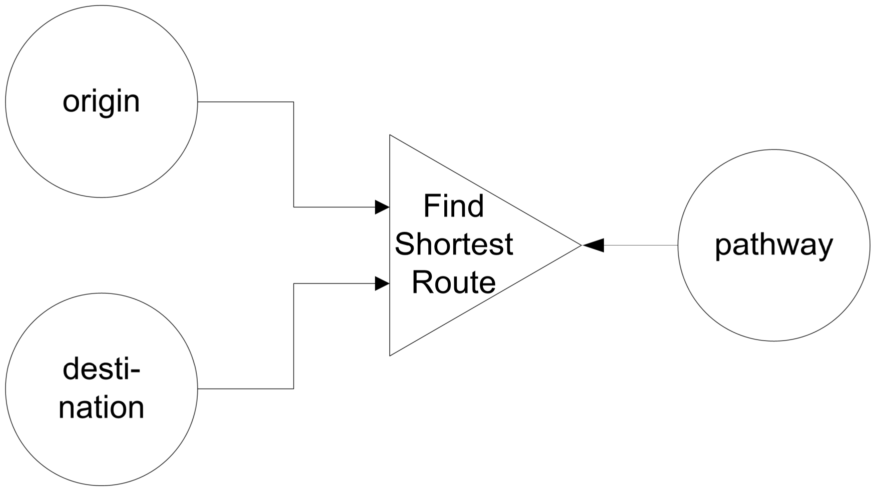


Figure 5.2 Modified CG representation for the FindShortestRoute action

The roles `origin` and `destination` need to be associated to entities presented in the query, but that is not possible because neither "`135 N. Bellefield Ave.`" nor "`Frick Park`" is an entity that can play either of the roles. The only entity which can make an association is `Route` which can play the role `pathway` and is the outcome of the action. It is necessary then to identify the meaning of the two text string by using actions which make them meaningful. In this case, the action `Locate` is used on both strings. However, since the intended meanings of the two strings are different, the appropriate role for each string must be chosen. In this case, "`135 N. Bellefield Ave.`" plays the role `address` and is a `Building`, and "`Frick Park`" plays the role `name` and is a `Park`, both `Building` and `Park` play the role `location`, which can be meaningfully used as `origin` and `destination`. The complete modified CG for the query is shown in Figure 5.3.

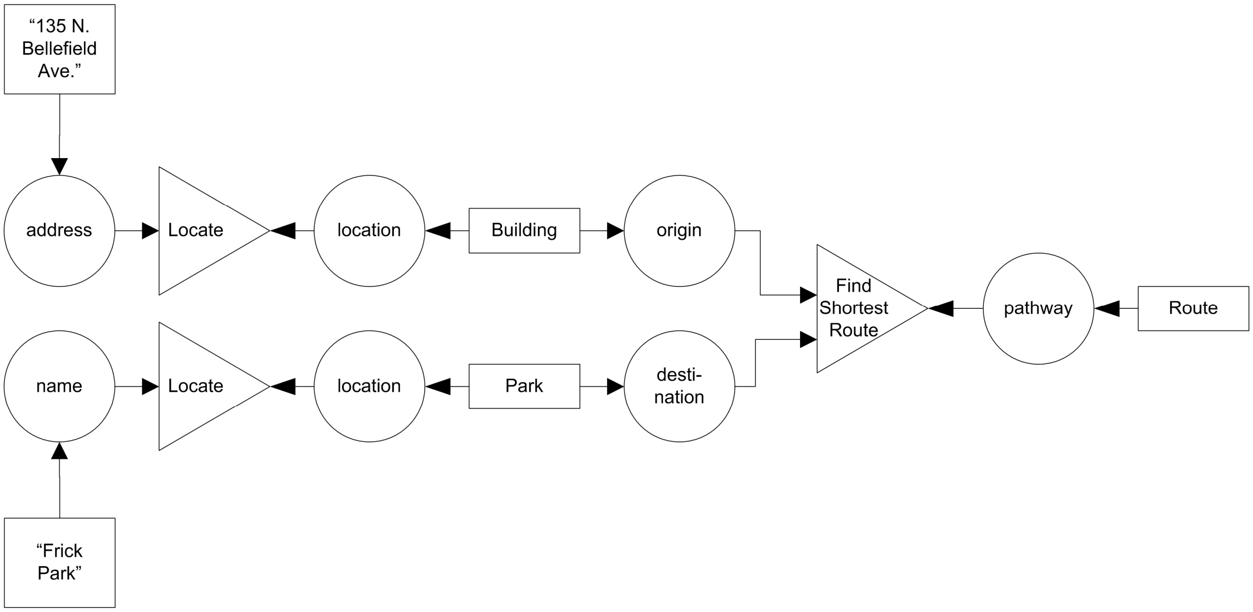


Figure 5.3 A query to find the shortest route from building to park

In another example, consider the query:

Select land parcels which are priced less than $100,000.

Using Q-GET, the modified CG for the query is shown in Figure 5.4.

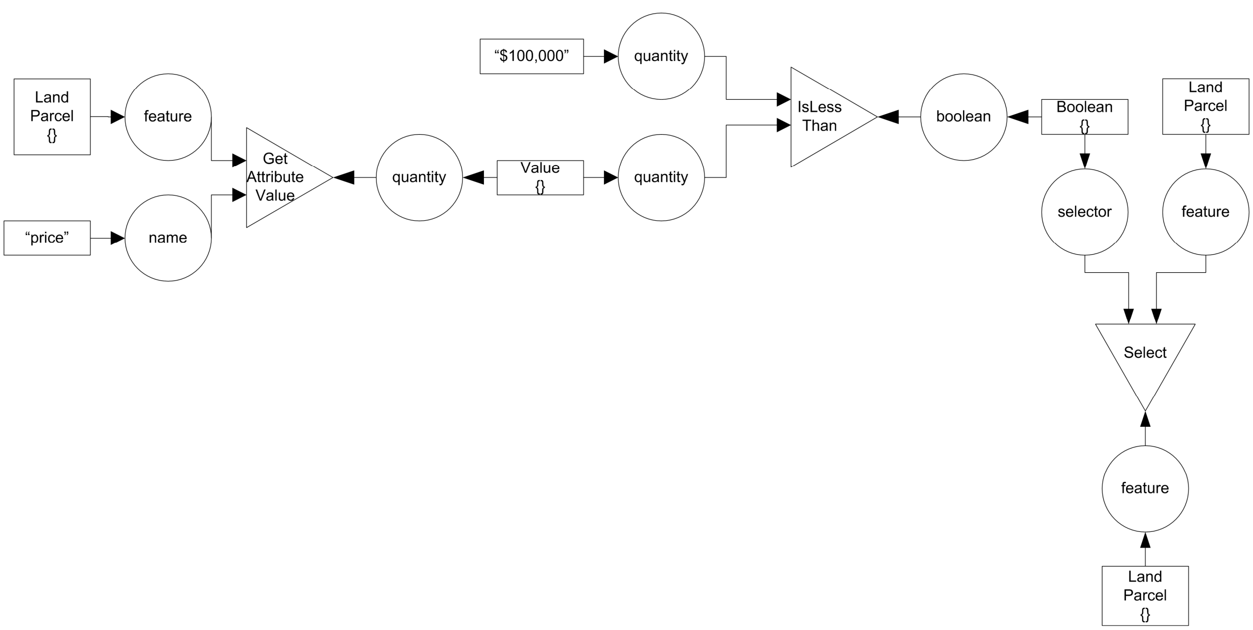


Figure 5.4 A query to select only land parcels that cost less than $100,000

The notation `{}` signifies that action is performed on a layer containing instances of an entity class; thus, `Value{}` and `Boolean{}` are two sets of instances of `Value` and `Boolean` entities, respectively. `Boolean{}` is used as `selector` to the `Select` action, which means that only a subset of instances of `LandParcel{}` are selected based on the corresponding value in `Boolean{}`.

5.2 Q-ANALYZE: MAPPING QUERIES TO OPERATIONS

Q-ANALYZE uses the result of Q-GET (a modified CG) to interpret and map geospatial queries to geoprocessing operations. The flowchart of the algorithm is shown in Figure 5.5.

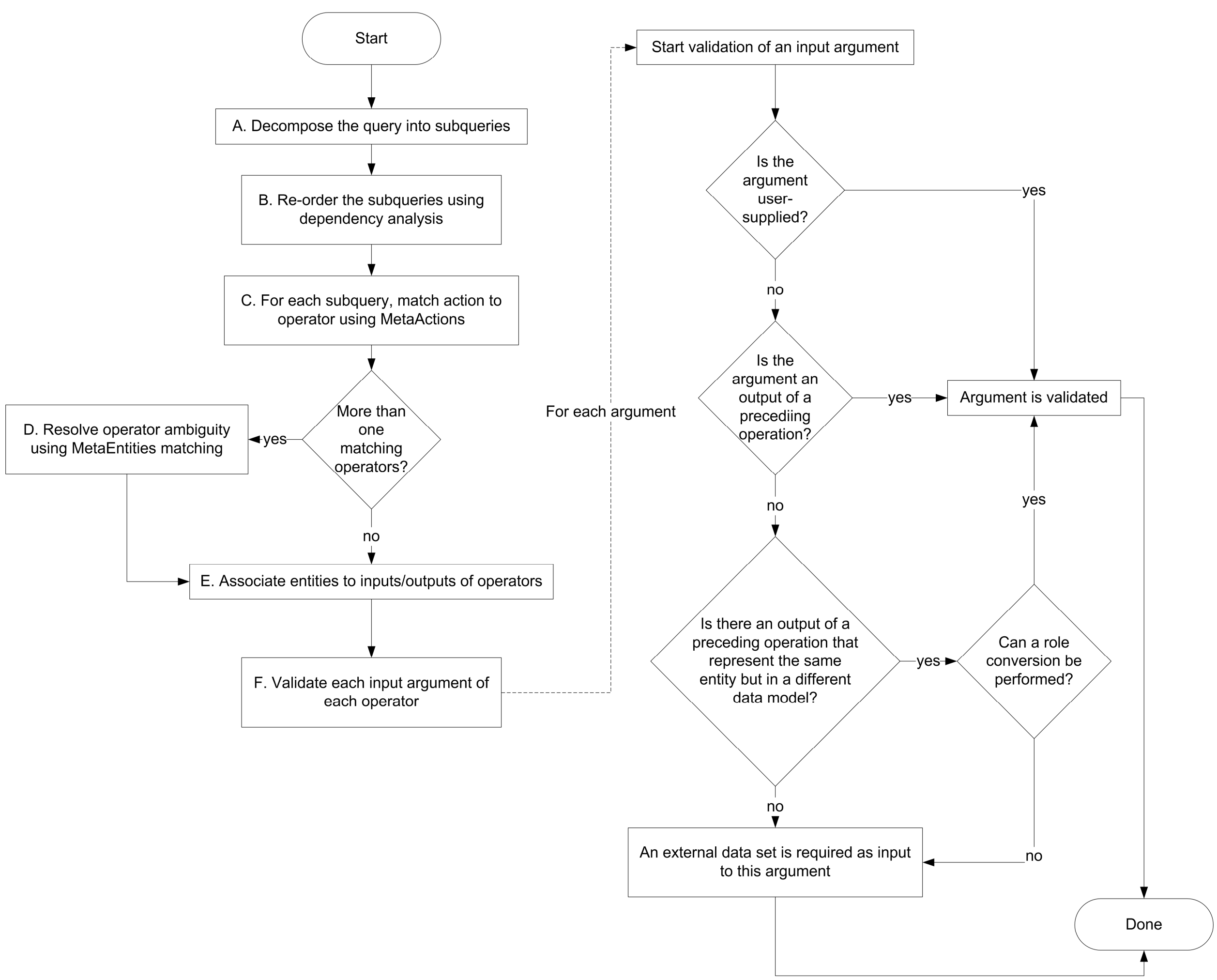


Figure 5.5 Flowchart of the Q-ANALYZE algorithm

Step A of the algorithm is to decompose query into subqueries. This is done by isolating each action nodes in a query. Each subquery consists of an action node, role nodes adjacent to action node, and entity nodes associated with role nodes. Next, the resultant subqueries are re-ordered (Step B) according to their dependency, which can be determined by the direction of (triangular) action nodes indicating the output produced by actions. Once a list of ordered subqueries is obtained, each one is matched against MetaActions (Step C) to find the candidate operators `F1,…,FN` for its solution. This matching is performed using action `A` and roles `r1,…,rn` of the subquery:

```
rz:[A]<-(r1,…,rn) is matched to TZ=F(T1,…,Tn)
```

In the cases where two or more operators are matched, MetaEntities is consulted (Step D) to determine, for each subquery, the appropriate data types `G1,…,GN` representing entities `E1,…,EN` under their respective roles `R1,…,RN`. The operator with the matching data type parameters `G1,…,GN` is selected for the subquery.

Once operators for the subqueries are determined, entities of the subqueries are associated with the input and output parameters of the respective operators (Step E). These bounded operators are now referred to as *operations*. In the last step, each of the input arguments from all operations are validated (Step F) as follows: An input argument to an operation is validated if it does not depend on an external data set. There are three conditions possible for an argument to be validated:

- The argument is a user-supplied input (e.g., “135 N. Bellefield Ave.”).
- The argument is an output of any one of the preceding operations.
- The argument can be obtained through role conversion based on available objects produced by any one of the preceding operations.

If none of these three conditions can be met, then the argument must be satisfied by an external data set (i.e., an external data set of the correct geospatial data model must be used to carry out the operation).

Consider the following example:

Display the shortest route from every land parcel costing less than $100,000 to the building located at 135 N. Bellefield Ave.

The modified CG for the query is shown in Figure 5.6.

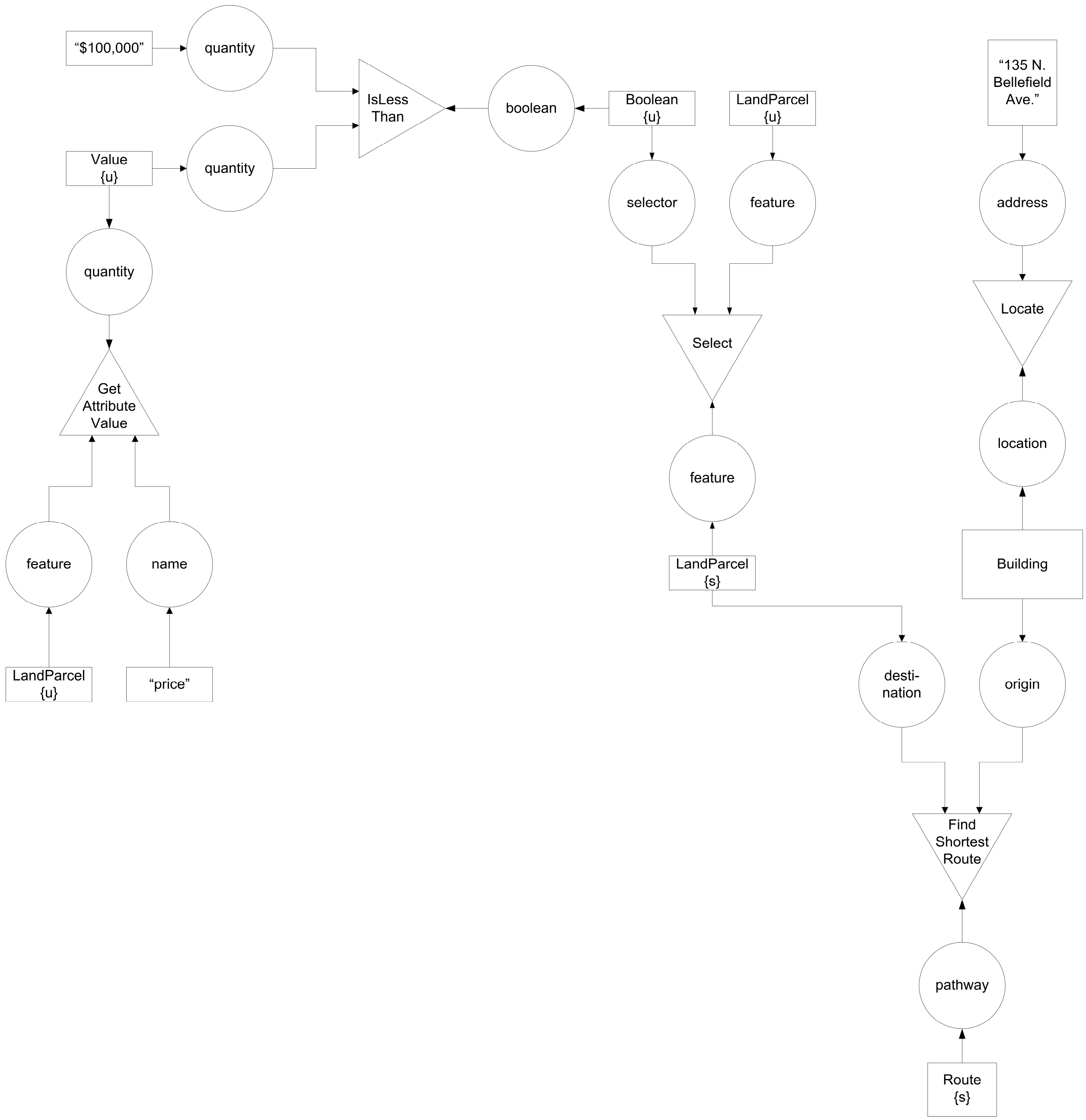


Figure 5.6 A scenario in modified CG form

Following the initial step of the algorithm, we first re-order the subqueries using their dependency:

```
1) quantity:Value{u}<-[GetAttributeValue]<-(object:LandParcel{u},name:"price")
2) boolean:Boolean{u}<-[IsLessThan]<-(quantity:"$100,000",quantity:Value{u})
3) feature:LandParcel{s}<-[Select]<-(selector:Boolean{u},feature:LandParcel{u})
4) location:Building<-[Locate]<-(address:"135 N Bellefield Ave.")
5) pathway:Route{s}<-[FindShortestRoute]<-(origin:Building,destination:LandParcel{s})
```

Next, each subquery is matched to possible operators through MetaActions:

```
1) CANDIDATE 1: NUMBER[]=GETATTRIBUTEVALUE(OBJECT[],STRING)
   CANDIDATE 2: STRING[]=GETATTRIBUTEVALUE(OBJECT[],STRING)
2) BOOLEAN=ISLESSTHAN(NUMBER,NUMBER)
3) OBJECT[]=FILTER(BOOLEAN[],OBJECT[])
4) POINT=GEOCODE(STRING)
5) POLYLINE[]=ROUTING(POINT[],POINT[])
```

The ambiguity of the first subquery is resolved through MetaEntities (i.e., role `quantity` is mapped to `NUMBER`) and the new list containing unique operator for each subquery is:

```
1) NUMBER[]=GETATTRIBUTEVALUE(OBJECT[],STRING)
2) BOOLEAN=ISLESSTHAN(NUMBER,NUMBER)
3) OBJECT[]=FILTER(BOOLEAN[],OBJECT[])
4) POINT=GEOCODE(STRING)
5) POLYLINE[]=ROUTING(POINT[],POINT[])
```

In the next step, input and output parameters of each operator are associated to the entities in the subqueries:

```
1) NUMBER[]:Value{u}=GETATTRIBUTEVALUE(OBJECT[]:LandParcel{u},STRING:"price")
2) BOOLEAN:Boolean{u}=ISLESSTHAN(NUMBER:"$100,000",NUMBER:Value{u})
3) OBJECT[]:LandParcel{s}=FILTER(BOOLEAN[]:Boolean{u},OBJECT[]LandParcel{u})
4) POINT:Building=GEOCODE(STRING:"135 N Bellefield Ave.")
5) POLYLINE[]:Route{s}=ROUTING(POINT[]:Building,POINT[]:LandParcel{s})
```

At this point, Q-ANALYZE validates the input arguments of the operations. An argument which is user-supplied (in quotes) is valid, as is an argument which is produced from one of the preceding operations. That leaves the following unresolved input arguments:

```
OBJECT[]:LandParcel{u}
POINT[]:LandParcel{u}
```

Since `OBJECT` subsumes `POINT` in the ontological hierarchy (i.e., the operator requiring `OBJECT[]:LandParcel{u}` can carry out its task using `LandParcel` data set in any geospatial data model), the only unresolved argument required to solve this query is `POINT[]:LandParcel{u}`. Because `LandParcel{u}` of the type `POINT` is not available as output of one of the operations and that a role conversion is not possible (because no `LandParcel{u}` of another data type is available), `POINT[]:LandParcel{u}` must be obtained externally as a data set containing Earth-reference `POINT` data of all land parcels within the required geographic extent.

5.3 W-COMPOSE: COMPOSING GEOPROCESSING WORKFLOWS

Once the subqueries of a given query have been mapped to geoprocessing operations, W-COMPOSE composes a geoprocessing workflow for solving the query. The flowchart of W-COMPOSE is shown in Figure 5.7.

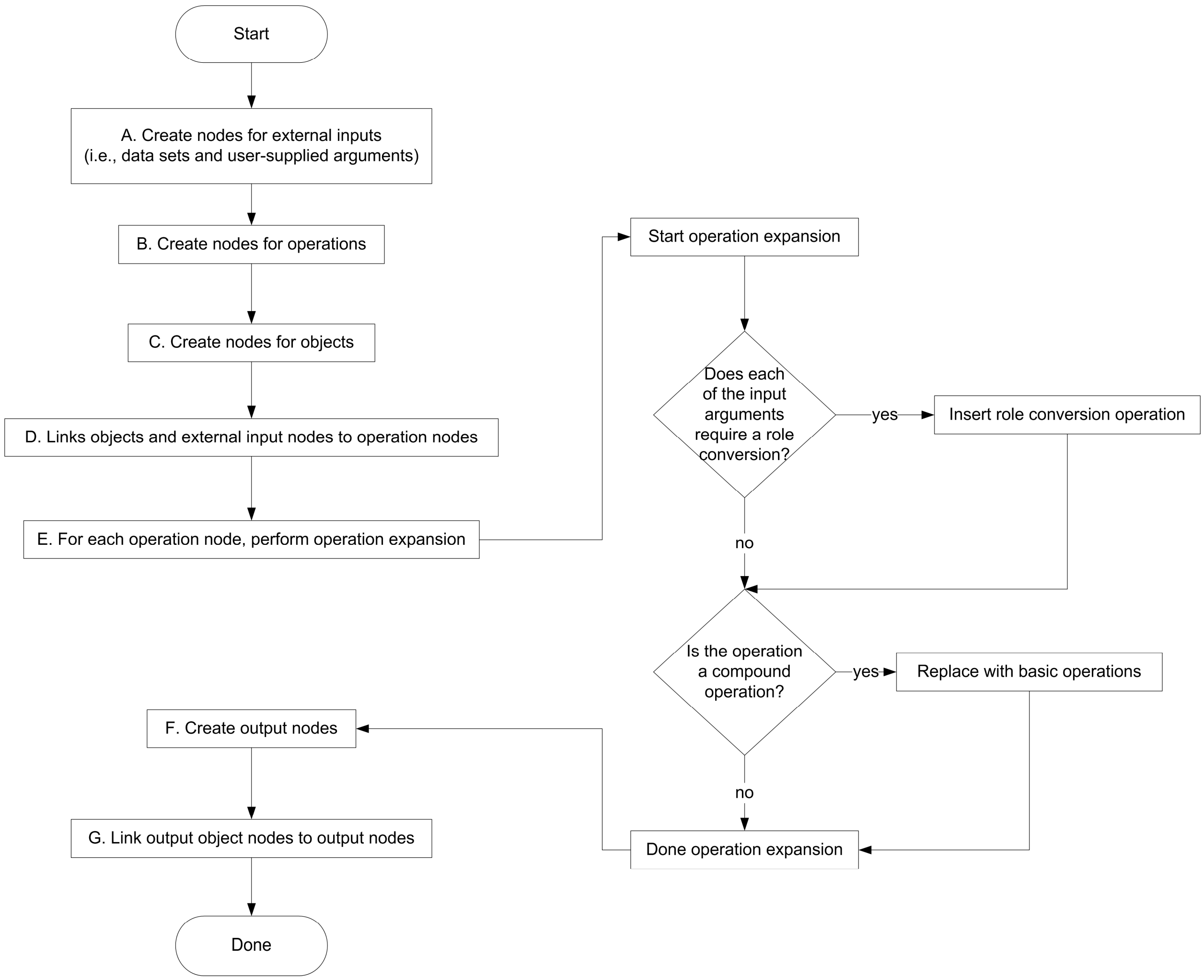


Figure 5.7 Flowchart of the W-COMPOSE algorithm

A geoprocessing workflow can be modeled as a *Directed Acyclic Graph* (DAG) with the following conventions for graph nodes: shaded pentagons for external inputs including data sets and user-supplied values; solid circles for operations; squares for objects; and stars for outputs. An operation takes one or more data sets and/or objects as inputs, and produces a single output that is an object (this can be used as input to other operations). An object is an output of a query if it has an outgoing arc pointing to the output (star) node. A complete workflow for a query starts with one or more external input nodes and terminates with one or more output nodes.

The first step in creating a geoprocessing workflow is to create all external input nodes (Step A), then operations nodes (Step B), and finally object nodes (Step C). Once the nodes are created, object and external input nodes are linked to the appropriate operation nodes (Step D). The information necessary to perform linking is available as a result from Q-ANALYZE. The next step in the algorithm is to perform *operation expansion* (Step E), which is a procedure that converts role conversions and compound operations to basic operations. The algorithm determines, for each operation node, if any of the input arguments is a role conversion. If it is, then it replaces the argument with the matching role conversion operation defined in MetaActions. Next, the algorithm determines if the operation is a compound operation. If it is, the algorithm replaces the operation with the corresponding basic operations defined in GPO. The resultant workflow contains only basic operations.

After the operation expansion step is completed, output (star) nodes are created (Step F). Each of the output nodes is then linked to the entities specified in Q-GET as outputs (Step G). The composed geoprocessing workflow for the query scenario is shown in Figure 5.8.

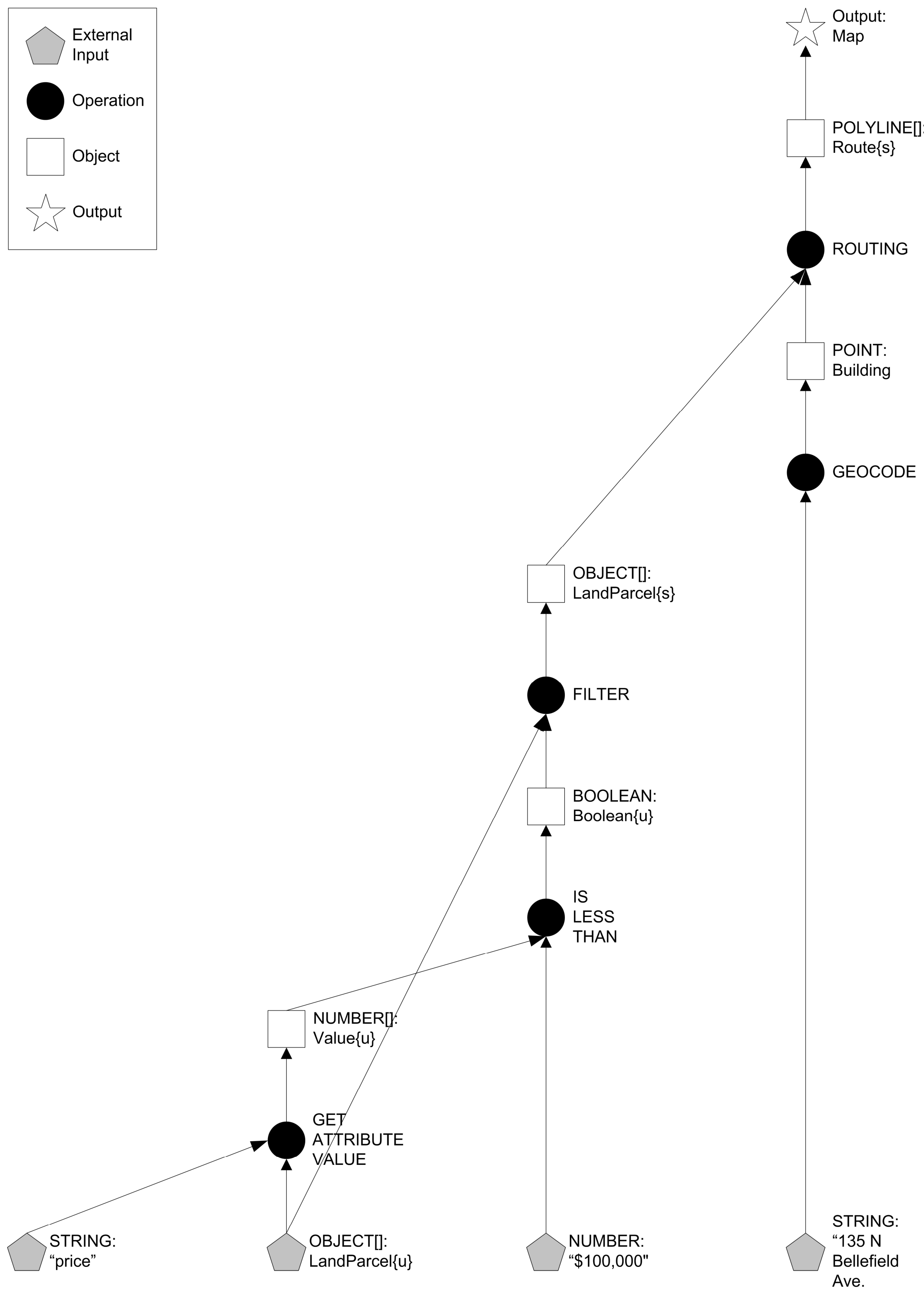


Figure 5.8 Geoprocessing workflow for the scenario

5.4 SUMMARY

This chapter discusses a set of algorithms that uses the ontological knowledge base to perform geospatial query formulation, map queries to geoprocessing operations, and compose geoprocessing workflows. These algorithms are Q-GET, Q-ANALYZE, and W-COMPOSE. Q-GET utilizes the application-level ontology, GPO, in formulating queries. Q-ANALYZE maps subqueries' elements to geoprocessing elements defined in GPO. W-COMPOSE constructs geoprocessing workflows based on the result of Q-ANALYZE.

6.0 PROOF OF CONCEPT

6.1 INTRODUCTION

To demonstrate the working of GeoInterpret, a proof of concept was developed and verified using two case studies. The first case involved generic queries dealing with general geospatial problems and the second case involved queries specific to the application domain of *Architecture, Engineering, Construction and Facility Management* (AEC/FM), which is a major branch in Civil Engineering discipline. In the second case, the objective is also to demonstrate the integration of application domain-specific concepts and terminology into the ontological components of GeoInterpret. The integration was performed based on an available community standard that defines an ontology of the domain.

Since the main contribution of this research is a general methodology that is based on general conceptualization of geospatial queries and does not depend on specific application domains, the proof of concept was intended to only verify the mapping techniques and algorithms of the methodology. Validation for domain-specific queries requires comprehensive community-based ontologies which is beyond the scope of this research.

6.2 PROOF OF CONCEPT

6.2.1 Software Tools

The proof of concept was implemented using the following software tools:

- *Protégé Ontology Editor* Version 2.1 [101], which was used to construct the knowledge base of GeoInterpret, including GDO, GPO, and Ontological Mediator.
- *Algernon* Version 3, a rule-based inference system [4], which was used as the programming interface for accessing the knowledge base as well as for implementing part of the Q-ANALYZE algorithm.
- *Java* programming language Version 1.4.2 [57], which was used to implement the user interface and the Q-GET, Q-ANALYZE, and W-COMPOSE algorithms.

The Protégé Ontology Editor was used to construct GDO, GPO, and Ontological Mediator. GDO was modeled in Protégé as the following classes: *Entity*, *Role*, and *Action*. The class Action defined actions and their descriptions, the class Role defined the taxonomy of roles, and the class Entity defined the taxonomy of entities. GPO was modeled in Protégé as two classes: *Representation* and *Operator*. The class Representation defined vector objects (e.g., point, line, polygon) and non-vector objects (e.g., number, string) involved in geoprocessing. The class Operator defined geoprocessing operators and their input/output parameters (which belong to the class Representation). Class Operator defines both compound and basic operator. Ontological Mediator was modeled separately in Protégé where MetaEntities and MetaActions are modeled as subclasses (Figure 6.1).

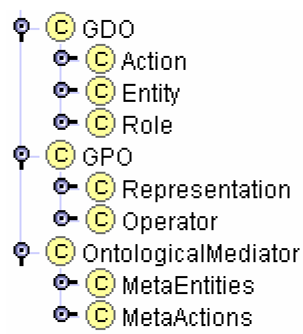
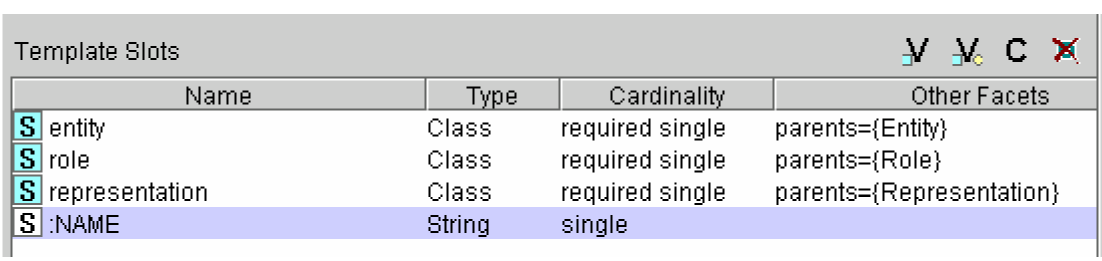


Figure 6.1 Knowledge base in Protege

Entries in MetaEntities and MetaActions are defined in Protégé using template slots. For example, entries in MetaEntities, where entities are mapped to geospatial data models under specific roles, are modeled using three template slots: entity, role, and representation; where the slots contain the corresponding classes defined in GDO and GPO (Figure 6.2).



Template Slots

Name	Type	Cardinality	Other Facets
S entity	Class	required single	parents={Entity}
S role	Class	required single	parents={Role}
S representation	Class	required single	parents={Representation}
S :NAME	String	single	

Figure 6.2 MetaEntities slots in Protege

The Q-GET, Q-ANALYZE, and W-COMPOSE algorithms were implemented using Java and the Algernon inference system. Algernon, which provides efficient and concise mechanism for knowledge base traversal, was used to implement part of Q-ANALYZE.

6.2.2 Using Proof of Concept

The proof of concept supports a simple graphical-based interface for query submission. Applying the Q-GET algorithm, the user is asked to formulate query by specifying an appropriate set of actions that makes up its subqueries (Figure 6.3). For each action, the system provides a list of possible sets of roles that can be associated with the action. After an action has been decided, its associated roles must be bound to entities (Figure 6.4).

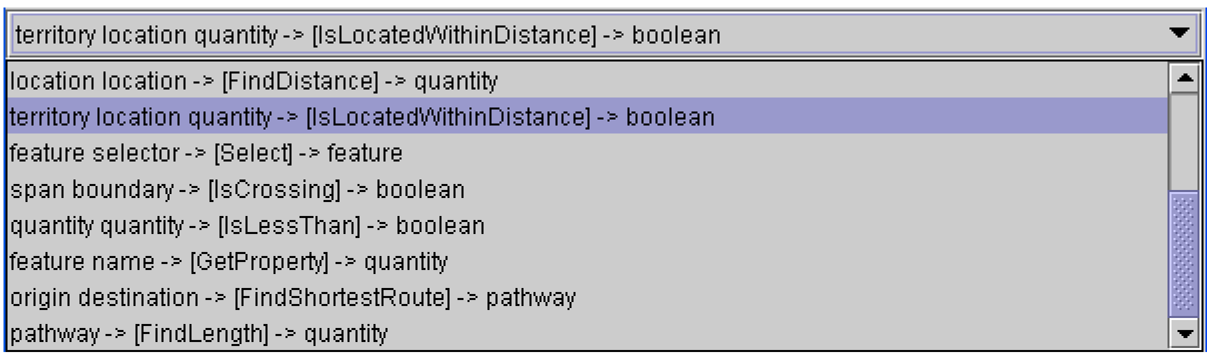


Figure 6.3 List of actions in the proof of concept

Figure 6.4 Binding entities to roles in the proof of concept

The construction of a subquery is completed when every role has a corresponding real-world entity associated with it. A newly constructed subquery is added to the list of subqueries displayed in the main system window (Figure 6.5). When all subqueries have been constructed and chained, the system proceeds with query interpretation and mapping (Q-ANALYZE), and workflow composition (W-COMPOSE).

The resultant geoprocessing workflow is then displayed as a list of operations in a separate dialog window (Figure 6.6). The required contents of external data sets are also listed separately.

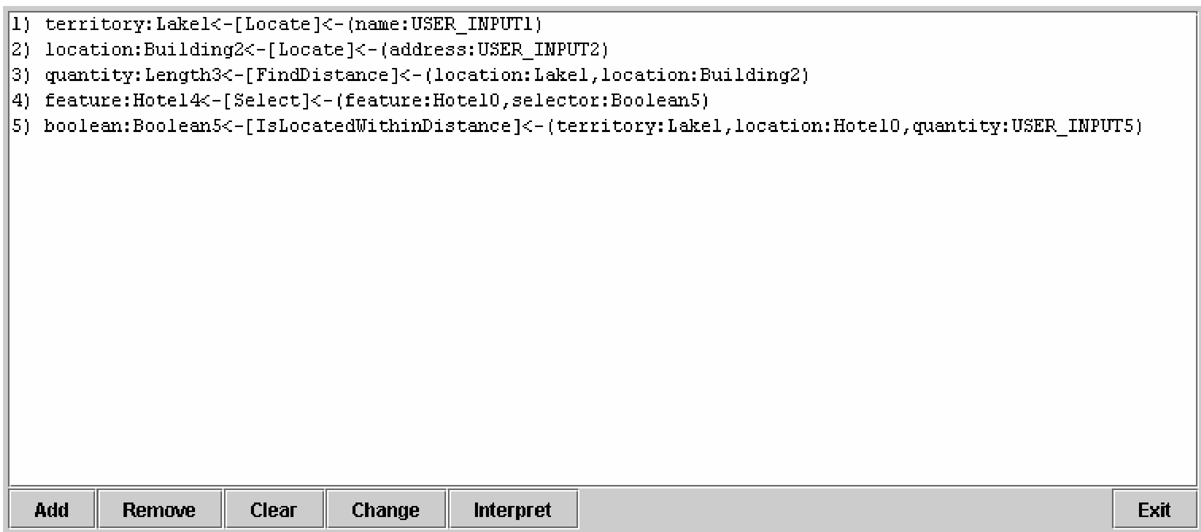


Figure 6.5 List of subqueries in the proof of concept

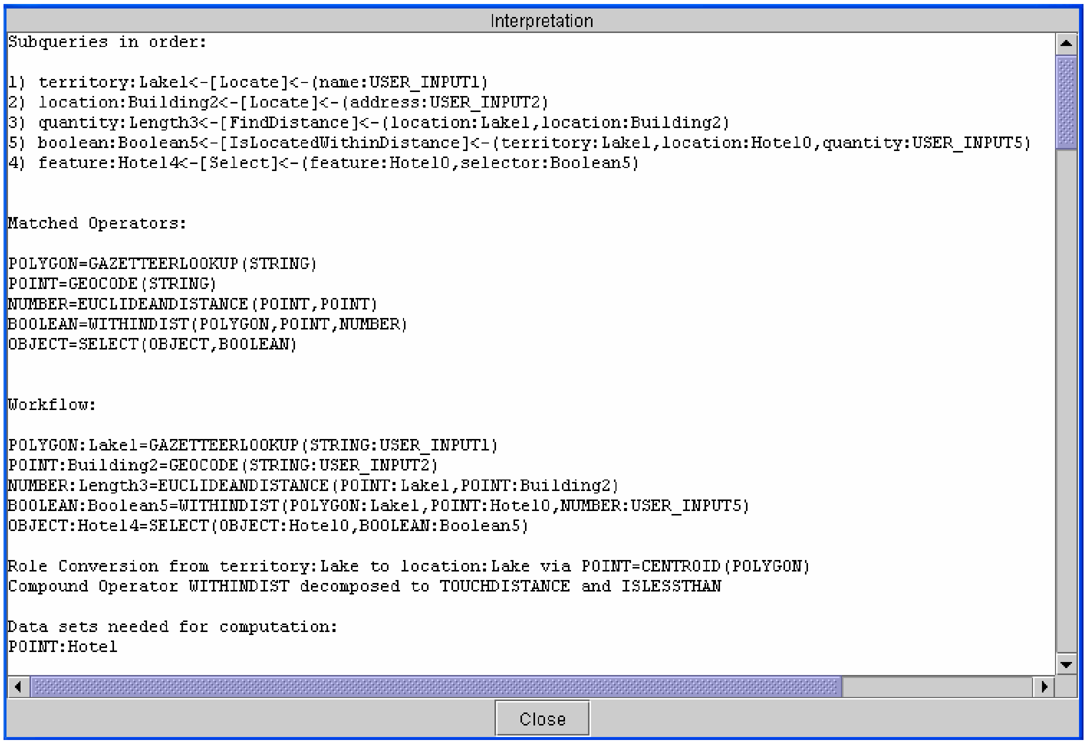


Figure 6.6 Resulting workflow generated by the proof of concept

6.3 CASE STUDY I: GENERIC GEOSPATIAL QUERIES

6.3.1 Queries

A set of general queries were selected to test GeoInterpret. Since the proof of concept was to demonstrate the methodology, the queries were selected to test logical paths presented in the algorithms, including:

- *Role conversion*, where a given role must be converted to another role using a geoprocessing operation.
- *Compound operator*, where a compound operator must be expanded into two or more basic operators.
- *Multiple affordances*, where a single action has multiple sets of associated roles.

The following queries were used as test cases:

Query #1: *Find land parcels which are priced less than $100,000.*

The first query is a selection query where the condition for the selection is the subquery "less than $100,000".

Query #2: *Find the shortest route and its distance from 135 N. Bellefield Ave. to Frick Park.*

The second query contains four subqueries: find the shortest route; locate a place using an address; locate a place using a name; and compute a distance of a route. In addition, an action of locating a place is afforded by two different roles: an address and a name.

Query #3: *Find all hotels located within 5 miles from Lake Erie and compute the distance from the lake to Pittsburgh, PA.*

The third query includes a selection condition "within 5 miles", which is a subquery that is mapped to a compound operator returning a Boolean value. This operator first computes the distance from each of the hotels in Erie to the lake and then determines whether it is less than a fixed quantity "5 miles". The computation of "within" involves a hotel entity, playing the role `location,` and the lake entity, playing the role `territory`, and is hence accomplished by using the `POINT` data model to represent the hotel and the `POLYGON` data model to represent the lake (i.e., the distance is computed from the hotel to the nearest lake's shore).

Typically, the Euclidean distance between two entities in the geographic space is computed based on either the centroids or the boundaries of the entities. In this query where the distance from the lake to Pittsburgh, PA is specified as one of the subqueries, an assumption is made that the subquery intends the distance to be computed in a point-to-point fashion. Hence, the lake entity in this subquery plays the role `location`, which is mapped to the `POINT` data model. As a consequence, a role conversion from `territory` to `location` for the lake entity will be needed.

6.3.2 Representation of Queries

The result of Q-GET for each query is shown in Figure 6.7, Figure 6.8, and Figure 6.9.

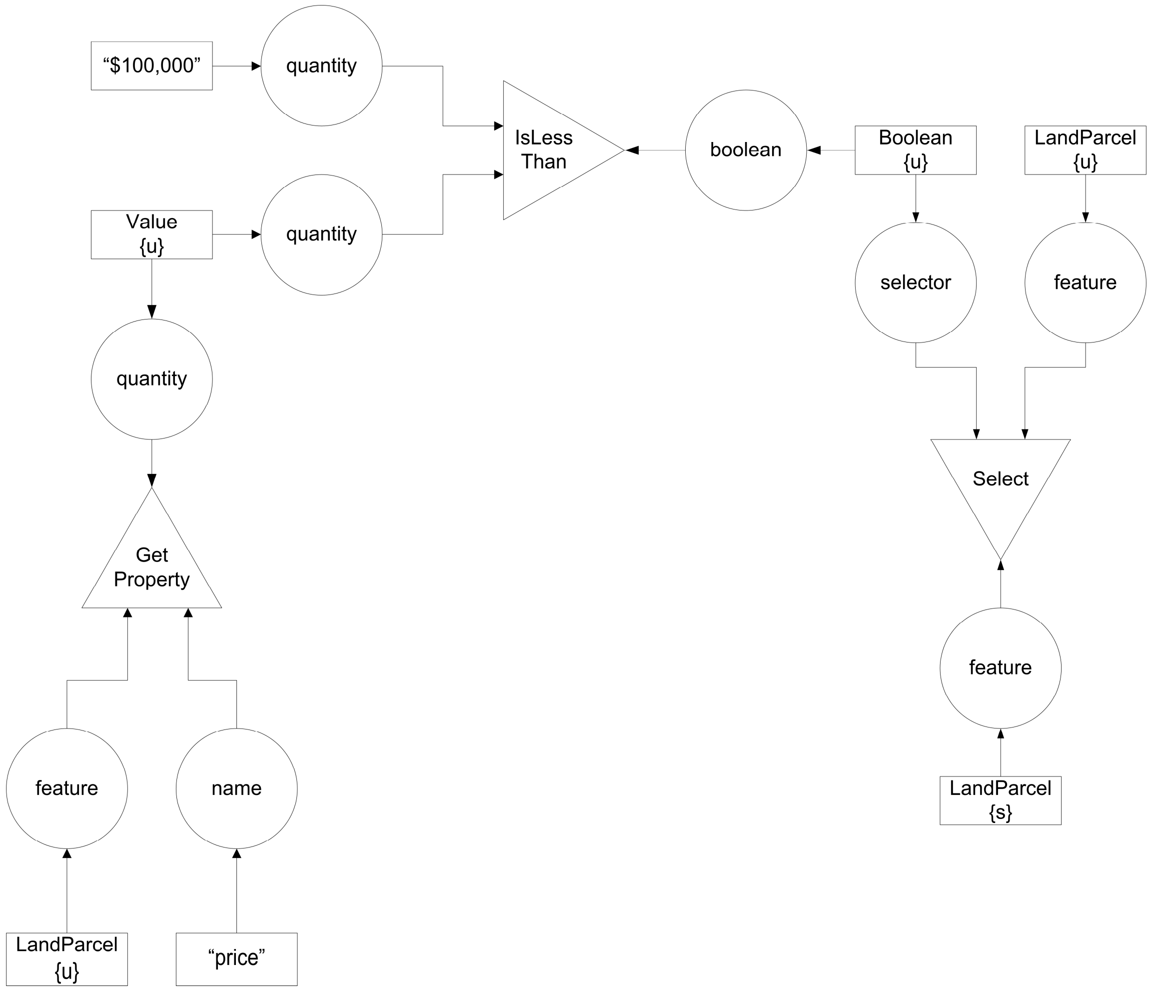


Figure 6.7 Modified CG for Query #1

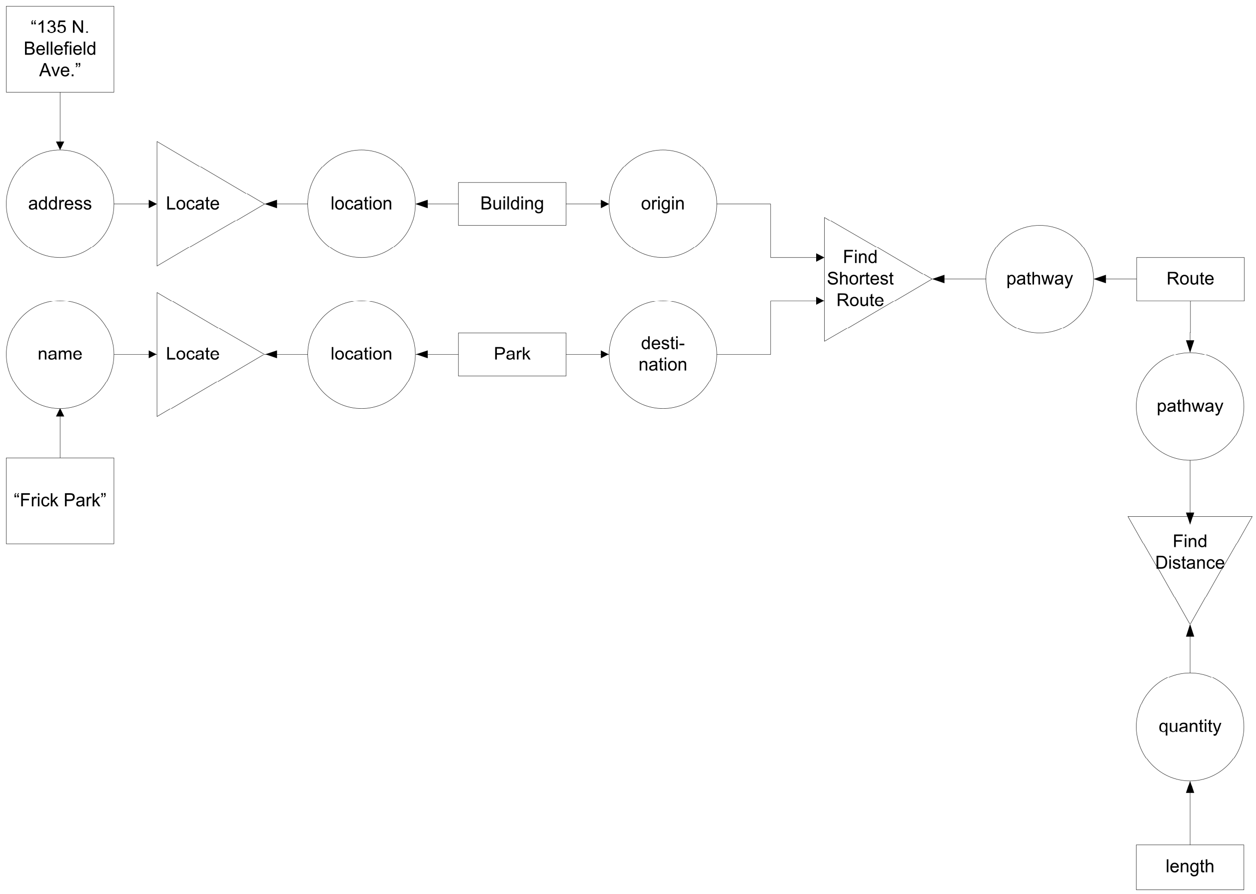


Figure 6.8 Modified CG for Query #2

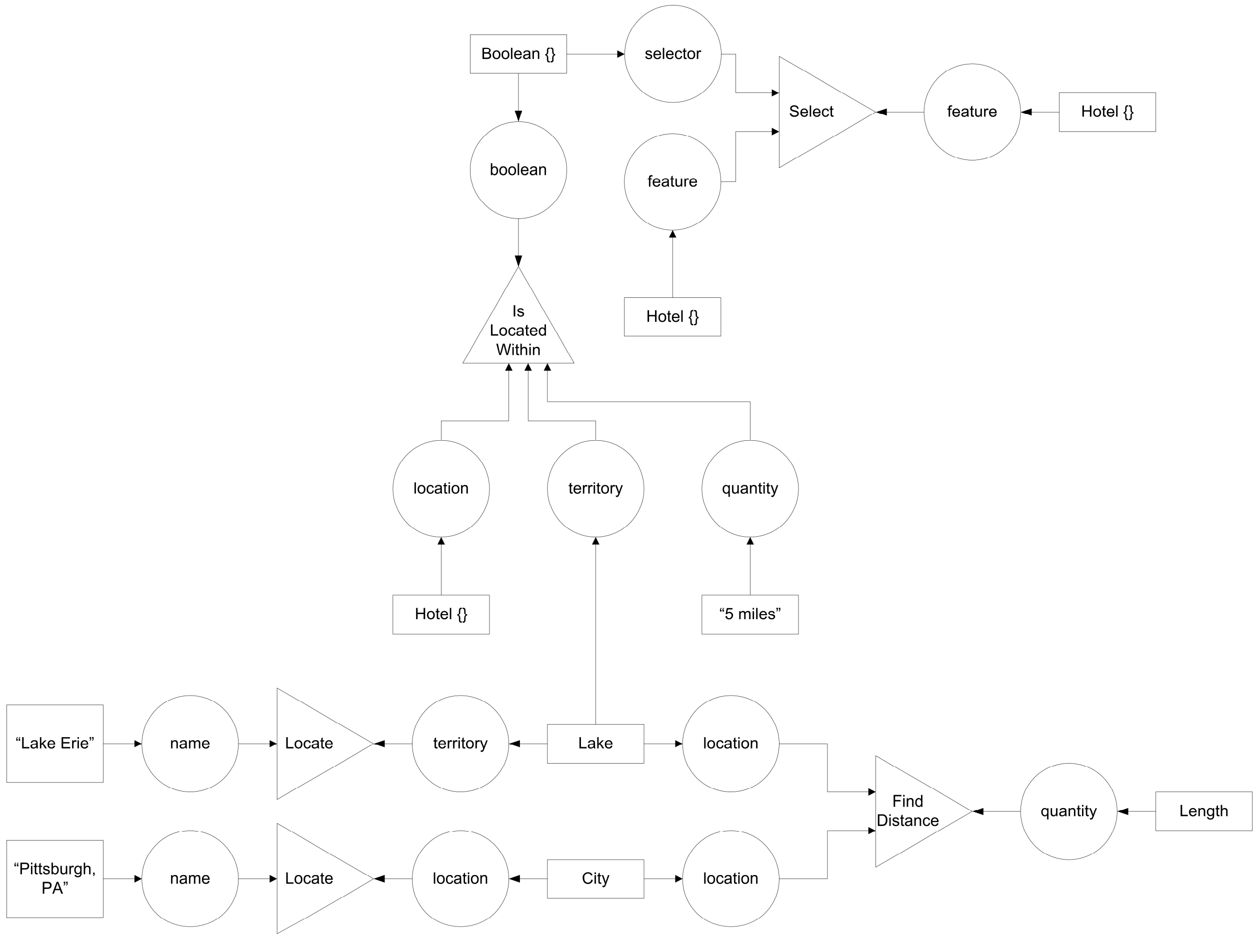


Figure 6.9 Modified CG for Query #3

6.3.3 Results

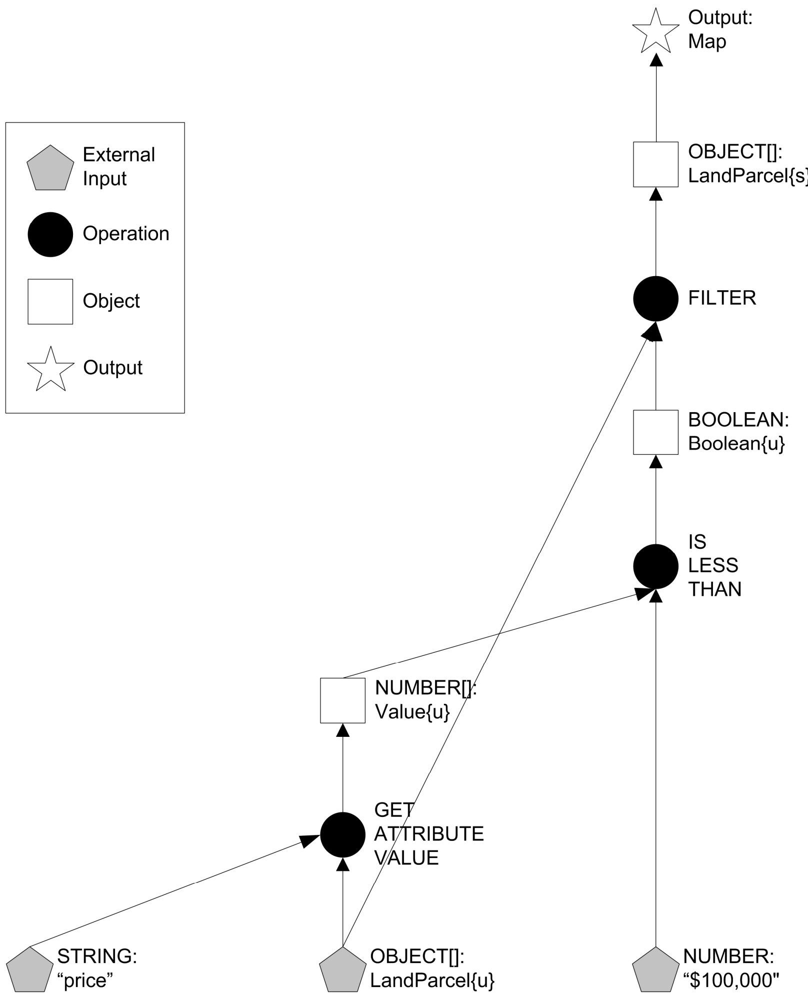


Figure 6.10 Geoprocessing workflow for Query #1

Workflow for Query #1:

```
NUMBER:Value{1}=GETATTRIBUTEVALUE(OBJECT:Land_Parcel{0},STRING:"PRICE")
BOOL:Condition{2}=ISLESSTHAN(NUMBER:Value{1},NUMBER:"100000")
POINT:Land_Parcel{3}=FILTER(OBJECT:Land_Parcel{0},BOOL:Condition{2})

Require OBJECT:Land_Parcel data set(s)
```

For this query, the resultant workflow is as follows (see Figure 6.10): first, the attribute values representing the price of all land parcels are retrieved, then each one is compared against the quantity

"*100,000*". Only the land parcels that satisfy the condition (less than 100,000) are then selected. A data set containing land parcels in `POINT` data model is needed for this query.

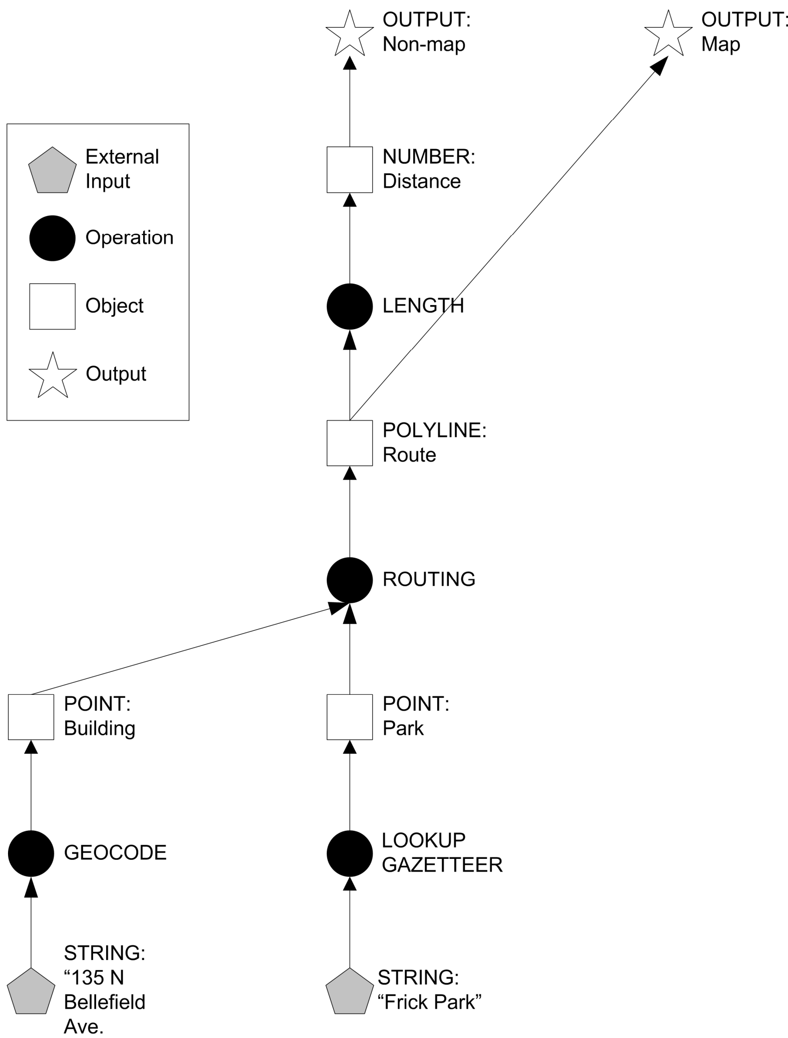


Figure 6.11 Geoprocessing workflow for Query #2

Workflow for Query #2:

```
POINT:Building{1}=GEOCODE(STRING:"135 N. Bellefield Ave.")
POINT:Park{2}=LOOKUPGAZETTEER(STRING:"Frick Park")
POLYLINE:Route{3}=ROUTING(POINT:Building{1},POINT:Park{2})
NUMBER:Distance=LENGTH(POLYLINE:Route{3})
```

For this query, the resulting workflow (see Figure 6.11) starts by obtaining the location of the building specified by the street address "135 N. Bellefield Ave." and the location of a park specified by the name "Frick Park". Once the two locations are obtained, a route is computed and the distance of the resulting route is calculated. No external data set is needed.

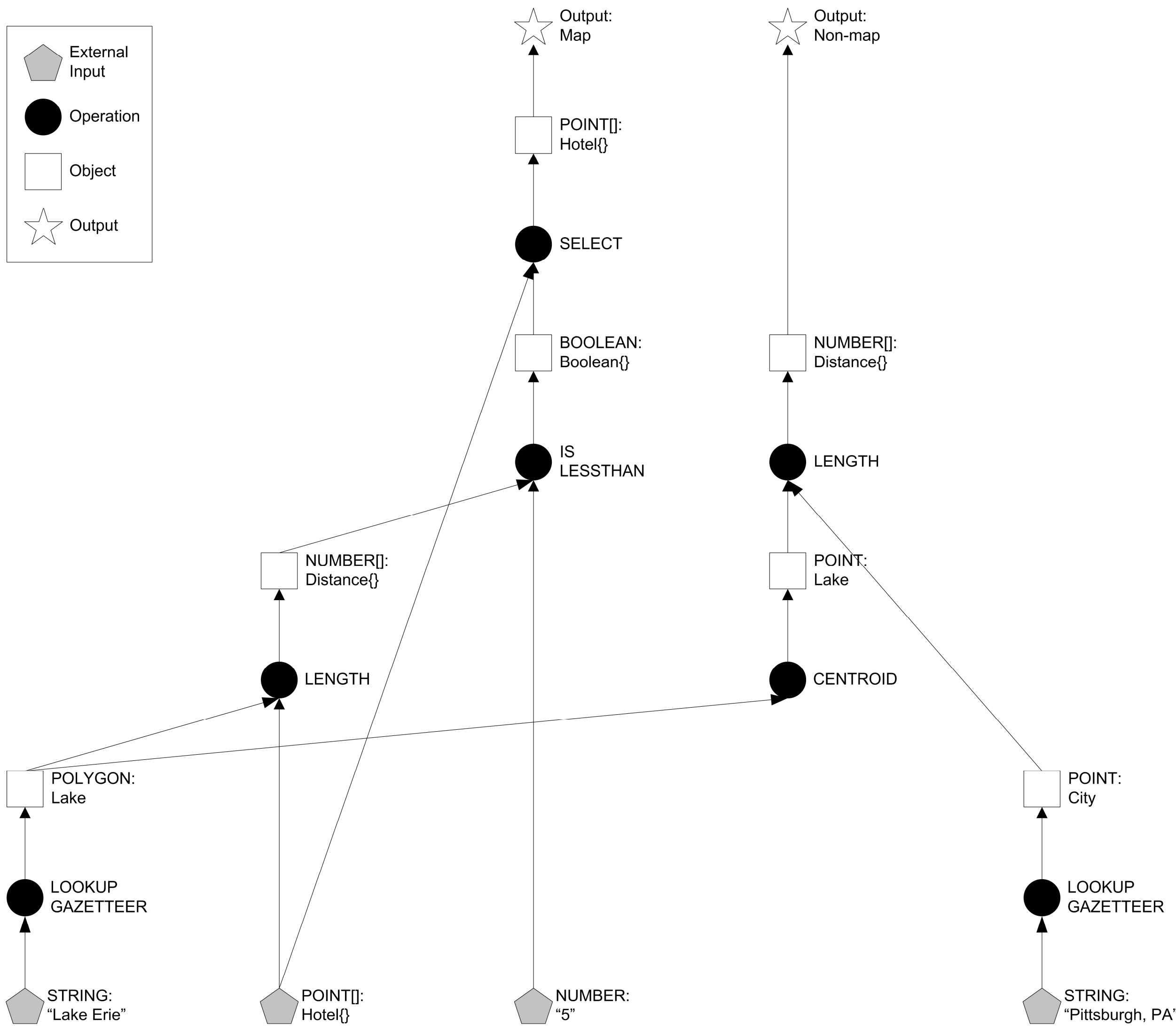


Figure 6.12 Geoprocessing workflow for Query #3

Workflow for Query #3:

```
POLYGON:Lake{1}=LOOKUPGAZETTEER(STRING:"Lake Erie")
NUMBER:Distance{2.1}=LENGTH(POINT:Hotel{0},POLYGON:Lake{1})
BOOL:Condition{2}=ISLESSTHAN(NUMBER:Distance{2.1},NUMBER:"5")
POINT:Hotel{3}=SELECT(POINT:Hotel{0},BOOL:Condition{2})
POINT:City{4}=LOOKUPGAZETTEER(STRING:"Pittsburgh, PA")
POINT:Lake{5.1}=CENTROID(POLYGON:Lake{1})
NUMBER:Distance{5}=LENGTH(POINT:Location{4},POINT:Lake{5.1})
```

```
Require POINT:Hotel data set(s)
```

The workflow for this query (see Figure 6.12) starts by looking up the location of "*Lake Erie*". The resultant location is then used to compute the distance from it to each hotel within the geographic extent. The distance is then compared with a fixed quantity to determine if it is less than "5" miles. The distance and comparison are the result of an operator expansion of the compound operator `WITHIN`. The resulting set of Boolean conditions is then used to select the hotels. Furthermore, the location of "Pittsburgh, PA" is determined by a lookup operator. However, in order to compute the point-to-point Euclidean distance from the lake to Pittsburgh, PA, the lake must be represented as a `POINT` object. Because the first lookup operation for the lake returns a `POLYGON` object (because the original action indicates that the lake must play the role `territory`; and hence the `LOOKUPPOLYGON` operator), a Role Conversion in the form of a `CENTROID` operator is applied to `POLYGON:Lake` to obtain the desired `POINT:Lake`. An external data set containing `POINT` representation of all hotels within the geographic extent is needed.

6.4 CASE STUDY II: APPLICATION-SPECIFIC GEOSPATIAL QUERIES

The second case study incorporated application domain-specific contents into the ontologies. The chosen domain is AEC/FM as many problems in this domain involve the location aspect of physical structures situated in a geographic space.

6.4.1 Background

AEC/FM problem solving traditionally relies heavily on *Computer-Aided Design* (CAD). However, when the location information related to objects is needed, GIS are utilized. For example, during a construction project there are often needs to know detailed information (e.g., building plan) about nearby infrastructures [62]. This type of problems requires the use of both GIS and CAD. A GIS is used for locating and mapping construction sites and analyzing their relationship with nearby infrastructures in a large geographic extent. A CAD is used to analyze and present detailed information about infrastructures or construction sites. Currently, GIS and CAD must be utilized separately for different processing. However, a seamless integration between the two systems has been recognized as a major research goal [55, 67, 68].

Although both GIS and CAD are tools for computer modeling and analysis of real-world physical objects, there are distinctive differences between them. In GIS, objects are *geospatially-referenced* to the surface of the Earth, while in CAD objects are referenced to local Cartesian coordinate systems. GIS typically deals with small-scale objects, while CAD deals with large-scale objects. In addition CAD are traditionally used for drafting, while GIS are used primarily for spatial analysis (Table 6.1).

Table 6.1 Comparisons between CAD and GIS

Characteristic	**CAD**	**GIS**
Reference system	Local Cartesian coordinate reference systems	Geospatial coordinate reference systems
Scale	Large (e.g., buildings, physical structures)	Small (e.g., city-wide, state-wide)
Traditional Usage	Drafting, geometry tools	Mapping, spatial analysis

Nevertheless, the types of queries invoked and processed in a CAD and GIS are similar in many ways. For example, queries on attributes are widely used in CAD to obtain properties related to the modeled objects. An example of this is an object *door* which may have an associated property, such as *color*, stored alongside. Although many geometric computations (e.g., length and area) are used in both CAD and GIS, their scales and reference systems are different. From the GeoInterpret point of view, objects and operations in CAD can be considered an extension of geoprocessing concepts in GPO. Similarly, AEC/FM features can be considered a specialized subclass of GDO.

6.4.2 Application Domain Specialization

An approach to incorporate AEC/FM features into GeoInterpret is to use an existing community standard, *Industry Foundation Classes* (IFC) [54], as the basis for the ontological extension. IFC is modeled as four layers of schemas for different levels of semantics (Figure 6.13). The layers, from the bottom to top, are: *Resource Layer*, *Core Layer*, *Interop Layer*, and *Domain Layer*. The Resource Layer contains *Resource Schema* which defines general-purpose concepts and objects that are independent from applications. The concepts include geometrical and topological entities and are used as the basis by the upper layers. Core Layer contains *Kernel* and *Extension Schema*. Kernel provides basic concepts required for IFC models and its elements are not specific to AEC/FM and may reference elements in Resource layer. Extension Schema extends elements in Kernel by specializing them for use in the AEC/FM domain. Interoperability Layer defines concepts common to multiple AEC/FM sub-domains so that they can be shared. Domain Layer specializes AEC/FM into application-context sub-domains (e.g., Electrical, Architecture, Construction Management).

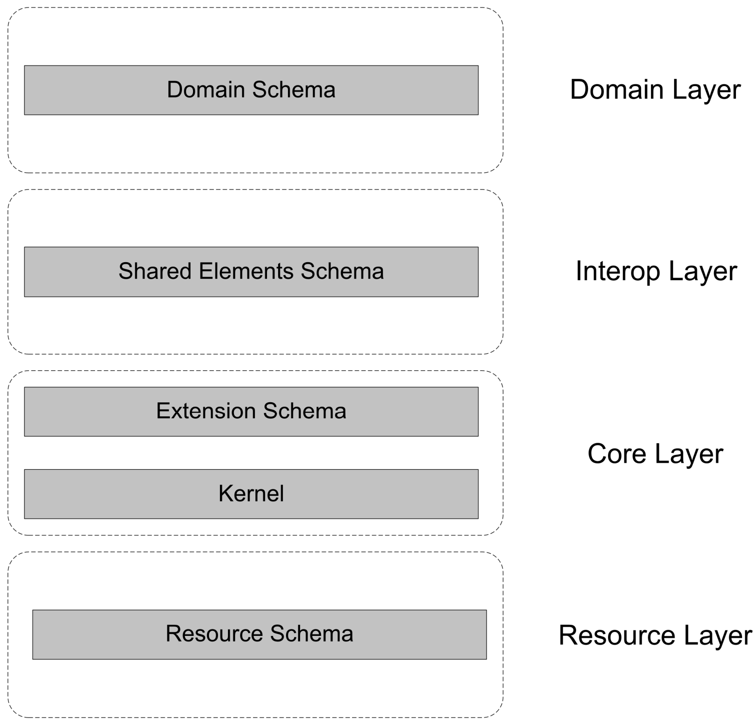


Figure 6.13 IFC architecture (after [54])

AEC/FM and CAD are integrated into GeoInterpret by incorporating elements from IFC to the ontologies (Figure 6.14). The IFC Domain Schema, which contains domain-specific AEC/FM features, is incorporated into GDO, while the IFC Resource Schema, which contains basic geometrical concepts, is incorporated into GPO. Although IFC provides concepts and definition of terms for the AEC/FM domain, it does not provide CAD-specific operators for computational tasks. To allow for the creation of workflows that utilize CAD operators, GPO must be populated with operators common in CAD.

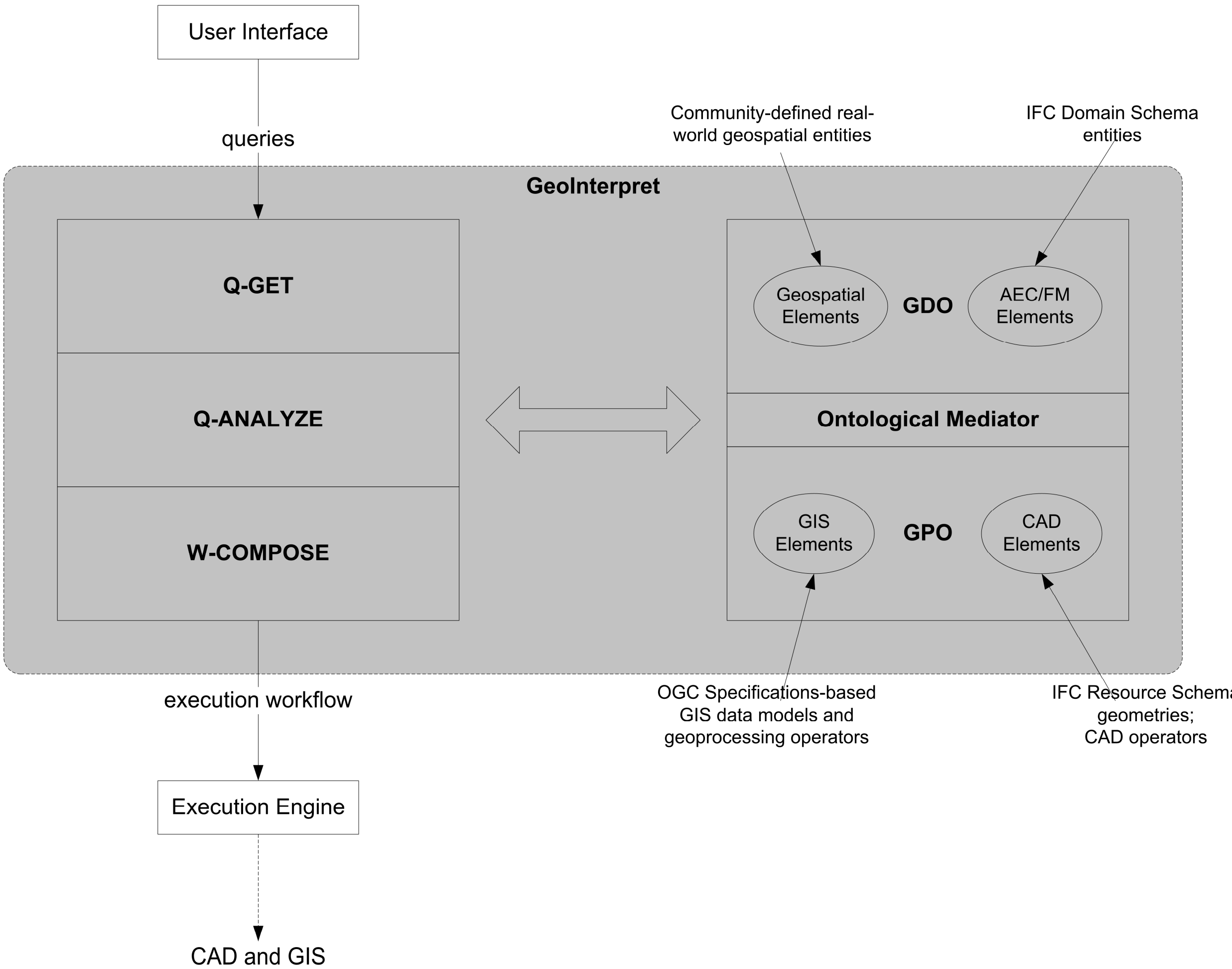


Figure 6.14 IFC knowledge integration to GeoInterpret

6.4.3 Query and Result

Typically, geospatial queries in AEC/FM are formulated to identify physical structures within a geographic extent. Once the structures have been identified, their detailed information is retrieved from CAD files. In this case study, a query test case that utilizes CAD and GIS components is formulated as follows:

> *Retrieve the first floor plans of all adjacent buildings to the construction site. The plan should be displayed in a geographic context highlighting all exit doors.*

For this test query, additional AEC/FM knowledge is added to the ontologies. First, additional roles and entities are added to Ontological Mediator (Table 6.2). In this case, the role `ifcBuilding` (prefix "ifc" indicates that the term is defined in IFC) can be played by entities representing physical structures (e.g., Site, Building, Facility, Warehouse, etc.). The role document represents CAD files, and the role ifcBuildingElement can be played by, for instances, ifcDoor and ifcBeam entities (which represent door and beam, respectively). In IFC, ifcBuildingElement is defined in the Kernel Extension Schema while ifcDoor and ifcBeam are defined in Interop Layer. The new mapping creates a correspondence from Core Layer to Interop Layer of IFC through roles.

Table 6.2 Roles and entities in AEC/FM

Role	Entities that can play the role
ifcBuilding	Site, Building, Facility, Warehouse, etc.
document	FloorPlan, UtilityPlan
ifcBuildingElement	ifcDoor, ifcBeam, etc.

In addition, mapping from entities to CAD data models are also added (Table 6.3) to Ontological Mediator. For the case study, the CAD data model for ifcDoor is ifcSurface, which is a geometrical primitive.

Table 6.3 Entity-role-data model mappings

Entity	Role	Data Model
Building	ifcBuilding	POLYGON
FloorPlan	document	FILE
ifcDoor	ifcBuildingElement	ifcSurface
USER_INPUT	floor	NUMBER

Since standard CAD operators are not defined in IFC, new operators for actions are created for the case study (Table 6.4).

Table 6.4 Action and roles mapped to operators

Action and Roles	Operators
document:[GetFloorPlan]<-(ifcBuilding,floor)	FILE:RETRIEVEFLOORPLAN(POLYGON,NUMBER)
document:[GeographicContext]<-(document)	FILE:AFFINETRANSFORM(FILE)
ifcBuildingElement:[Highlight]<-(document,ifcBuildingElement)	ifcSurface:HIGHLIGHT(FILE,ifcSurface)

The action GetFloorPlan specifies the subquery where a floor plan of a certain object playing the role ifcBuilding are retrieved (the role floor specifies the floor number). The action GeographicContext puts the document in the geographic context by assigning Earth-referenced coordinates to objects in the document (hence, `AFFINETRANSFORM`). `Highlight` is a presentation-oriented action which is mapped to the CAD operator `HIGHLIGHT` for graphically enhancing the presentation of objects.

The query in the case study is represented as modified CG using the Q-GET algorithm (Figure 6.15)

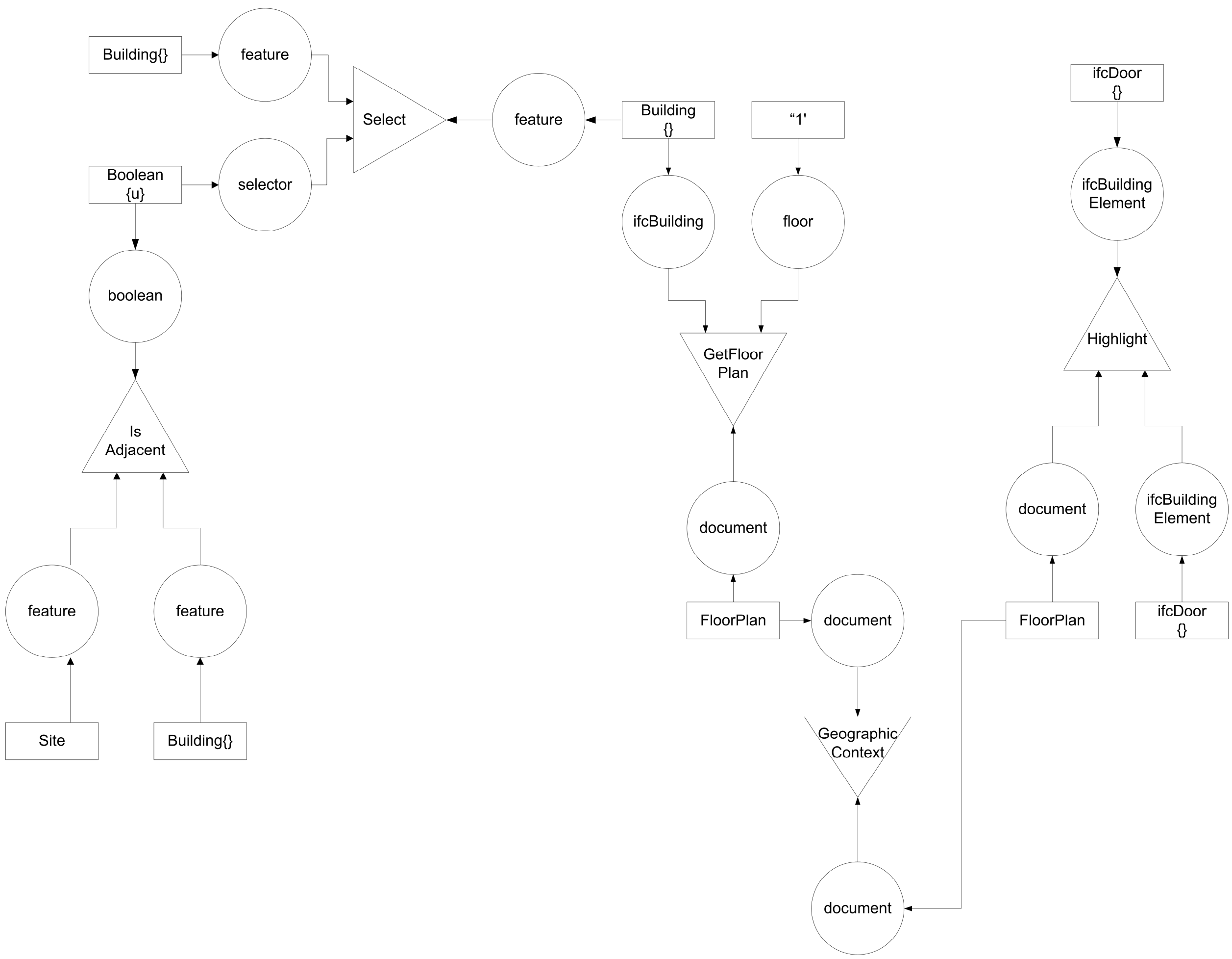


Figure 6.15 Modified CG for the AEC/FM query

The resulting workflow, which includes both CAD and GIS operations is shown in Figure 6.16.

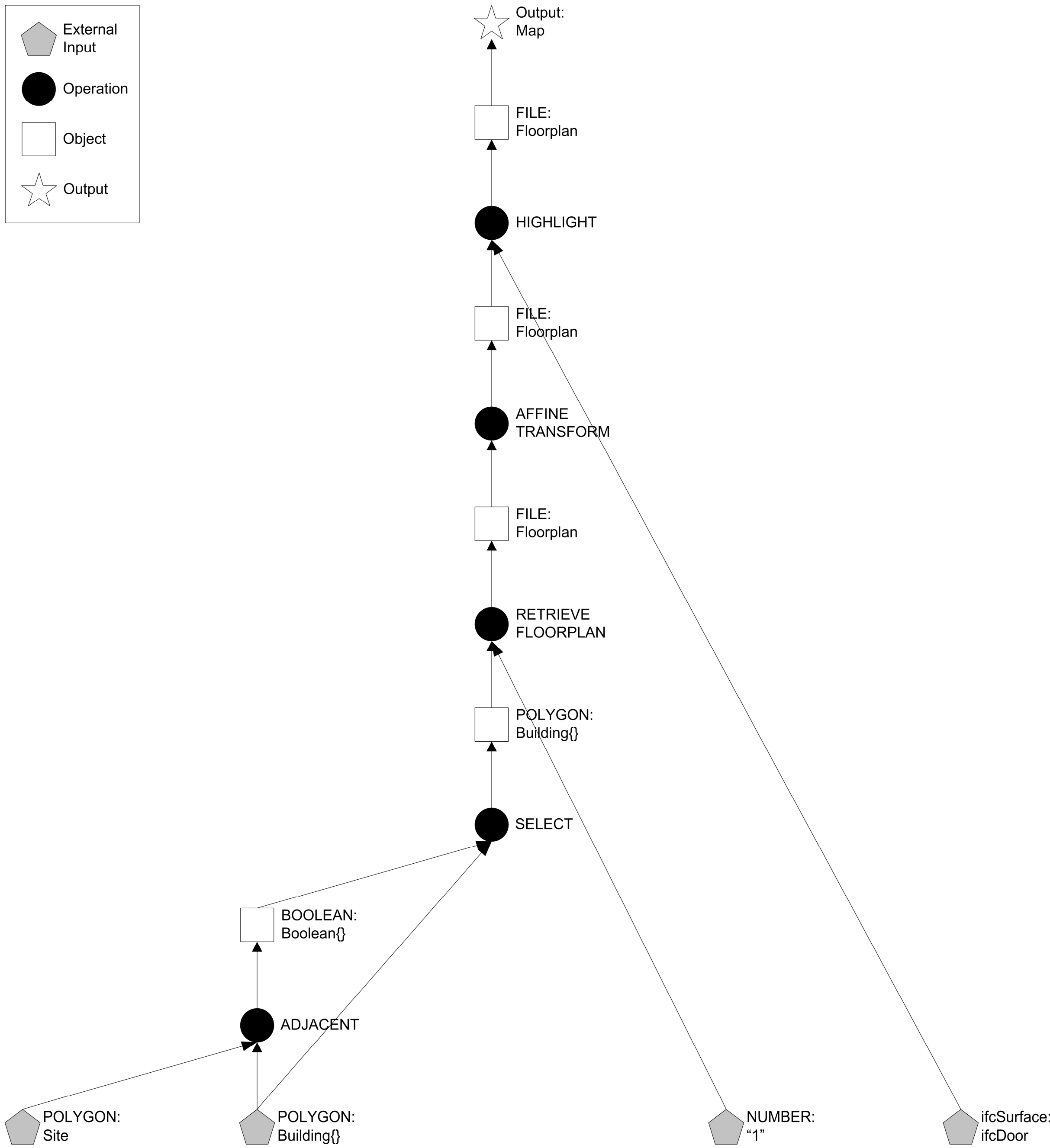


Figure 6.16 Workflow for the AEC/FM query

```
BOOL:Condition{1}=ADJACENT(POLYGON:Site{0},POLYGON:Building{0})
POLYGON:Building{2}=SELECT(POLYGON:Building{0},BOOL:Condition{1})
FILE:FloorPlan{3}=RETRIEVEFLOORPLAN(POLYGON:Building{2},NUMBER:"1")
FILE:FloorPlan{4}=AFFINETRANSFORM(FILE:FloorPlan{3})
FILE:FloorPlan{5}=HIGHLIGHT(FILE:FloorPlan{4},ifcSurface:ifcDoor)
```

```
Require POLYGON:Building and POLYGON:Site data set(s)
```

The workflow for the query begins with an adjacency test of the construction site against all buildings within the geographic extent. The resulting Boolean conditions are used to select only those buildings adjacent to the site. Once the buildings have been identified, three CAD operators are invoked successively to obtain the floor plans, perform affine transformation to put the plans in a geographical context, which allows the doors to be highlighted while preserving their locations relative to the construction site. The query requires two external data sets: one represents all buildings within the geographic extent in the form of `POLYGON` objects and another represents the `POLYGON` of the construction site itself.

6.5 SUMMARY

This chapter discusses the proof of concept that demonstrates the working of the developed methodology. Generic geospatial queries were used as test cases for verifying that the mapping techniques and algorithms work as expected. In addition, a query expressed in application domain-specific concepts and terminology was used to demonstrate the integration of application-specific ontological elements to GeoInterpret.

7.0 CONCLUSIONS AND FUTURE RESEARCH

7.1 SUMMARY OF THE RESEARCH

This research provides a methodology for automating the task of geospatial query interpretation, which is currently performed manually by GIS users posing a significant burden, time and cost on them. The premise of the methodology, called GeoInterpret, is a set of ontologies as the knowledge base which captures the requisite semantic knowledge necessary to interpret queries. GeoInterpret by itself does not create ontological contents, but rather it provides a set of knowledge representation, techniques, and algorithms for using ontologies to perform query interpretation. One of the main features of GeoInterpret is that it is implementation independent. The ultimate goal of GeoInterpret is to develop methodologies and techniques that will pave the way for the emergence of a new paradigm in geospatial problem solving. The main contributions of this research can be summarized as follows:

- A new ontological-based methodology for automating the tasks involved in geospatial query interpretation.
- A general conceptualization of geospatial queries based on typical queries in various application domains. The conceptualization includes the structure, components, types, and different formulation phases of geospatial queries (Chapter 3).
- An architecture based on multiple-ontology approach that distinguishes different levels of geospatial semantic. The architecture offers techniques for mappings between ontologies and a knowledge representation of geospatial queries for the purpose of query interpretation (Chapter 4).
- A set of algorithms that utilizes the ontologies to perform geospatial query interpretation (Chapter 5).
- A proof of concept that demonstrates the working of GeoInterpret including the mapping techniques and the algorithms (Chapter 6).

7.2 CONCLUSIONS

The GeoInterpret methodology discussed in this dissertation has the potential to eliminate the need for users to be trained in the intricacy of geoprocessing and the use of GIS to solve geospatial problems in application domains. Based on the research, several conclusions can be drawn:

- An ontological-based approach deems suitable for automating the tasks related to geospatial query interpretation by providing a common conceptualization of queries and geoprocessing tasks.
- Multiple ontologies are needed in order to capture different levels of geospatial concepts. The mappings between ontologies are the knowledge required to perform the interpretation of queries.
- The multiple-ontology approach is adaptable by allowing for flexible integration of domain-specialized knowledge.
- A query-based approach is seen as a means of paving the way for a new paradigm in geospatial problem solving.
- To facilitate ontological-based solutions, there is a need for semantic agreements among users and systems in the form of standards and formal ontologies.

7.3 FUTURE RESEARCH

Because research in the area of ontology in information systems has only recently become widespread, especially in the field of GIS, many new research opportunities remain to be addressed. In addition, advancements in such fields as Semantic Web and Grid Computing have the potential to enable semantically-aware information retrieval in heterogeneous networked environments, where ontologies will play a major role in representing domain semantics in a manner that can automatically be processed by computer systems. In the context of this dissertation, several future research ideas are identified as follows:

- *Methodologies and techniques for automating the task of geoprocessing workflow implementation*. In the current form, GeoInterpret provides a geoprocessing workflow for a given query which must be traversed in order to obtain the solution to the query. New techniques and algorithms are needed to automatically implement the workflows in an efficient manner.
- *Ontologies for the real-world geospatial domain, geoprocessing domain, and application domains.* This dissertation provides a methodology for query interpretation that integrates and

utilizes ontological contents both at the application level and geoprocessing level. However, the ontologies themselves remain to be conceived by respective research communities. At the geoprocessing level, the OGC has laid the groundwork for interoperable geoprocessing semantics that include both geospatial data models and operations [52]. At the application domain-level, the geospatial information science community is actively pursuing research on geospatial ontologies that represent real-world geospatial concepts and entities [6, 33, 81, 102, 128]. For specialized application domains, the task of creating ontologies lies in the hand of the respective communities (e.g., the effort to create interoperable data model for the AEC/FM domain [54, 55]).

- *Improvements in user interfaces that make use of ontologies in order for non-expert users to fully benefit from ontological-based GIS*. Although commercial GIS software vendors constantly make improvements in how users interact with GIS, those improvements are typically incremental and do not make fundamental changes in how application-level semantic contents are integrated to the systems. The need for new approaches to GIS user interface has been identified [7, 12] and through GeoInterpret (i.e., the ontological engineering methodology), generic GIS platforms can be made domain-specific without the loss of geoprocessing power or generality.
- *Methodologies and techniques related to distributed geoprocessing that make use of geoprocessing ontologies and workflows*. Current GIS platforms primarily implement geoprocessing tasks sequentially within the confine of stand-alone computing environments. As new distributed computing paradigms emerge (e.g., Grid Computing [14, 36, 37]), distributed geoprocessing is a logical next step to be applied to such environments. Geoprocessing ontologies and workflows can be used as the basis for enabling distributed geoprocessing in heterogeneous environments as they provide a common and shared knowledge for implementing geoprocessing tasks. To allow for automated distributed geoprocessing, many research issues remain to be addressed including geoprocessing service discovery, invocation, and optimization [7, 66, 76, 77, 100]. Interoperability will play a crucial role in this environment due to the heterogeneity of geoprocessing platforms.

APPENDIX: SAMPLE GEOSPATIAL QUERIES

Public Health

- Create a “dot” map that shows incidences of asthma.
- Create a “choropleth” map that shows incidences of asthma.
- Create a map that shows pollution sources.
- Estimate level of air pollution at a location adjacent to a road.
- Estimate levels of water quality of stream segments on the basis of data from monitoring sites.
- Create a map that shows voronoi polygons around air pollution monitoring sites.
- Integrate infant mortality data into census district areas by interpolation.
- Create a map that shows potential area where an individual could travel based on person’s home location, workplace, and church.
- Create a table that shows health markets with populations greater than or equal to the threshold needed to support managed competition.
- Create a table reporting age-specific cancer incidence rates per 100,000 population by ethnic group for all regions.
- Create a map that shows how fast the infection is spreading.
- Create a map that shows Townsend Deprivation Index based on census data.
- Create a map that shows localized industrial chimneys pollution sources.
- Create a map that shows roads as pollution sources.
- Create a map that shows agricultural activities as diffused pollution sources.
- Create a map that shows controlled pathway such as stacks or discharge pipes as pollution sources.
- Create a map that shows fugitive emissions – leaks.
- Create a map that shows difference in pollution sources from last year to this year.
- Predict levels of air pollution at a location between two monitoring stations.
- Predict levels of air pollution at a location within a mapped area of contamination.
- Extrapolate the water quality classes from classified segments to unclassified segments of the water stream network.

- Create a map that shows migration of population.
- Create a map that shows census block areas that received contaminated drinking water from wells adjacent to a National Priority List hazardous waste site.
- Create a map that shows exposure zone where children might lived based on position of the street network in relation to the exposure zone and the child population of the census block where the street segment is located.
- Create a map that shows temporal peaks in rotavirus infection.

Business

- Create a map that shows real estate properties satisfying these conditions – zip code, proximity to shopping facilities, square feet.
- Create a map that displays "drive time" of areas.
- Create a map that shows the ratio of loans to deposits to see evidence of discriminatory credit practices.
- Create a map that shows available land parcels that are near major roads and are reachable within 1 hour of driving by at least 1 million people.
- Create a map that shows different areas with different customer potential based on demographic.
- Create a chart that shows drive time to store on the X-axis and percent of customers on the Y-axis.
- Create a map that shows car theft incidence on travel routes.
- Create a map that shows catchment area of 15-km from "A" dealers.
- Create a map that shows percentage of adults eating at steak houses in the last 3 months.
- Create a map that shows targeted customer areas for marketing plan.
- Create a map that show logistic of baling operations on each farm field, including shortest path from each field to warehouse.
- Create a map that show field size, type of wheat straw planted on the field.
- Create a map that shows real estate properties and public data sets associated with each property such as appraisal, footprints, certificate of occupancy.
- Create a map that shows locations of mines and its accessible relationship to transport routes.
- Create a map that shows sales performance by regions.
- Create a map that shows potential eat-out customers.
- Create a map that shows bank branches and/or ATM machines.

- Create a map that shows median household income by areas.
- Create a map that shows credit needs of community based on demographic.
- Create a map that shows Volvo car segment as percentage of total market.

Ecology

- Create a map that shows the number of dead wildlife per square mile in Rockland County, NY.
- Create a map that shows areas that are covered by biodiversity-rich tropical evergreen broadleaf forest in Africa overlaid by policy-protected areas.
- Create a map that shows land uses by category – commercial, industrial, residential, forest, grass, agriculture, water.
- Create a map that shows alternative land use scenarios and create a chart of runoff of present scenario and alternative scenarios.
- Create a map that shows management (prescribed) burns and wildfires recorded.
- Create a map that shows designated sheep restriction zones overlaid on areas with large sheep population and high rainfall.
- Create a map that shows soil erosion potential for the area of study in tons per acre per year.
- Create a map that shows hierarchy of settlements in Somerset, England – classify based on 5, 10, and 20 shops. Urban centers are distinguished.
- Create a map that shows areas of likelihood afforestation within an area by using altitude data and relative utility of the class of forest data.
- Create a map that shows habitat suitability for red deer based on altitude, land cover, and accumulated frost data and rules.
- Create a map that shows “environmental domains” for northern Australia (coastline, island, dessert, etc.).
- Create a map that shows areas with population density greater than 100/km2 overlaid by area with agricultural potential.
- Create a map that shows distribution of rainforest types.
- Create a map that shows different areas of ecological stability of Slovakia (unprotected, ecologically important, most threatened, etc.).

Construction Planning

- Create a map that shows construction sites on campus and identify whether there is a traffic delay or detour near the sites.
- Create a map that shows underground water pipes distinguished by their materials (cast iron, concrete, PVC, steel) by different colors.
- Create a map that shows levels of soil strength at the construction site.
- Create a map that shows elevation terrain around the construction site.
- Create a map that shows a 1-km service network from the construction site.
- Create a map that shows a 1-km service area from the construction site overlaid with houses.
- Create a map that shows public drinking water system including wells, water mains, and reservoirs.

Transportation

- Create a map that identifies clusters of city blocks that house families with socioeconomic and demographic characteristics conductive to transit ridership (family size, income, parking rates, distance to work).
- Create a map that shows the optimal transit route from origin to destination.
- Create a map that shows levels of congestions of street network.
- Create a map that shows incidences of traffic accidents on street network.
- Create a map that shows the location of fire, the location of fire stations, and the drive time from each fire station to the fire location.
- Create a map that shows key roadway attributes (speed limits, traffic signs and signals, crosswalks) in areas considered to be within walking distance from an elementary school for distribution to students.
- Estimate the time of arrival of the shipment currently on a traveling truck.
- Create a map that shows dynamic segmentations of streets according to the type of pavement (paved, gravel).
- Create a map that shows dynamic segmentations of streets according to the posted speed limits.
- Create a map that shows terrain constraints, bike routes, and existing commuting patterns for a park.
- Create a map that shows the shortest path through a street network from an ambulance dispatch site to a location of an emergency call.

- Create a map that show confirmed pedestrian injuries in the context of reported total traffic injuries in a neighborhood in Boston.

Sources:

1. *GIS in Business '93 : conference proceedings*. GIS World, Fort Collins, Colo., 1993.

2. Corvalâan, C., Briggs, D. and Zielhuis, G. *Decision-making in environmental health : from evidence to action*. Published on behalf of the World Health Organization by E & FN Spon, London ; New York, 2000.

3. Cromley, E.K. and McLafferty, S. *GIS and public health*. Guilford Press, New York, 2002.

4. Easa, S., Chan, Y. and American Society of Civil Engineers. Geographic Information Systems Committee. *Urban planning and development applications of GIS*. American Society of Civil Engineers, Reston, Va., 2000.

5. Grimshaw, D.J. *Bringing geographical information systems into business*. John Wiley, New York, 1999.

6. Gurnell, A.M. and Montgomery, D.R. *Hydrological applications of GIS*. John Wiley, Chichester ; New York, 2000.

7. Haines-Young, R.H., Green, D.R. and Cousins, S. *Landscape ecology and geographic information systems*. Taylor & Francis, BLondon ; New York, 1993.

8. Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W. *Geographic information systems and science*. Wiley, Chichester ; New York, 2001.

9. Mather, P.M. *Geographical information handling : research and applications*. Wiley & Sons, Chichester England ; New York, 1993.

10. Melnick, A.L. *Introduction to geographic information systems in public health*. Aspen Publishers, Gaithersburg, Md., 2002.

11. O'Sullivan, D. and Unwin, D. *Geographic information analysis*. J. Wiley, Hoboken, N.J., 2003.

12. Papacostas, C.S. and Prevedouros, P.D. *Transportation engineering and planning*. Prentice Hall, Upper Saddle River, NJ, 2001.

13. Shamsi, U.M. *GIS tools for water, wastewater, and stormwater systems*. American Society of Civil Engineers, Reston, Va., 2002.

14. Verbyla, D.L. *Practical GIS analysis*. Taylor & Francis, London ; New York, 2002.

BIBLIOGRAPHY

1. Abel, D., Ooi, B., Tan, K. and Tan, S. Towards integrated geographical information processing. *International Journal of Geographical Information Science*, *12* (4). 353-371, 1998.

2. Adnani, M., Yetongnon, K. and Benslimane, D., A multiple layered functional data model to support multiple representations and interoperability of GIS: application to urban management systems. in *Ninth ACM international symposium on Advances in geographic information systems*, (Atlanta, GA, 2001), ACM Press, 70-75.

3. Albano, A., Bergamini, G., Ghelli, G. and Orsini, R., An object data model with roles. in *19th VLDB Conference*, (Dublin, Ireland, 1993), Morgan Kaufmann Publishers, San Mateo, CA, USA, 39-51.

4. Algernon, Algernon - Rule-Based Programming, 2004, http://algernon-j.sourceforge.net/.

5. ANSI/NISO, ANSI/NISO Z39.19-1993: Guidelines for the construction, format, and management of monolingual thesauri, National Information Standards Organization, 1993, http://www.niso.org/standards/standard_detail.cfm?std_id=518.

6. Bernard, L., Einspanier, U., Haubrock, S., Hubner, S., Kuhn, W., Lessing, R., Lutz, M. and Visser, U. Ontologies for Intelligent Search and Semantic Translation in Spatial Data Infrastructure. *Photogrammetrie - Fernerkundung - Geoinformation (PFG)* (6). 451-462, 2003.

7. Bernard, L., Einspanier, U., Lutz, M. and Portele, C., Interoperability in GI service chains – The way forward. in *6th AGILE Conference on Geographic Information Science 2003*, (Lyon, France, 2003), 179-188.

8. Bishr, Y. Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science*, *12* (4). 299-314, 1998.

9. Briggs, D.J. Mapping environmental exposures. in Elliott, P., Cuzick, J. and Stern, R. eds. *Geographical and Environmental Epidemiology*, Oxford University Press, Oxford, 1992, 158-176.

10. Brodie, M.L., The promise of distributed computing and the challenges of legacy information systems. in *Advanced Database Systems: 10th British National Conference on Databases*, (1992), Springer-Verlag, 1-28.

11. Burrough, P.A. and McDonnell, R. *Principles of geographical information systems*. Oxford University Press, Oxford ; New York, 1998.

12. Cai, G., GeoVSM: An Integrated Retrieval Model for Geographic Information. in *GIScience 2002, LNCS 2478*, (2002), Springer-Verlag, 65-79.

13. Camara, G., Monteiro, A.M., Paiva, J. and R., S., Action-driven ontologies of the geographical space: Beyond the field-object debate. in *GIScience 2000*, (Savannah, GA, 2000), http://www.giscience.org/GIScience2000/papers/210-Camara.pdf.

14. Casanova, H. Distributed computing research issues for grid computing. *Quarterly Newsletter for the ACM Special Interest Group on Algorithms and Computation Theory (SIGACT News)*, *33* (2), 2002.

15. Clarke, K.C. *Getting started with geographic information systems*. Prentice Hall, Upper Saddle River, N.J., 1999.

16. Couclelis, H., People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. in *International Conference on GIS -- From Space to 10 Territory: Theories and Methods of Spatial Reasoning, Notes in Computer Science 639*, (1992), Springer Verlag, 65-77.

17. Cox, S., Daisey, P., Lake, R., Portele, C. and Whiteside, A., Open GIS Geography Markup Language (GML) Implementation Specification, 2003, http://www.opengeospatial.org/specs/?page=specs.

18. Cruz, I., Sunna, W. and Chaudhry, A., Semi-Automatic Ontology Alignment for Geospatial Data Integration. in *GIScience 2004*, (Adelphi, MD, 2004), 51-66.

19. Daconta, M.C., Obrst, L.J. and Smith, K.T. *The Semantic Web : a guide to the future of XML, Web services, and knowledge management*. Wiley Pub., Indianapolis, Ind., 2003.

20. De Graauw, M., What is 'is'? in *XML Conference and Exposition 2001*, (Orlando, FL, 2001), http://www.marcdegraauw.com/files/whatisis.pdf.

21. Dehn, M., Gartner, H. and Kikau, R., Principles of semantic modeling of landform structures. in *GeoComputation 99*, (Fredericksburg, VA, 1999), 1005-1010, http://www.geovista.psu.edu/sites/geocomp99/Gc99/067/gc_067.htm.

22. DeMers, M.N. *Fundamentals of geographic information systems*. J. Wiley, New York, 2000.

23. Egenhofer, M.J., Toward the Semantic Geospatial Web. in *Tenth ACM International Symposium on Advances in Geographic Information Systems*, (McLean, Virginia, USA, 2002), ACM, 1-4.

24. Egenhofer, M.J., Clementini, E. and Di Felice, P. Topological relations between regions with holes. *International Journal of Geographical Information Systems*, *8* (2). 129-144, 1994.

25. Egenhofer, M.J. and Franzosa, R. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, *5* (2). 161-174, 1991.

26. Egenhofer, M.J. and Mark, D.M., Naive geography. in *COSIT'95*, (Semmering, Austria, 1995), Springer-Verlag, 1-15.

27. ESRI, ESRI Inc., 2004, http://www.esri.com.

28. Feng, C., Bittner, T. and Flewelling, D., Modeling Surface Hydrology Concepts with Endurance and Perdurance. in *GIScience 2004*, (Adelphi, MD, 2004), 67-80.

29. Feng, C. and Sorokine, A., Incorporating hydrologic semantic information for interoperable GIS with hydrologic model. in *Ninth ACM international symposium on Advances in geographic information systems*, (Atlanta, Georgia, USA, 2001), ACM Press, 59-63.

30. FGDC, Content Standard for Digital Geospatial Metadata, Federal Geographic Data Committee, 1998, http://fgdc.gov/metadata/csdgm/.

31. FGDC, Federal Geographic Data Committee, 2003, http://www.fgdc.gov/.

32. Fonseca, F.T., Ontology-driven geographic information systems, Ph.D. Thesis, University of Maine, 2001.

33. Fonseca, F.T., Davis, C. and Camara, C. Bridging Ontologies and Conceptual Schemas in Geographic Information Integration. *GeoInformatica*, *7* (4). 355-378, 2003.

34. Fonseca, F.T., Egenhofer, M.J., Agouris, P. and Camara, C. Using ontologies for integrated geographic information systems. *Transactions in GIS*, *6* (3). 231-257, 2002.

35. Fonseca, F.T., Egenhofer, M.J., Davis, C. and Camara, G. Semantic granularity in ontology-driven geographic information systems. *Annals of Mathematics and Artificial Intelligence, Kluwer Academic Publishers, 36* (1-2). 121-151, 2002.

36. Foster, I., Kesselman, C., Nick, J.M. and Tuecke, S., The physiology of the grid: an open grid services architecture for distributed systems integration, Open Grid Service Infrastructure WG, Global Grid Forum, 2002, www.globus.org/research/papers/ogsa.pdf.

37. Foster, I., Kesselman, C. and Tuecke, S. The anatomy of the grid-enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, *15* (3). 200-222, 2001.

38. Frank, S.M., Goodchild, M.F., Onsrud, H.J. and Pinto, J.K., Framework data sets for the NSDI, The Federal Geographic Data Committee, 1996, http://www.fgdc.gov/framework/refmat/ncgia1.html.

39. Frege, G. On sense and reference. in Geach, P. and Black, M. eds. *Translations from the Philosophical Writing of Gottlob Frege*, Blackwell, Oxford, 1960, 58.

40. Gahegan, M., Brodaric, B. and Takatsuka, M., Designing a framework to support the semantic interoperability of geospatial information. in *GIScience 2000: The First International Conference on Geographic Information Science*, (2000), www.geovista.psu.edu/publications/ aag99vr/GIScience2000/102-Gahegan.pdf.

41. Gâomez-Pâerez, A., Fernâandez-Lâopez, M. and Corcho, O. *Ontological engineering : with examples from the areas of knowledge management, e-commerce and the semantic Web Asunciâon Gâomez-Pâerez, Mariano Fernâandez-Lâopez, and Oscar Corcho*. Springer-Verlag, London ; New York, 2004.

42. Gibson, J.J. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, 1979.

43. Gibson, J.J. The theory of affordances. in Shaw, R.E. and Bransford, J. eds. *Perceiving, Acting, and Knowing.*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

44. Giordano, A., Veregin, H., Borak, E. and Lanter, D. A conceptual model of GIS-based spatial analysis. *Cartographica*, *31* (4). 44-57, 1994.

45. Gomes, J. and Velho, L. Abstraction paradigms for computer graphics. *The Visual Computer*, *11* (5). 227-239, 1994.

46. Goodchild, M.F., Egenhofer, M.J. and Fegeas, R., Interoperating GISs: report of the specialist meeting, National Center for Geographic Information and Analysis, University of California, Santa Barbara, 1998, http://www.spatial.maine.edu/~max/InteropReport.pdf.

47. Gruber, T.R. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human and Computer Studies*, *43* (5/6). 907-928, 1993.

48. Guarino, N. Formal ontology and information systems. *First International Conference on Formal Ontologies in Information Systems, FOIS'98*. 3-15, 1998.

49. Guarino, N., Formal ontology and information systems. in *First International Conference on Formal Ontologies in Information Systems, FOIS'98*, (Trento, Italy, 1998), IOS Press, 3-15.

50. Guarino, N. and Welty, C., A Formal Ontology of Properties. in *The 12th International Conference on Knowledge Engineering and Knowledge Management.*, (2000), 97-112.

51. Harvey, F., Kuhn, W., Pundt, H. and Bishr, Y. Semantic interoperability: a central issue for sharing geographic information. *The Annals of Regional Science*, *33*. 213-232, 1999.

52. Herring, J., The OpenGIS Abstract Specification, Topic 1: Feature Geometry (ISO 19107 Spatial Schema), Version 5. OGC Document Number 01-101., 2001, http://www.opengis.org/docs/01-101.pdf.

53. Hickman, C., SDTS Profiles. in *FGDC SDTS Workshop*, (University of Missouri - Rolla, 1997), http://mcmcweb.er.usgs.gov/sdts/profiles/index.html.

54. IAI, IFC 2x Technical Guide, 2000, http://www.iai-international.org/iai_international/Technical_Documents/documentation/IFC_2x_Technical_Guide.pdf.

55. IFG, Industry Foundation Classes for GIS (IFG), 2004, http://www.iai.no/ifg/index.html.

56. ISO The OpenGIS Abstract Specification, Topic 11: OpenGIS Metadata (ISO/TC 211 DIS 19115), Version 5. OGC Document Number 01-111., 2001.

57. Java, The Java Programming Language, 2004, http://java.sun.com.

58. Jensen, J.R. *Introductory digital image processing: a remote sensing perspective*. Prentice Hall, Upper Saddle River, N.J., 1996.

59. Jones, C., Abdelmoty, A., Finch, D., Fu, G. and Vaid, S., The SPIRIT Spatial Search Engine: Architecture, Ontologies, and Spatial Indexing. in *GIScience 2004*, (Adelphi, MD, 2004), 125-139.

60. Jordan, T., Raubal, M., Gartrell, B. and Egenhofer, M., An Affordance-Based Model of Place in GIS. in *8th International Symposium on Spatial Data Handling*, (Vancouver, Canada, 1998), 98-109.

61. Karimi, H.A. Real-time optimal-route computation: a heuristic approach. *Intelligent Transportation Systems Journal*, *3* (2). 111-127, 1996.

62. Karimi, H.A., Akinci, B., Boukamp, F. and Peachavanish, R., Semantic interoperability in infrastructure systems. in *The 4th Joint Symposium on Information Technology in Civil Engineering*, (Nashville, TN, 2003), 1-13.

63. Karimi, H.A., Durcik, M. and Rasdorf, W. Evaluation of Uncertainties Associated with Geocoding Techniques. *Journal of Computer-Aided Civil and Infrastructure Engineering*, *19* (3). 170-185, 2003.

64. Karimi, H.A. and Peachavanish, R. Interoperability in Geospatial Information Systems. in Khosrow-Pour, M. ed. *Encyclopedia of Information Science and Technology (in press)*, Idea Group Inc., Hershey, PA, 2004.

65. Karimi, H.A., Peachavanish, R. and Peng, J. Finding optimal bus service routes: an internet-based methodology to serve the transit patrons. *ASCE Journal of Computing in Civil Engineering*, *18* (2). 83-91, 2003.

66. Klien, E., Einspanier, U., Lutz, M. and Hubner, S., An architecture for ontology-based discovery and retrieval of geographic information. in *7th Conference on Geographic Information Science (AGILE 2004)*, (2004), http://ifgi.uni-muenster.de/~lutzm/agile04_klien_et_al.pdf.

67. Kolbe, T.H. Bridging the Gap between GIS and CAAD. *GIM International* (July 2004). 12-15, 2004.

68. Kolbe, T.H. and Groger, G., Towards unified 3D city models. in *ISPRS Comm. IV Joint Workshop on Challenges in Geospatial Analysis, Integration and Visualization II*, (Stuttgart, 2003), http://www.iuw.uni-vechta.de/personal/geoinf/jochen/papers/28.pdf.

69. Kremer, R., Constraint Graphs: A Concept Map Meta-Language, Ph.D. Thesis, University of Calgary, 1997.

70. Kuhn, W., Formalizing Spatial Affordances, Formal Models of Commonsense Worlds, 1996, http://www.geog.buffalo.edu/ncgia/i21/papers/kuhn.html.

71. Kuhn, W. Ontologies in support of activities in geographical space. *International Journal of Geographical Information Science*, *15* (7). 613-631, 2001.

72. Kuhn, W. and Raubal, M., Implementing semantic reference systems. in *6th AGILE Conference on Geographic Information Science 2003*, (Lyon, France, 2003), 63-72.

73. Laudon, K.C. and Laudon, J.P. *Management information systems: organization and technology*. Prentice Hall, Upper Saddle River, N.J., 1996.

74. Lo, C.P. and Yeung, A.K.W. *Concepts and techniques of geographic information systems*. Prentice Hall, Upper Saddle River, N.J., 2002.

75. Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W. *Geographic information systems and science*. Wiley, Chichester ; New York, 2001.

76. Lutz, M., Non-taxonomic Relations in Semantic Service Discovery and Composition. in *1st "Ontology in Action" Workshop, in conjunction with 16th Conference on Software Engineering and Knowledge Engineering (SEKE 2004)*, (Banff, Canada, 2004), 482-485.

77. Lutz, M., Riedemann, C. and Probst, F., A classification framework for approaches to achieving semantic interoperability between gi web services. in *Conference on Spatial Information Theory 2003*, (2003), 186-203.

78. Maedche, A.D. *Ontology learning for the semantic Web*. Kluwer Academic Publishers, Boston, 2002.

79. Mark, D.M. Human spatial cognition. in Medyckyj-Scott, D. and Hearnshaw, H.M. eds. *Human Factors in Geographical Information Systems*, Belhaven Press, London, 1993, 51-60.

80. Mark, D.M., Freksa, C., Hirtle, S., Lloyd, R. and Tversky, B. Cognitive models of geographical space. *International Journal of Geographical Information Science*, *13* (8). 747-774, 1999.

81. Mark, D.M., Smith, B., Egenhofer, M. and Hirtle, S. Ontological Foudations for Geographic Information Science. in Usery, R.B.M.a.L. ed. *Research Challenges in Geographic Information Science*, John Wiley & Sons, New York, 2003.

82. Mark, D.M., Smith, B. and Tversky, B. Ontology and geographic objects: an emperical study of cognitive categorization. in Freksa, C. and Mark, D.M. eds. *Spatial Information Theory - A Theoretical Basis for GIS. Lecture Notes in Computer Science 1661*, Berlin-Heidelberg, Springer-Verlag, 1999, 283-298.

83. Marr, D. *Vision: a computational investigation into the human representation and processing of visual information*. W.H. Freeman, San Francisco, 1982.

84. Merriam-Webster, Online dictionary, 2004, http://m-w.com.

85. Mitchell, W.B., Guptill, S.C., Anderson, E.A., Fegeas, R.G. and Hallam, C.A., GIRAS - a Geographic Information Retrieval and Analysis System for handling land use and land cover data, professional paper 1059, U.S. Geological Survey, 1977.

86. Morehouse, S., The architecture of ARC/INFO. in *Auto-Carto 9 Proceedings*, (Fall Church, VA, 1989), American Society of Photogrammetry and Remote Sensing., 266-277.

87. National Research Council (U.S.). Committee on Intersections Between Geospatial Information and Information Technology. *IT roadmap to a Geospatial Future*. National Academies Press, Washington, D.C., 2003.

88. NIST, Role Based Access Control, 2004, http://csrc.nist.gov/rbac/.

89. Norman, D.A. *The design of everyday things*. Basic Books, New York, 2002.

90. Observer, BMW Computer Makes Driver Turn into Drinker, 1998.

91. OCLC, Dewey decimal classification, 2002, http://www.oclc.org/dewey/.

92. OGC, Open Geospatial Consortium, 2004, http://www.opengeospatial.org.

93. OGC, Open GIS Abstract Specification Overview, 2003, http://www.opengeospatial.org/specs/?page=abstract.

94. OGC, Open GIS Catalog services specification, 2002, http://portal.opengis.org/files/?artifact_id=5929&version=1.

95. OGC, Open GIS Implementation Specification: coordinate transformation services, revision 1.0, 2001, http://www.opengis.org/docs/01-009.pdf.

96. OGC, Open GIS Implementation Specifications Overview, 2003, http://www.opengeospatial.org/specs/?page=specs.

97. Ogden, C.K. and Richards, I.A. *The meaning of meaning. a study of the influence of language upon thought and of the science of symbolism. supplementary essays by B. Malinowski and F.G. Crookshank*. Harcourt, New York,, 1923.

98. Ozemoy, V.M., Smith, D.R. and Sicherman, A. Evaluating computerized geographic information systems using decision analysis. *Interfacees*, *11*. 92-98, 1981.

99. Peachavanish, R. and Karimi, H.A., GeoInterpret: Ontological Engineering Framework for Geospatial Query Analysis. in *GIScience 2004*, (Adelphi, MD, 2004), 168-171.

100. Probst, F. and Lutz, M., Giving Meaning to GI Web Service Descriptions (Extended Abstract). in *7th Conference on Geographic Information Science (AGILE 2004)*, (Heraklion, Greece, 2004), http://ifgi.uni-muenster.de/~lutzm/agile04_probst_lutz.pdf.

101. Protege, The Protégé Ontology Editor and Knowledge Acquisition System, 2004, http://protege.stanford.edu/.

102. Raubal, M. and Kuhn, W. Ontology-Based Task Simulation. *Spatial Cognition and Computation*, *4* (1). 15-37, 2004.

103. Rigaux, P., Scholl, M.O. and Voisard, A. *Spatial databases: with application to GIS*. Morgan Kaufmann Publishers, San Francisco, 2002.

104. Sheth, A. Changing focus on interoperability in information systems: from system, syntax, structure to semantics. in Goodchild, M.F., Egenhofer, M.J., Fegeas, R.G. and Kottman, C.A. eds. *Interoperating Geographic Information Systems*, Kluwer, 1999, 5-30.

105. Smith, B., Fiat objects. in *ECAI 94 Workshop on Parts and Wholes: Conceptual Part-Of Relations and Formal Mereology*, (1994), 15-23, http://ontology.buffalo.edu/smith//articles/fiat1994.pdf.

106. Smith, B. and Mark, D.M. Geographical categories: an ontological investigation. *International Journal of Geographical Information Science*, *15* (7). 591-612, 2001.

107. Smith, B. and Mark, D.M., Ontology and geographic kinds. in *International Symposium on Spatial Data Handling (SDH'98)*, (Vancouver, Canada, 12-15 July, 1998, 1998).

108. Smith, B. and Varzi, A.C. Fiat and bona fide boundaries. *Philosophy and Phenomenological Research*, *60* (2). 401-420, 2000.

109. Smith, B. and Welty, C., Ontology: towards a new synthesis. in *The Second International Conference on Formal Ontology in Information Systems*, (Ogunquit, Maine, USA, 2001).

110. Smith, B. and Welty, C., Ontology: towards a new synthesis. in *The Second International Conference on Formal Ontology in Information Systems*, (Ogunquit, Maine, USA, 2001), iii-x.

111. Smith, T.R., Menon, S., Starr, J.L. and Estes, J.E. Requirements and principles of the implementation and construction of large-scale geographic information systems. *International Journal of Geographical Information Systems*, *1*. 13-31, 1987.

112. Snyder, J.P., Map projections - a working manual. U.S. Geological Survey professional paper 1395, United States Government Printing Office., 1987.

113. Sorrows, M.E. and Hirtle, S.C. The nature of landmarks for real and electronic spaces. in Freksa, C. and Mark, D.M. eds. *Spatial Information Theory*, Springer-Verlag., Heidelberg, 1999, 37-50.

114. Sowa, J.F., Conceptual Graph - proposed draft standard, 2001, http://users.bestweb.net/~sowa/cg/cgstand.htm.

115. Sowa, J.F. *Knowledge representation : logical, philosophical, and computational foundations*. Brooks/Cole, Pacific Grove, CA, 2000.

116. Steimann, F. On the Representation of Roles in Object-Oriented and Conceptual Modelling. *Data & Knowledge Engineering*, *35* (1). 83-106, 2000.

117. Steimann, F. Roles = Interfaces: A Merger of Concepts. *Journal of Object-Oriented Programming* (October-November 2001). 23-32, 2001.

118. Tanaka, M. and Ichikawa, T. A visual user interface for map informatino retrieval based on semantic significance. *IEEE Transactions on Software Engineering*, *14* (5). 666-670, 1988.

119. Theobald, D.M. Understanding topology and shapefiles. *ArcUser*, 2001, http://www.esri.com/news/arcuser/0401/topo.html.

120. Timpf, S. and Frank, A. Using hierarchical spatial data structures for hierarchical spatial reasoning. in Hirtle, S.a.F., A. ed. *Spatial Information Theory - A Theoretical Basis for GIS (International Conference COSIT'97). Lecture Notes in Computer Science 1329*, Berlin-Heidelberg, Springer-Verlag, 1997, 69-83.

121. Tomlinson, R. The Canada Geographic Information System. in Foresman, T.W. ed. *History of Geographic Information Systems*, Prentice-Hall, 1998, 21-32.

122. Turk, A. The relevance of human factors to geographical information systems. in Medyckyj-Scott, D. and Hearnshaw, H.M. eds. *Human Factors in Geographical Information Systems*, Belhaven Press, London, 1993, 15-31.

123. Uschold, M. and Jasper, R., A Framework for Understanding and Classifying Ontology Applications. in *IJCAI'99 Workshop on Ontology and Problem Solving Methods: Lesson Learned and Future Trends*, (Stockholm, Sweden, 1999), 18:11.11-11.12.

124. USGS, "Geographic Information Systems" - an information brochure published by the United States Geological Survey, 1997, http://info.er.usgs.gov/research/gis/title.html.

125. USGS, USGS Digital Elevation Model guide, 2003, http://edc.usgs.gov/guides/dem.html.

126. USGS, What is SDTS?, 2003, http://mcmcweb.er.usgs.gov/sdts/whatsdts.html.

127. Verbyla, D.L. *Practical GIS analysis*. Taylor & Francis, London ; New York, 2002.

128. Visser, U., Stuckenschmidt, H., Schuster, G. and Vogele, T. Ontologies for geographic information processing. *Computers & Geosciences*, *28*. 103-117, 2002.

129. W3C, Semantic Web, 2004, http://www.w3.org/2001/sw/.

130. Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hubner, S., Ontology-based integration of information - a survey of existing approaches. in *IJCAI-01 Workshop: Ontologies and Information Sharing*, (2001), 108-117.

131. Wiederhold, G., Interoperation, mediation, and ontologies. in *Proceedings International Symposium on Fifth Generation Computer Systems (FGCS94), Workshop on Heterogeneous Cooperative Knowledge-Bases*, (Tokyo, Japan, 1994), 33-48.