

LỜI CẢM ƠN

Đầu tiên chúng em xin chân thành cảm ơn Thầy Trần Cao Đệ, người đã chỉ bảo chúng em, giúp chúng em giải quyết những khó khăn vấp phải khi thực hiện luận văn. Thầy đã cung cấp cho chúng em những kiến thức mới giúp chúng em có thể nhanh chóng hoàn thành luận văn này.

Trong thời gian học tập ở trường quý Thầy Cô khoa Công nghệ Thông Tin và Truyền Thông đã tận tình dạy bảo cho chúng em những kiến thức bổ ích, giúp chúng em vượt qua những kỳ thi, những khó khăn trong quá trình học tập, cũng như là đã chỉ bảo chúng em trong suốt thời gian làm luận văn này. Chúng em xin chân thành cảm ơn quý Thầy Cô khoa Công nghệ Thông Tin và Truyền Thông, trường Đại học Cần Thơ. Kính chúc quý Thầy Cô dồi dào sức khỏe và thành công trong công việc của mình.

Cảm ơn các bạn lớp Tin Học 4 - Khóa 33 các bạn đã trợ giúp, trao đổi với chúng tôi về ngôn ngữ, giải thuật, giúp chúng tôi có nhiều kinh nghiệm hơn trong quá trình thực hiện luận văn này. Chúc các bạn thành công trên con đường học tập của mình.

Cuối cùng chúng con xin tỏ lòng biết ơn chân thành đến ba mẹ, người đã tạo mọi điều kiện tốt nhất cho việc học của chúng con và động lực để giúp chúng con vượt qua những khó khăn trong thời gian qua.

Cần Thơ, ngày 16 tháng 05 năm 2011

Nhóm sinh viên thực hiện

Trần Văn Hoàng – Lương Minh Liêm Pha

MỤC LỤC

LỜI CẢM ƠN	i
TÓM TẮT	v
ABSTRACT	vi
KÝ HIỆU, THUẬT NGỮ VÀ VIẾT TẮT	vii
DANH MỤC HÌNH	viii
DANH MỤC BẢNG	x
Chương 1 - TỔNG QUAN	11
1.1 ĐẶT VÂN ĐỀ	11
1.2 PHẠM VI CỦA ĐỀ TÀI.....	12
1.3 PHƯƠNG PHÁP VÀ PHẠM VI NGHIÊN CỨU.....	12
Chương 2 - BẢN ĐỒ VÀ XÂY DỰNG BẢN ĐỒ	14
2.1 TỔNG QUAN VỀ GIS	14
2.1.1 Bản đồ địa lý	14
2.1.2 Hình dạng và kích thước của trái đất.....	15
2.1.3 Hệ tọa độ địa lý	15
2.1.4 Phép chiếu bản đồ.....	17
2.1.5 Phép chiếu UTM (Universal Transverse Mercator).....	19
2.1.6 Tỷ lệ bản đồ.....	20
2.1.7 Hệ thống thông tin địa lý	20
2.2 MÔ HÌNH DỮ LIỆU CỦA HỆ THỐNG THÔNG TIN ĐỊA LÝ	21
2.2.1 Mô hình dữ liệu	21
2.2.2 Thu thập dữ liệu	23
2.3 XÂY DỰNG BẢN ĐỒ SỐ	25
2.3.1 Số hóa bản đồ	25
2.3.2 Qui trình thành lập bản đồ số từ bản đồ giấy.....	25
2.3.3 Sơ đồ quy trình	26
2.3.4 Giới thiệu MapInfo	30
Chương 3 - CƠ SỞ DỮ LIỆU KHÔNG GIAN VÀ WEBGIS.....	31
3.1 CƠ SỞ DỮ LIỆU KHÔNG GIAN	31

3.1.1	Cấu trúc cơ sở dữ liệu không gian.....	31
3.1.2	Cách thức lưu trữ - Quan hệ không gian topology.....	36
3.1.3	Hệ cơ sở dữ liệu không gian	37
3.1.4	Giới thiệu PostgreSQL (PostGIS)	38
3.2	CÔNG NGHỆ WEBGIS	42
3.2.1	Giới thiệu về WebGIS	42
3.2.2	Phân loại WebGIS	44
3.2.3	Mô hình xử lý và kiến trúc triển khai WebGIS.....	47
3.2.4	Giới thiệu GeoServer	51
3.2.5	Thư viện Openlayers.....	57
3.3	KIẾN TRÚC HƯỚNG DỊCH VỤ VÀ DỊCH VỤ WEB	60
3.3.1	Kiến trúc hướng dịch vụ	60
3.3.2	Dịch vụ web	61
3.4	HIỆN TRẠNG ỨNG DỤNG CÔNG NGHỆ GIS TẠI VIỆT NAM.....	65
Chương 4 - CHUẨN OPENGIS	67	
4.1	TỔ CHỨC OPENGIS OGC	67
4.2	TỔNG QUAN VỀ OPENGIS	68
4.2.1	Các khái niệm.....	68
4.2.2	Đặc tả OpenGIS (OpenGIS Specification).....	68
4.3	WEB MAP SERVICE (WMS).....	74
4.4	WEB FEATURE SERVICE (WFS)	77
4.5	WEB COVERAGE SERVICE (WCS)	77
Chương 5 - XÂY DỰNG HỆ THỐNG WEBGIS	80	
5.1	ĐẶC TẢ YÊU YÊU CẦU HỆ THỐNG.....	80
5.1.1	Hiển thị bản đồ với các thông tin chi tiết.....	80
5.1.2	Phóng to, thu nhỏ bản đồ	80
5.1.3	Dịch chuyển bản đồ	81
5.1.4	Xem thông tin các địa điểm trên bản đồ	81
5.1.5	Tìm và hiển thị đường đi ngắn nhất giữa hai điểm	81
5.1.6	Tìm và hiển thị các địa điểm xung quanh vị trí được chọn	82
5.1.7	Tìm kiếm và hiển thị các địa điểm theo vùng.....	83
5.1.8	Tìm và hiển thị địa điểm theo chủ đề	83

5.1.9	Tìm và hiển thị địa điểm trên lộ trình.....	84
5.1.10	Đo khoảng cách đường chim bay giữa hai điểm bất kỳ	84
5.1.11	Tìm và hiển thị các con đường theo tên	84
5.2	MÔ HÌNH TỔNG THỂ CỦA HỆ THỐNG.....	85
5.3	PHƯƠNG PHÁP LÀM VIỆC VÀ PHÂN CHIA CÔNG VIỆC.....	85
5.3.1	Phân chia công việc	85
5.3.2	Sử dụng SVN	86
5.4	XÁC ĐỊNH CÁC GIẢI PHÁP VÀ XÂY DỰNG DỊCH VỤ	87
5.4.1	Xây dựng dữ liệu bản đồ bằng phần mềm MapInfo	87
5.4.2	Chuyển dữ liệu sang HQTCSDL PostgreSQL (PostGIS)	88
5.4.3	Tạo kết nối giữa Geoserver và CSDL PostGIS	99
5.5	BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT GIỮA HAI ĐIỂM.....	100
5.5.1	Giới thiệu	100
5.5.2	Mô hình hệ thống giao thông thành phố thị có hướng	100
5.5.3	Đánh giá	101
5.6	TÌM KIÉM ĐỊA ĐIỂM.....	101
5.7	XÂY DỰNG TẬP DỊCH VỤ TRUY XUẤT BẢN ĐỒ.....	103
5.8	GiẢI PHÁP PHÍA CLIENT	104
5.9	THIẾT KẾ GIAO DIỆN HỆ THỐNG	106
5.10	GIỚI THIỆU GIAO DIỆN HỆ THỐNG	107
Chương 6 - KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN.....	112	
6.1	KẾT QUẢ ĐẠT ĐƯỢC	112
6.2	HƯỚNG PHÁT TRIỂN	113
TÀI LIỆU THAM KHẢO	115	
PHỤ LỤC A - STYLED LAYER DESCRIPTOR IMPLEMENTATION SPECIFICATION	117	
PHỤ LỤC B - OPENGIS SIMPLE FEATURES SPECIFICATION FOR SQL ...	130	
PHỤ LỤC C - GIAO DIỆN QUẢN LÝ THÔNG TIN VỚI GEOSERVER	143	
PHỤ LỤC D - MỘT SỐ THỦ TỤC TRONG POSTGIS HỖ TRỢ XÂY DỰNG DỊCH VỤ	144	

TÓM TẮT

Đề tài này tập trung vào việc nghiên cứu và thiết kế hệ thống thông tin địa lý (GIS) với công nghệ MapInfo, WebGIS cho phép thể hiện bản đồ giao thông và các địa điểm quan trọng (cơ quan, trường học, bệnh viện...) trong thành phố Cần Thơ. Ứng dụng được xây dựng trên công nghệ WebGIS theo chuẩn mở OpenGIS thực hiện các chức năng hiển thị, tìm kiếm trên bản đồ số. Ứng dụng được phát triển trên nền tảng phần mềm mã nguồn mở GeoServer, trong đó phía máy chủ cung cấp dịch vụ bản đồ, và thư viện OpenLayers phía máy khách để hiển thị bản đồ.

Bên cạnh đó, đề tài cũng tìm hiểu các công nghệ SOA, Webservices, cơ sở dữ liệu không gian PostgreSQL (PostGIS) và các công nghệ có liên quan nhằm hỗ trợ việc khai thác dịch vụ theo chuẩn OpenGIS.

Kết quả thực hiện đề tài là hệ thống WebGIS hỗ trợ các chức năng tìm kiếm, hiển thị các địa điểm quan trọng và đường đi ngắn nhất trên bản đồ thành phố Cần Thơ từ những dịch vụ đã xây dựng.

Từ khóa: WebGIS, OpenGIS, hệ thống thông tin địa lí (GIS), bản đồ số giao thông.

ABSTRACT

This thesis focuses on researching and designing GIS information system using MapInfo and WebGIS technology to display route map and some interesting places of Can Tho City such as government offices, schools, hospitals... The application is built on WebGIS technology committed to the OpenGIS standard that provides displaying and searching functions on digital map. The application is developed on the open source software GeoServer which provides map services, and Openlayers library which supports displaying digital map on client side.

Furthermore, this thesis also reserves to research on technology, including: SOA, Web Services, spatial database PostgreSQL (PostGIS) and related technologies to use primary web services based on OpenGIS standard.

The outcome of the thesis is a WebGIS system supporting main functions such as: displaying interesting places and finding shortest path on the map of Can Tho city that have been built as web services.

KÝ HIỆU, THUẬT NGỮ VÀ VIẾT TẮT

STT	Ký hiệu, viết tắt	Điễn giải
1	GIS	Geographic Information System
2	OGC	Open GIS Consortium
3	OpenGIS	Open Geodata Interoperability Specification
4	WFS	Web Feature Service
5	WMS	Web Map Service
6	WCS	Web Coverage Service
7	XML	Extensible Markup Language
8	GML	Geography Markup Language
9	SOA	Service Oriented Architecture
10	UDDI	Universal Description, Discovery and Integration
11	SOAP	Simple Object Access Protocol
12	WSDL	Web Services Description Language
13	CSDL	Cơ sở dữ liệu
14	HQTCSDL	Hệ quản trị cơ sở dữ liệu
15	SLD	Styled Layer Descriptor
16	HTTP	Hypertext Transfer Protocol
17	RIA	Rich Internet Application
18	SVG	Scalable Vector Graphics
19	COM	Component Object Model
20	ORDBMS	Object-Relational Database Management System
21	SVN	Subversion
22	Ajax	Asynchronous JavaScript and XML

DANH MỤC HÌNH

Hình 2.1- Biểu thị bề mặt trái đất lên mặt phẳng.....	14
Hình 2.2- Dạng Geoid và hình Ellipsoid	15
Hình 2.3- Hệ tọa độ địa lý	16
Hình 2.4- Mặt chiếu hình nón	18
Hình 2.5- Mặt chiếu hình trụ	19
Hình 2.6- Các vị trí của mặt phẳng phương vị	19
Hình 2.7- Mô hình dữ liệu dạng vector	22
Hình 2.8- Mô hình dữ liệu dạng raster	23
Hình 2.9 - Sơ đồ quy trình số hóa ảnh số	26
Hình 3.1- Biểu diễn thế giới thực bằng các lớp chuyên đề	36
Hình 3.2 - Sơ đồ hoạt động của WebGIS	42
Hình 3.3 - Kiến trúc Client- Server trong WebGIS	43
Hình 3.4 - Kiến trúc WebGIS	50
Hình 3.5 – các định dạng dữ liệu được GeoServer hỗ trợ	53
Hình 3.6- Kết quả định dạng cho lớp kiểu điểm bằng SLD	56
Hình 3.7- Sự cộng tác trong một kiến trúc hướng dịch vụ.....	61
Hình 3.8- Các tác nhân của dịch vụ web	62
Hình 3.9- Quá trình xử lý của SOAP	64
Hình 3.10- Đăng ký và sử dụng các dịch vụ	65
Hình 5.1- Mô hình tổng thể của hệ thống.....	85
Hình 5.2- Bảng phân chia công việc	86
Hình 5.3- Cơ chế hiển thị và chèn lớp dữ liệu.....	105
Hình 5.4- Thiết kế giao diện	106
Hình 5.5- Giao diện chính của hệ thống.....	107
Hình 5.6- Hiển thị thông tin của một địa điểm được chọn	108
Hình 5.7- Tìm địa điểm theo tên	108
Hình 5.8- Chức năng tìm địa điểm theo vùng.....	109
Hình 5.9- Chức năng chọn điểm bắt đầu và kết thúc của đường đi.....	109

Hình 5.10- Chức năng tìm đường đi	110
Hình 5.11- Đo khoảng cách đường chim bay	110
Hình 5.12- Chức năng tìm kiếm con đường	111

DANH MỤC BẢNG

Bảng 4.1- Bảng mô tả các tham số trong GetCapabilities request URL.....	75
Bảng 4.2- Bảng mô tả các yêu cầu của GetMap	75
Bảng 4.3- Bảng mô tả các yêu cầu của GetFeatureInfo.....	76
Bảng 5.1- Bảng ‘truong’: trường học	89
Bảng 5.2- Bảng ‘coquan’: các cơ quan hành chính	89
Bảng 5.3- Bảng ‘benhvien’: các bệnh viện, trạm y tế.....	90
Bảng 5.4- Bảng ‘ben’: các bến tàu, bến xe, bến đò.....	90
Bảng 5.5- Bảng ‘buudien’: các bưu điện.....	91
Bảng 5.6- Bảng ‘cau’: các cầu.	91
Bảng 5.7- Bảng ‘cho’: các siêu thị, chợ, cửa hàng	92
Bảng 5.8- Bảng ‘congty’: các công ty, nhà máy.....	92
Bảng 5.9- Bảng ‘congvien’: các công viên.....	93
Bảng 5.10- Bảng ‘denchua’: các đền, chùa, miếu, nhà thờ	93
Bảng 5.11- Bảng ‘giaitri’: các khu vui chơi, giải trí, quán cafe	94
Bảng 5.12- Bảng ‘khachsan’: các khách sạn, nhà nghỉ.....	94
Bảng 5.13- Bảng ‘nganhang’: các ngân hàng, máy ATM	95
Bảng 5.14- Bảng ‘thuvien’: các thư viện, nhà sách	95
Bảng 5.15- Bảng ‘giaothong’: các đường đi.....	96
Bảng 5.16- Bảng ‘dinh’: các điểm giao giữa các đường đi và các đầu mút.....	96
Bảng 5.17- Bảng ‘quanhuyen’: địa giới hành chính của các quận huyện	97
Bảng 5.18- Bảng ‘xaphuong’: địa giới hành chính của các xã phường	97
Bảng 5.19- Bảng ‘geometry_columns’: lưu thông tin các bảng dữ liệu không gian	98
Bảng 5.20- Bảng ‘spatial_ref_sys’: lưu thông tin các hệ qui chiếu không gian.....	98

Chương 1 - TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ

Hệ thống thông tin địa lý (GIS) đã phát triển phổ biến vào những năm 80 trên thế giới. Các hệ thống này cung cấp các công cụ cho phép tạo lập bản đồ cung cấp các thông tin địa lý, thể hiện các sự kiện, giải quyết các bài toán phức tạp trong thực tế. Với những ưu điểm mà nó mang lại cùng với nhu cầu thực tế, GIS được ứng dụng ngày càng vào nhiều lĩnh vực nghiên cứu và quản lý như tra cứu thông tin địa lý, mô phỏng và dự báo các hiện tượng tự nhiên,....

Ứng dụng công nghệ GIS ngày càng phát triển đa dạng cả về chiều rộng lẫn chiều sâu, đặt ra nhiều thách thức về kỹ thuật trong quá trình thu thập, phân tích và xử lý dữ liệu với khối lượng lớn. Các dữ liệu GIS thường được thu thập từ nhiều nguồn khác nhau và có thể được định dạng theo nhiều chuẩn khác nhau. Do đó, khi khai thác dữ liệu này, người sử dụng dễ gặp phải những vấn đề lớn về tính tương thích, cũng như những khó khăn trong việc mua bán phần mềm, làm quen với việc sử dụng phần mềm, chưa kể đến việc phải bỏ ra một lượng tiền lớn để mua toàn bộ các dữ liệu do nhà cung cấp xây dựng, trong khi hầu hết các trường hợp ta chỉ cần một phần thông tin trong đó mà thôi. Hơn nữa, các dữ liệu này cần được lấy về và lưu trữ tập trung ở một nơi, lại phải bỏ ra một chi phí không nhỏ khác cho việc lưu trữ, bảo trì và cập nhật chúng. Điều này rõ ràng là lãng phí công sức, thời gian và tiền bạc một cách vô ích. Để khắc phục những trở ngại này, việc tiến hành xây dựng các dịch vụ web hỗ trợ GIS (WebGIS) là một giải pháp tốt hiện đang được rất nhiều nơi trên thế giới triển khai thực hiện. Thay vì dồn các dữ liệu lại một nơi và xử lý tập trung trên đó, giải pháp dịch vụ web lại đi theo con đường xử lý phân tán. Mọi thông tin yêu cầu và đáp ứng đều được gửi và nhận thông qua Internet.

Hiện nay, tại nước ta công nghệ GIS không phải là một công nghệ mới. Nhưng chỉ có một số ít viện nghiên cứu, các cơ quan và vài công ty là có nghiên cứu và sử dụng GIS. Về WebGIS thì số lượng người nghiên cứu còn ít hơn.

Bên cạnh đó chính sự phát triển nhanh và rộng khắp của các hệ thống WebGIS đã làm nảy sinh nhu cầu xây dựng một chuẩn chung áp dụng trên toàn cầu để tạo ra sự giao tiếp dễ dàng cho các ứng dụng WebGIS. Nhận thấy được điều đó, tổ chức OGC (Open Geospatial Consortium) đã đưa ra chuẩn chung thống nhất cho các dịch vụ WebGIS.

Do đó, mục tiêu của đề tài là xây dựng hệ thống WebGIS cho phép hiển thị bản đồ dựa trên dữ liệu địa lý và phi địa lý, khai thác và sử dụng các dịch vụ theo chuẩn OpenGIS. Hỗ trợ các chức năng cơ bản của một bản đồ số, các tiện ích tìm đường đi và địa điểm trên nền web cho phép người dùng xem, tương tác và tìm kiếm các thông tin từ dịch vụ đã xây dựng.

1.2 PHẠM VI CỦA ĐỀ TÀI

- Sử dụng phần mềm MapInfo để xây dựng dữ liệu bản đồ cho hệ thống WebGIS.
- Nghiên cứu các công nghệ WebGIS theo chuẩn mở để kết nối và chia sẻ tài nguyên.
- Tìm hiểu tổng quan về chuẩn OpenGIS để khai thác sử dụng các dịch vụ hỗ trợ cho việc xây dựng hệ thống.
- Xây dựng hệ thống WebGIS, hiển thị mạng giao thông, các cơ quan hành chính, các địa điểm quan trọng... ở thành phố Cần Thơ.
- Tìm hiểu về Webservice để xây dựng các dịch vụ hỗ trợ tìm đường đi và địa điểm.

1.3 PHƯƠNG PHÁP VÀ PHẠM VI NGHIÊN CỨU

Lý thuyết: Để thực hiện đề tài cần nghiên cứu tổng quan về hệ thống thông tin địa lý, nghiên cứu các khái niệm liên quan đến WebGIS để hiểu cách thức cũng như các quy tắc xây dựng GIS trên nền web. Tìm hiểu phần mềm MapInfo để xây dựng dữ liệu bản đồ cho thành phố Cần Thơ. Bên cạnh đó, cũng cần nghiên cứu các hệ quản trị cơ sở dữ liệu không gian để có cái nhìn tổng quan về cách thức lưu trữ và

xử lý dữ liệu không gian để có sự lựa chọn phù hợp. Tìm hiểu GeoServer, một server cho phép kết nối và chia sẻ dữ liệu không gian. Ngoài ra, cũng cần nghiên cứu SOA và Web Service làm cơ sở cho việc xây dựng, phân tích và thỏa thuận chuẩn giao tiếp chung. Sau cùng là tìm hiểu về chuẩn OpenGIS của tổ chức OGC để có cái nhìn chung về các chuẩn mà tổ chức này cung cấp.

Thực nghiệm: Dựa trên những kiến thức đã nghiên cứu tiến hành so sánh đánh giá các công nghệ đã nghiên cứu để lựa chọn giải pháp phù hợp đi đến xây dựng hệ thống WebGIS đáp ứng các yêu cầu đặt ra.

Bố cục luận văn được chia thành các phần sau đây:

Chương 1 - Tổng quan: Giới thiệu chung về đề tài, mục tiêu của đề tài, phương pháp và phạm vi nghiên cứu.

Chương 2 - Bản đồ và xây dựng bản đồ: Giới thiệu hệ thống thông tin địa lý, gồm những khái niệm về bản đồ địa lý, các hệ tọa độ, hệ qui chiếu. Đề cập đến các công nghệ liên quan đến GIS được sử dụng trong đề tài như: công nghệ số hóa bản đồ dùng MapInfo

Chương 3 - Cơ sở dữ liệu không gian và webgis: Giới thiệu hệ quản trị cơ sở dữ liệu không gian PostgreSQL (PostGIS), công nghệ WebGIS và các công nghệ hỗ trợ để xây dựng một hệ thống WebGIS như: Geoserver, Openlayer, kiến trúc hướng dịch vụ và dịch vụ web.

Chương 4 - Chuẩn opengis: Giới thiệu chuẩn OpenGIS, các đặc tả của chuẩn OpenGIS và các đặc tả được sử dụng trong đề tài tuân theo chuẩn OpenGIS.

Chương 5 - Xây dựng hệ thống webgis: Bao gồm các bước phân tích, thiết kế xây dựng hệ thống WebGIS phục vụ tìm kiếm các địa điểm quan trọng như cơ quan, trường học, bệnh viện... và đường đi ngắn nhất ở thành phố Cần Thơ.

Chương 6 - Kết quả đạt được và hướng phát triển: Tóm tắt các vấn đề đã đặt ra trong luận văn, cách giải quyết, những cái làm được những cái không làm được và đề ra một số hướng phát triển trong tương lai.

Chương 2 - BẢN ĐỒ VÀ XÂY DỰNG BẢN ĐỒ

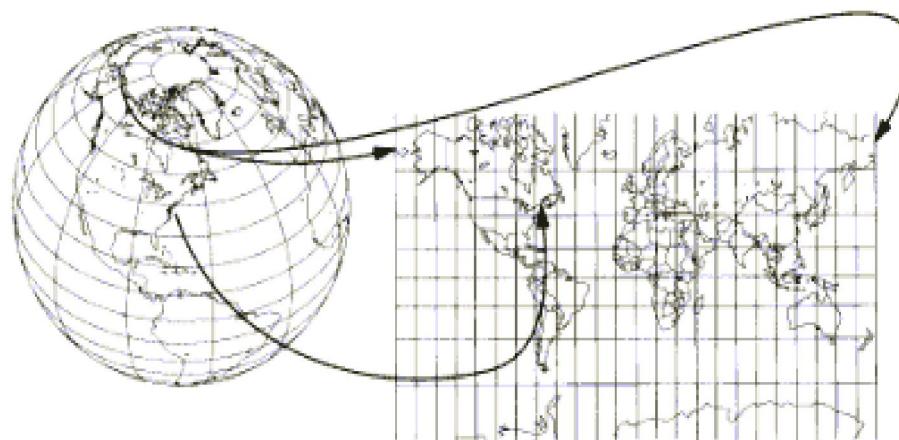
2.1 TỔNG QUAN VỀ GIS

2.1.1 Bản đồ địa lý

Thế giới thực rất rộng lớn và phức tạp để chúng ta có thể thấy bao quát được. Nếu một phần không gian được chọn để trình bày dưới một tỷ lệ nhỏ hơn thực tế thì chúng ta có thể thấy và nhận biết được phần không gian đó dễ dàng hơn nhiều. Và từ đó có thể hiểu rõ khu vực nghiên cứu và đưa ra quyết định chính xác (như việc tìm đường đi, tìm kiếm địa điểm, việc qui hoạch một tuyến đường).

Bản đồ địa lý là sự thu nhỏ của các thực thể và hiện tượng trên bề mặt trái đất lên mặt phẳng được xây dựng dựa trên cơ sở toán học và sử dụng các ký hiệu được qui ước để phản ánh sự phân bố, trạng thái và mối quan hệ tương quan của các hiện tượng tự nhiên và xã hội. Bản đồ chứa thông tin về vị trí, các tính chất của vật thể và các hiện tượng mà nó trình bày.

Thực chất bản đồ là một hệ thống thông tin về không gian. Chúng ta có thể tìm thấy các thông tin trên bản đồ mà người vẽ muốn truyền tải, ví dụ như bản đồ dân số, bản đồ địa chất, bản đồ địa hình...

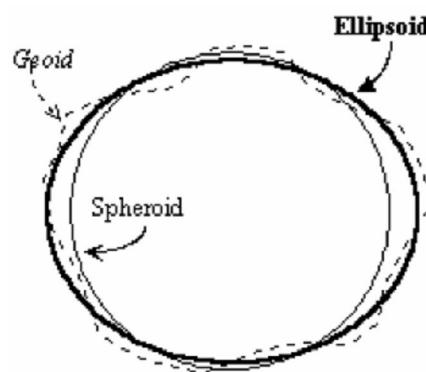


Hình 2.1- Biểu thị bề mặt trái đất lên mặt phẳng

2.1.2 Hình dạng và kích thước của trái đất

Bề mặt tự nhiên của trái đất rất phức tạp về mặt hình học và không thể biểu thị nó bởi một qui luật xác định, hình dạng trái đất được hình thành và bị chi phối bởi hai lực là lực hấp dẫn và lực ly tâm tạo nên hình dạng ellipsoid của trái đất. Trong trắc địa người ta dùng mặt geoid, bề mặt này được tạo bởi mặt nước biển trung bình yên tĩnh kéo dài qua các lục địa và hải đảo tạo thành một mặt cong khép kín. Do tác dụng của trọng lực, sự phân bố không đồng đều của vật chất có tỷ trọng khác nhau trong lớp vỏ trái đất làm cho bề mặt geoid bị biến đổi phức tạp về mặt hình học.

Cho nên, bề mặt hoàn chỉnh của trái đất không phải là bề mặt đúng toán học, mà chỉ là bề mặt sẵn có của chính trái đất. Trong khoa học trắc địa bản đồ, để tiện lợi cho các bài toán đo đạc, người ta lấy mặt ellipsoid tròn xoay có hình dạng và kích thước gần giống mặt geoid thay cho bề mặt của trái đất.



Hình 2.2- Dạng Geoid và hình Ellipsoid

2.1.3 Hệ tọa độ địa lý

Hệ tọa độ địa lý sử dụng bề mặt cầu để xác định vị trí của một điểm trên trái đất. Vì đây là hệ tọa độ gắn liền với trực trái đất nên để xác định vị trí của đối tượng người ta chia bề mặt trái đất thành các đường kinh tuyến và vĩ tuyến. Kinh tuyến là các đường cong cách đều nhau chạy qua hai điểm cực Bắc và Nam, vĩ tuyến là các đường tròn song song có tâm nằm trên trực của trái đất. Giao điểm giữa kinh tuyến và vĩ tuyến tạo thành các ô lưới.

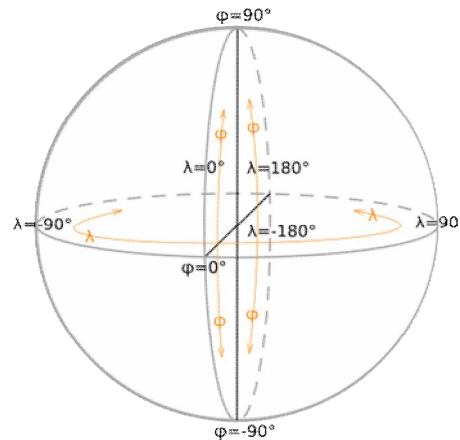
Trong số các kinh tuyế̄n và vĩ tuyế̄n có hai đường quan trọng nhất được lấy làm gốc tọa độ đó là: vĩ tuyế̄n có bán kính lớn nhất (đường xích đạo) và kinh tuyế̄n chạy đi qua Đài Thiên văn Hoàng gia Greenwich của nước Anh là kinh tuyế̄n gốc. Giao điểm giữa hai đường này là gốc tọa độ. Hai đường này cũng đồng thời chia trái đất làm bốn phần bằng nhau: nửa Bắc và Nam nằm phía trên và dưới của đường xích đạo, nửa Đông và Tây nằm phía bên phải và trái của kinh tuyế̄n gốc.

Ngày nay hệ thống tọa độ được sử dụng rộng rãi là hệ thống kinh độ, vĩ độ và độ cao (longitude, latitude, height). Mặt phẳng qua kinh tuyế̄n gốc và đường xích đạo là mặt phẳng quy chiếu, chúng được sử dụng để xác định kinh độ và vĩ độ.

Kinh độ (ký hiệu: λ) của một điểm là góc giữa mặt phẳng quy chiếu và mặt phẳng đi qua điểm mà cả hai mặt phẳng này vuông góc với mặt phẳng xích đạo.

Vĩ độ (ký hiệu: ϕ) của một điểm bất kỳ trên mặt trái đất là góc tạo thành giữa đường thẳng đứng (phương của dây dọi, có đỉnh nằm ở tâm hệ tọa độ – chính là trọng tâm của địa cầu) tại điểm đó và mặt phẳng xích đạo.

Theo quy định, góc của kinh tuyế̄n có giá trị âm khi nằm phía Tây kinh tuyế̄n gốc và có giá trị dương khi nằm ở phía Đông. Góc của vĩ tuyế̄n được xác định tùy thuộc ở Bắc hay Nam, trong đó phía Bắc được coi là dương, phía Nam là âm.



Hình 2.3- Hệ tọa độ địa lý

Trong hệ tọa độ địa lý có hai bề mặt hình cầu được sử dụng đó là mặt cầu (tuyệt đối) và mặt ellipsoid. Vì bề mặt trái đất gần giống với hình ellipsoid nên nó

thường được sử dụng để biểu diễn cho bề mặt trái đất. Tuy nhiên đôi khi người ta cũng sử dụng để công việc tính toán dễ dàng hơn. Khi tỷ lệ bản đồ nhỏ $< 1:5.000.000$, ở tỷ lệ này thì sự khác biệt giữa dữ liệu biểu diễn bằng mặt cầu và mặt ellipsoid là không thể phân biệt bằng mắt thường. Lúc này mặt cầu được dùng. Nhưng khi ở tỷ lệ lớn $> 1:1.000.000$ thì việc sử dụng mặt ellipsoid để đảm bảo độ chính xác là cần thiết. Do đó, việc sử dụng mặt cầu hay mặt ellipsoid tùy thuộc vào mục đích của bản đồ và độ chính xác của dữ liệu.

2.1.4 Phép chiếu bản đồ

Phép chiếu bản đồ là sự chuyển đổi toán học từ hệ trực kinh vĩ độ cầu sang hệ trực tọa độ cực. Nó đảm bảo mối quan hệ đã biết giữa vị trí thực trên bản đồ và vị trí thực trên Trái đất. Hay nói cách khác phép chiếu bản đồ là cách biểu diễn mặt cong của Trái đất lên một mặt phẳng, để mỗi điểm trên mặt cong tương ứng với một điểm trên mặt phẳng. Hiện nay có rất nhiều phép chiếu bản đồ, tuy nhiên có ba phép chiếu cơ bản và thường được sử dụng nhất đó là phép chiếu với mặt chiếu: *mặt hình nón*, *mặt hình trụ* và *mặt phẳng phương vị*. Bước đầu tiên khi tiến hành phép chiếu là tạo ra một hay một tập các điểm tiếp xúc. Các điểm tiếp xúc này gọi là các tiếp điểm. Các điểm này có vai trò rất quan trọng, vì độ biến dạng của phép chiếu trên những điểm này là bằng không. Độ biến dạng sẽ tăng khi khoảng cách giữa điểm chiếu và điểm tiếp xúc tăng.

2.1.4.1 Mặt hình nón

Để thực hiện phép chiếu này người ta dùng một mặt hình nón úp lên bề mặt cầu. Đường thẳng tiếp xúc giữa mặt nón và mặt cầu là một vĩ tuyến và được gọi là vĩ tuyến chuẩn. Các đường kinh tuyến sau khi chiếu sẽ thành những đường thẳng đứng, các đường vĩ tuyến sẽ tạo thành những đường tròn.

Sau khi thực hiện phép chiếu, người ta sẽ cắt hình nón dọc theo một kinh tuyến bất kỳ, lúc này ta sẽ được kết quả của phép chiếu trên bề mặt nón. Sự giao nhau giữa những đường thẳng và cung tròn sẽ tạo nên một mặt lưới. Đường thẳng đối diện với mặt cắt được gọi là kinh tuyến trung tâm.

Càng xa vĩ tuyến chuẩn độ biến dạng càng tăng. Do đó để tăng độ chính xác người ta cắt bỏ phần đỉnh của mặt nón hay ta không tiến hành chiếu lên vùng này.



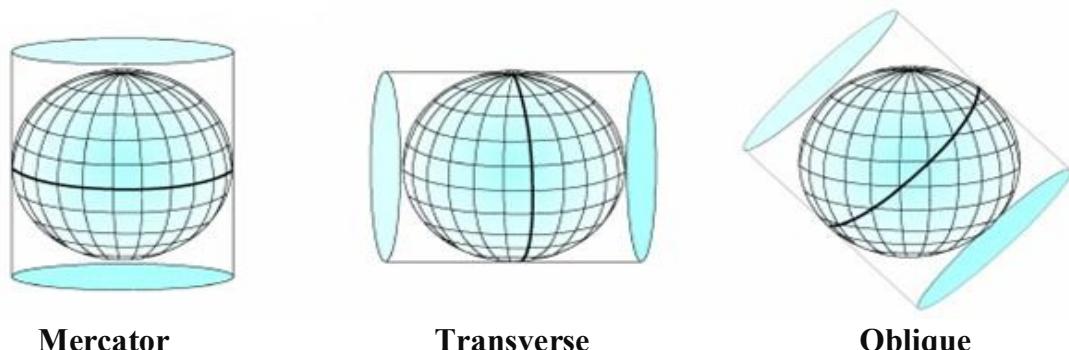
Hình 2.4- Mặt chiếu hình nón

2.1.4.2 **Mặt hình trụ**

Giống như phép chiếu mặt nón, phép chiếu này cũng có một đường thẳng tiếp tuyến. Khi sử dụng mặt trụ người ta phân thành 3 loại tùy thuộc vào vị trí tương đối của mặt trụ so với mặt cầu:

- Hình trụ được đặt theo phương thẳng đứng và tiếp xúc với mặt cầu tại một vĩ tuyến, thường là đường xích đạo. Gọi là phép chiếu Mercator.
- Hình trụ được đặt theo phương nằm ngang, đường thẳng tiếp xúc là một kinh tuyến. Gọi là phép chiếu Transverse.
- Hình trụ đặt nghiêng và tiếp xúc với mặt cầu theo một đường tròn có bán kính lớn nhất (bằng với bán kính đường xích đạo). Gọi là phép chiếu Oblique.

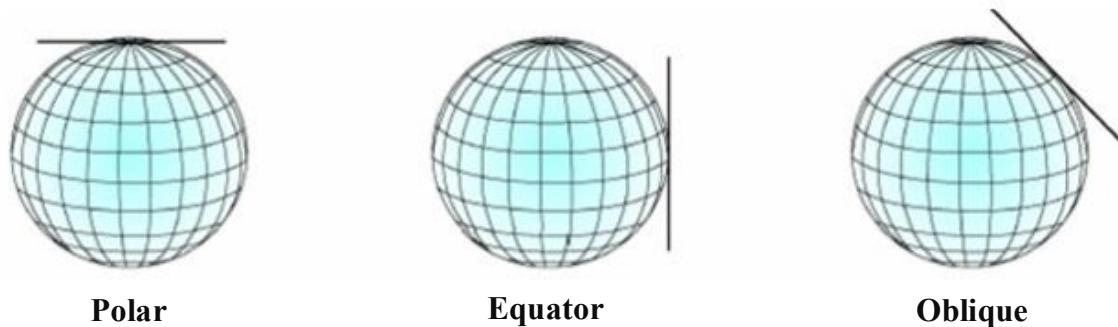
Phép chiếu thường được sử dụng nhất là phép chiếu Mercator. Trong phép chiếu này, đường kinh tuyến sẽ được chiếu thành những đường thẳng đứng cách đều nhau và các đường vĩ tuyến sẽ trở thành các đường nằm ngang khoảng cách không đều nhau và tăng dần về phía hai cực. Do đó biến dạng sẽ tăng dần về phía hai cực. Sau khi thực hiện phép chiếu, người ta sẽ cắt mặt trụ dọc theo một kinh tuyến trãi ra trên mặt phẳng ta sẽ thu được kết quả.



Hình 2.5- Mật chiếu hình trụ

2.1.4.3 Mật phẳng phương vị

Là phép chiếu dữ liệu bản đồ lên một mặt phẳng tiếp xúc với mặt cầu. Điểm tiếp xúc này có thể là: nằm tại hai cực, tại đường xích đạo hoặc tại một vị trí bất kỳ nằm giữa. Vị trí của điểm tiếp xúc cho ta biết vị trí tương đối của mặt phẳng chiếu với mặt cầu và tạo nên ba kiểu chiếu khác nhau: polar, equator và oblique. Mật phẳng chiếu tiếp xúc với cực của mặt cầu là kiểu chiếu đơn giản nhất và cũng hay dùng nhất. Trong phép chiếu này, các đường kinh tuyến sẽ được chiếu thành một chùm đường thẳng giao nhau ở điểm cực, vĩ tuyến là các đường tròn có cùng tâm là cực của mặt cầu. Góc giữa các đường kinh tuyến được bảo tồn [2]



Hình 2.6- Các vị trí của mặt phẳng phương vị

2.1.5 Phép chiếu UTM (Universal Transverse Mercator)

Nguồn gốc là phép chiếu Mercator ngang (Transverse Mercator) và thường được sử dụng để đo và vẽ bản đồ có độ chính xác cao. Phép chiếu UTM chia bìa mặt trái đất thành 60 múi theo vĩ tuyến, mỗi múi 6° và đánh số từ 1 đến 60 kể từ kinh độ $180^{\circ}W$. Mỗi múi kéo dài từ vĩ độ $84^{\circ}S$ đến vĩ độ $80^{\circ}N$. Phép chiếu UTM còn chia

bè mặt trái đất thành các dãi 8° theo kinh tuyến mở ra hai cực từ đường xích đạo. Lãnh thổ Việt Nam nằm trọn vẹn trong múi 48. Trước năm 1945 tại miền Nam Việt Nam, phép chiếu UTM trên cơ sở ellip tròn xoay Everest - 1830 được sử dụng để lập hệ thống bản đồ.

Hiện nay nước ta sử dụng hệ tọa độ quốc gia là VN-2000, trong đó dùng phép chiếu UTM và ellipsoid WGS-84 [2]

2.1.6 Tỷ lệ bản đồ

Tỷ lệ của một bản đồ địa lý là tỷ số giữa một khoảng cách trên bản đồ và khoảng cách ngoài thực địa. Chẳng hạn, nếu 1 cm trên bản đồ ứng với 1 km ngoài thực địa thì bản đồ có tỷ lệ 1:100000.

Kí hiệu của tỷ lệ có dạng 1:M, trong đó số M chỉ khoảng cách thực tế lớn gấp bao nhiêu lần khoảng cách tương ứng đo trên bản đồ. Bản đồ có tỷ lệ lớn thì càng chi tiết và tương ứng với số M nhỏ. Bản đồ tỷ lệ nhỏ kém chi tiết hơn và có số M lớn.

2.1.7 Hệ thống thông tin địa lý

Hệ thống thông tin địa lý (GIS- Geographic Information System) là một hệ thống để tạo lập bản đồ và phân tích các sự vật hiện tượng thật trên trái đất. Có các chức năng xử lý thông tin địa lý, nhằm phục vụ việc qui hoạch, trợ giúp quyết định trong một số lĩnh vực nhất định.

GIS là một công cụ trên cơ sở máy tính để lập bản đồ và phân tích những cái đang tồn tại và các sự kiện xảy ra trên trái đất. Công nghệ GIS tích hợp các thao tác cơ sở dữ liệu như truy vấn và phân tích thống kê với lợi thế quan sát và phân tích thống kê bản đồ. Các khả năng này sẽ phân biệt GIS với các hệ thống khác. Có rất nhiều chương trình máy tính sử dụng dữ liệu không gian như AutoCAD và các chương trình thống kê, nhưng chúng không phải là GIS vì chúng không có khả năng thực hiện các thao tác không gian [1]

Các hệ thống thông tin địa lý nói chung đều bao gồm các thành phần: phần cứng, phần mềm, dữ liệu, và con người.

Phần cứng: Là các máy tính điện tử như PC, mini Computer, MainFrame ... và các thiết bị mạng cần thiết khi triển khai GIS trên môi trường mạng. GIS cũng đòi hỏi các thiết bị ngoại vi đặc biệt cho việc nhập và xuất dữ liệu như: máy số hoá (digitizer), máy vẽ (plotter), máy quét (scanner)...

Phần mềm: phần mềm GIS cung cấp các chức năng và các công cụ cần thiết để lưu giữ, phân tích và hiển thị thông tin địa lý.

Dữ liệu: có thể xem thành phần quan trọng nhất trong hệ thống GIS là dữ liệu. Dữ liệu trong hệ thống GIS thường có hai loại được liên kết với nhau để mô tả về các đối tượng địa lý. Đó là dữ liệu không gian và dữ liệu phi không gian

Con người: những người tham gia vào phát triển và quản lý hệ thống GIS, có thể là các chuyên viên tin học, các chuyên gia về các lĩnh vực khác nhau, chuyên viên GIS, nhà phát triển ứng dụng GIS.

2.2 MÔ HÌNH DỮ LIỆU CỦA HỆ THỐNG THÔNG TIN ĐỊA LÝ

2.2.1 Mô hình dữ liệu

Dữ liệu là trung tâm của hệ thống GIS, hệ thống GIS chứa càng nhiều thì chúng càng có ý nghĩa. Dữ liệu của hệGIS được lưu trữ trong CSDL và chúng được thu thập thông qua các mô hình thế giới thực. Dữ liệu trong hệGIS còn được gọi là thông tin không gian. Đối tượng không gian là có khả năng mô tả “vật thể ở đâu” nhờ vào vị trí tham chiếu, đơn vị đo và quan hệ không gian. Chúng còn có khả năng mô tả “hình dạng hiện tượng” thông qua mô tả chất lượng, số lượng của hình dạng và cấu trúc.

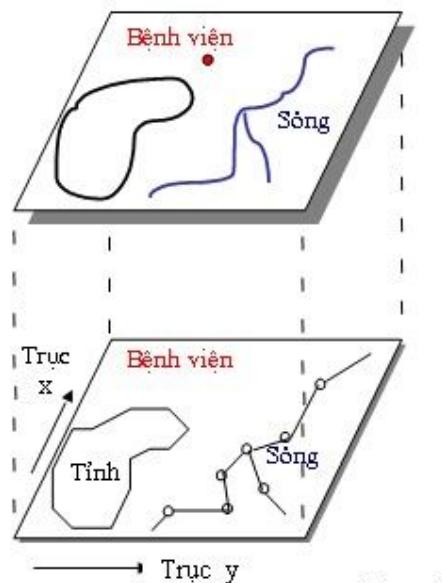
Một cơ sở dữ liệu của hệ thống thông tin địa lý có thể được chia làm hai loại dữ liệu cơ bản: dữ liệu không gian và phi không gian. Mỗi loại có những đặc điểm riêng và có yêu cầu khác nhau về lưu giữ, xử lý và hiển thị.

- Dữ liệu không gian (spatial): là những mô tả số của hình ảnh bản đồ, cho ta biết kích thước vật lý và vị trí địa lý của các đối tượng trên bề mặt trái đất. Hệ thống thông tin địa lý dùng các dữ liệu không gian để tạo ra hình ảnh bản đồ.

- Dữ liệu phi không gian (non-spatial): là các dữ liệu ở dạng văn bản cho ta biết thêm thông tin thuộc tính của đối tượng như tên gọi, tính chất, đặc điểm...

Có hai mô hình cơ bản để biểu diễn thành phần không gian trong GIS: mô hình dữ liệu vector và mô hình dữ liệu raster.

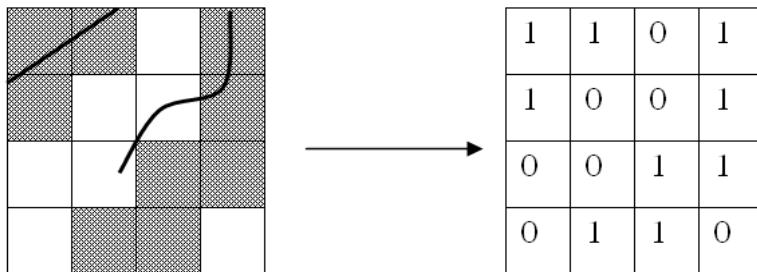
- Trong mô hình dữ liệu vector các đối tượng không gian được tổ chức dưới dạng điểm (point), đường (line) và vùng (polygon), và được biểu diễn trên một hệ thống tọa độ nào đó. Trong mô hình 2D các đối tượng biểu diễn trên mặt phẳng, mỗi đối tượng điểm được biểu diễn bởi một cặp tọa độ (x, y), đối tượng đường được xác định bởi một chuỗi liên tiếp các điểm, đoạn thẳng được nối giữa các điểm hay còn gọi là cạnh (segment). Đối tượng vùng được xác định bởi các đường và khép kín. Trong mô hình 3D đối tượng điểm được biểu diễn bởi tọa độ (x, y, z), thường được áp dụng cho bề mặt ba chiều và khối. Như vậy, mô hình dữ liệu vector sử dụng các đoạn thẳng hay các điểm rời rạc để nhận biết các vị trí của thời giới thực.



Hình 2.7- Mô hình dữ liệu dạng vector

Đối với mô hình dữ liệu raster các đối tượng được định vị trí và lưu trữ dữ liệu địa lý bằng cách sử dụng ma trận, mỗi vị trí được xác định bởi hàng và cột có thuộc tính chính bằng giá trị của ô đó. Khi đó, điểm có thể được biểu diễn bằng một

ô. Đường được biểu diễn bởi một tập các ô có hướng xác định, độ rộng của đường bằng độ rộng của một ô. Còn vùng được biểu diễn bởi một dãy các ô nằm kề sát nhau. Dữ liệu raster gắn liền với dữ liệu dạng ảnh, mô hình dữ liệu này phù hợp trong biểu diễn dữ liệu biến đổi liên tục: độ cao, nhiệt độ...



Hình 2.8- Mô hình dữ liệu dạng raster

Khi so sánh giữa hai mô hình, nhận thấy rằng mô hình dữ liệu vector cho phép nhiều thao tác hơn trên các đối tượng so với mô hình raster. Việc đo khoảng cách, diện tích của các đối tượng được thực hiện bằng các tính toán hình học từ các tọa độ của các đối tượng hình học thay vì việc đếm các tế bào như trong mô hình raster. Do đó, các thao tác như tìm đường đi trong hệ thống giao thông hay hệ thống thủy lợi... được thực hiện nhanh và chính xác hơn trong mô hình dữ liệu vector. Tuy nhiên, đối với một số thao tác như nạp chồng các lớp hay các thao tác vùng đệm thì mô hình raster vượt trội hơn so với mô hình vector.

Thế giới thực có thể được biểu diễn ở cả hai dạng là vector và raster, sự lựa chọn mô hình vector hay raster làm cơ sở tuỳ thuộc vào mục đích và yêu cầu của hệ thống.

2.2.2 Thu thập dữ liệu

Thu thập dữ liệu là quá trình thu nhận dữ liệu theo khuôn mẫu được áp dụng cho GIS. Mức độ đơn giản nhất của thu thập dữ liệu là chuyển đổi các dữ liệu có sẵn từ bên ngoài. Trong trường hợp này, GIS phải có các tiện ích để hiểu được các dạng dữ liệu chuẩn khác nhau để trao đổi. GIS còn phải có khả năng nhập các ảnh bản đồ. Trong thực tế, nhiều kỹ thuật trắc địa được áp dụng để thu thập dữ liệu thô, bao gồm thu thập dữ liệu về bề mặt trái đất như địa hình, địa chất học và thảm thực

vật nhô trắc địa đo đạc hay ảnh chụp từ vệ tinh, máy bay. Các dữ liệu như kinh tế - xã hội thu thập từ điều tra phỏng vấn hay chuyển đổi từ các bài tư liệu viết. Bản đồ vẽ bằng tay trên giấy phải được số hóa sang dạng raster. Việc sử dụng ảnh vệ tinh hay ảnh chụp từ máy bay được xem là nguồn dữ liệu quan trọng khi nghiên cứu tài nguyên thiên nhiên và đo vẽ bản đồ địa hình.

Nhìn chung, có nhiều kỹ thuật để thu thập thông tin cho các nguồn dữ liệu. Tuy nhiên, một số lớn dữ liệu có thể được chuyển đổi từ bản đồ giấy sang hình thức lưu trữ của bản đồ điện tử. Các phương pháp thường được sử dụng:

Phương pháp quét (scanning): đây là kỹ thuật thông dụng mà lại ít tốn kém, có thể được thực hiện trên các máy tính cá nhân. Máy quét sẽ lưu lại các hình ảnh của bản đồ giấy dưới hình thức số và hiển thị chúng trở lại màn hình. Việc quét hình ảnh từ bản đồ giấy tương đối đơn giản và nhanh chóng, tuy nhiên phương pháp này lại không cung cấp các thuộc tính của đối tượng địa lý như địa chỉ của một bệnh viện hay số điện thoại của một trường học nào đó. Dữ liệu có được dưới dạng này thường dưới dạng raster cho kích thước rất lớn.

Phương pháp số hóa: Kỹ thuật này đòi hỏi phải có các thiết bị chuyên ngành. Bản đồ nền sẽ được trãi trên bề mặt ngang, một con trỏ sẽ xác định tọa độ các điểm tạo nên hình dạng bản đồ. Sau quá trình số hóa, thuộc tính của các đối tượng mới được thêm vào. Phương pháp này đòi hỏi nhiều thời gian và dữ liệu có được từ kỹ thuật này dưới dạng vector.

Phương pháp vector hóa: một vài hệ thống máy tính chuyên nghiệp có thể chuyển đổi dữ liệu raster sang dạng vector. Phương pháp này cho tốc độ nhanh do tính toán tự động nhưng lại kém chính xác hơn so với việc số hóa thủ công.

Tuy nhiên, có một số công cụ phân tích của GIS phụ thuộc chặt chẽ vào các mô hình dữ liệu raster, do đó đòi hỏi quá trình biến đổi mô hình dữ liệu vector sang dữ liệu raster, quá trình này được gọi là raster hóa. Một số công cụ phân tích khác lại làm việc chủ yếu với mô hình vector, nên đòi hỏi quá trình biến đổi ngược từ raster sang vector, hay còn gọi là vector hóa. Raster hóa là quá trình phân tích

đường (line) hay miền (polygon) thành các điểm ảnh (pixel). Ngược lại, vector hoá là quá trình tập hợp các điểm ảnh để tạo thành đường hay miền. Dữ liệu ban đầu của ta thông thường là dưới dạng raster nên nếu dữ liệu không có cấu trúc tốt thì việc nhận dạng mẫu sẽ rất phức tạp.

2.3 XÂY DỰNG BẢN ĐỒ SỐ

2.3.1 Số hóa bản đồ

Số hóa bản đồ là quá trình vẽ lại một bản đồ giấy trên máy tính nhằm tạo một bản vẽ dạng số (digital format) của bản đồ đó. Số hóa là một cách nhập dữ liệu không gian, nó ghi nhận tọa độ địa lý của các đối tượng trên mặt đất, lưu trữ dưới dạng số để có thể xử lý trên máy tính. Vấn đề cần quan tâm trong quá trình số hóa là chất lượng của bản đồ giấy và chọn hệ tọa độ quy chiếu, hai yếu tố này sẽ ảnh hưởng đến độ chính xác của bản đồ sau khi được số hóa.

Có hai phương pháp số hóa bản đồ thường được sử dụng: số hóa với bàn số hóa (digitizer) và số hóa từ ảnh quét qua máy quét (scanner) của bản đồ giấy.

Sau khi đăng ký ảnh quét cần phân tích những thông tin có trên bản đồ và tách chúng thành các lớp. Mỗi lớp bản đồ chứa các thông tin thuộc cùng một nhóm ví dụ như: lớp giao thông, lớp ranh giới quận huyện, lớp cơ quan, lớp bệnh viện....

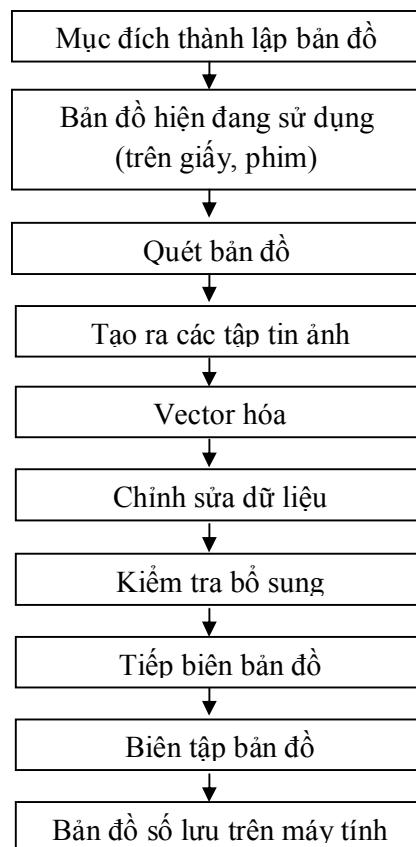
2.3.2 Qui trình thành lập bản đồ số từ bản đồ giấy

Việc thành lập bản đồ số, một trong những bước đi ban đầu trong việc xây dựng cơ sở dữ liệu địa chính là rất cần thiết. Bản đồ số có thể được thành lập từ nhiều nguồn khác nhau: từ ảnh quét scanner, từ ảnh hàng không, ảnh vệ tinh, từ các số liệu đo mặt đất. Sau đây là quá trình thành lập bản đồ số từ bản đồ giấy thông qua ảnh quét scanner

Bản đồ là một chỉnh thể bao gồm nhiều lớp thông tin chồng xếp lên nhau để mô tả thế giới thực. Thông tin trên bản đồ được phân ra thành 4 loại cơ bản sau:

- Đối tượng dạng điểm (point): thể hiện các đối tượng chiếm diện tích nhỏ nhưng là thông tin rất quan trọng không thể thiếu như; trụ sở cơ quan, các công trình xây dựng, cầu cống...
- Đối tượng dạng đường (line): thể hiện các đối tượng không khép kín hình học, chúng có thể là các đường thẳng, các đường gấp khúc và các cung, ví dụ như đường giao thông, sông, suối...
- Đối tượng dạng vùng (region): thể hiện các đối tượng khép kín hình học bao phủ một vùng diện tích nhất định, chúng có thể là các polygon, ellipse và hình chữ nhật, ví dụ lãnh thổ địa giới 1 xã, hồ nước, khu rừng...
- Đối tượng dạng chữ (text): thể hiện các đối tượng không phải là địa lý của bản đồ như nhãn, tiêu đề, ghi chú...

2.3.3 Sơ đồ quy trình



Hình 2.9 - Sơ đồ quy trình số hóa ảnh số

2.3.3.1 Mục đích thành lập bản đồ

Bản đồ đang sử dụng hiện nay phần lớn là bản đồ giấy bao gồm rất nhiều loại khác nhau như: bản đồ địa hình, bản đồ địa chính, bản đồ hiện trạng sử dụng đất, bản đồ đất đai, và các bản đồ chuyên đề khác ... Do đó tùy thuộc vào mục đích cụ thể mà thành lập bản đồ thích hợp. Tuy nhiên, hiện nay Tổng cục địa chính quy định thống nhất dùng bản đồ địa hình và bản đồ địa chính làm bản đồ nền cơ sở trong toàn quốc. Do đó tất cả các bản đồ dù là thành lập với mục đích nào cũng đều được xây dựng trên bản đồ nền cơ sở trên.

Các đối tượng bản đồ khi tồn tại dưới dạng số được thể hiện và lưu trữ trên những lớp thông tin khác nhau. Vì vậy, trước khi số hóa thành lập bản đồ số, các đối tượng cần thể hiện trên bản đồ phải được xác định trước cần phải lưu trữ trên lớp thông tin nào.

2.3.3.2 Quét bản đồ

Quét bản đồ là quá trình chuyển các bản đồ được lưu trữ trên giấy, phim thành các tập tin dữ liệu dưới dạng ảnh (raster file), sau đó tùy thuộc vào phần mềm xử lý ảnh và phần mềm quản lý bản đồ hiện có mà chuyển các raster file sang các định dạng khác như: *.TIFF, *.RLE, *.EPS, *.BMP,...

Một giai đoạn rất quan trọng trong việc thành lập bản đồ số từ bản đồ giấy đó là việc chọn độ phân giải khi quét. Tuy nhiên, việc chọn độ phân giải cao hay thấp còn tùy thuộc vào nhiều yếu tố bao gồm: chất lượng tài liệu gốc, mục đích sử dụng, dung lượng trống của đĩa cứng. Cái giá phải trả cho một raster file có chất lượng cao là kích cỡ raster file đó sẽ lớn gây ra nhiều khó khăn cho việc lưu trữ và chuyển đổi.

2.3.3.3 Các định dạng raster files

Tùy thuộc vào phần mềm xử lý ảnh và phần mềm quản lý bản đồ hiện có mà chuyển thành các định dạng raster files khác nhau. Tuy nhiên, mỗi định dạng khác nhau đều có những thuận lợi và rắc rối riêng của nó. Sau đây là một số định dạng file:

- TIFF (Tagged Image File Format) là dạng phổ biến nhất có khả năng lưu trữ các ảnh quét bằng nhiều độ phân giải, dạng màu và kiểu nén khác nhau, đặc biệt là thích nghi với nhiều trình ứng dụng.
- EPS (Encapsulated Postscript) thích hợp cho dùng các bản vẽ vector nhưng lại không dùng cho lineart.
- GIF là dạng dùng để lưu trữ các ảnh gồm 256 màu hoặc 256 các bóng xám.
- PSP là dạng ảnh nội của Adobe Photoshop.
- JPEG là dạng lưu trữ màu sắc hoặc các files thang độ xám.
- PCX được sử dụng trong nhiều chương trình vẽ khác nhau cũng rất thích hợp cho các ảnh quét và rất thích nghi với cách sử dụng PC (máy tính cá nhân).

2.3.3.4 Nắn bản đồ

Đây là bước quan trọng nhất trong quy trình thành lập bản đồ số vì nó ảnh hưởng đến toàn bộ độ chính xác của bản đồ sau khi được số hóa dựa trên nền ảnh. Nắn bản đồ là quá trình chuyển đổi ảnh đang ở tọa độ hàng - cột của các điểm ảnh (pixel) về tọa độ trắc độ thực (hệ tọa độ địa lý hoặc tọa độ phẳng). Việc xác định tọa độ các điểm trên ảnh phải thật chính xác và trùng với bản đồ giấy. Tọa độ một điểm được xác định trên ảnh và thực tế có sự sai lệch nhau, tùy thuộc vào tỷ lệ bản đồ và mục đích thành lập bản đồ mà sai số cho phép sẽ khác nhau. Các điểm định vị trên vừa định nghĩa vùng làm việc cho quá trình số hóa, vừa là cơ sở cho quá trình tiếp biên giữa các mảnh bản đồ.

2.3.3.5 Vector hóa

Vector hóa là quá trình biến đổi dữ liệu raster thành dữ liệu vector, hay nói cách khác đây là quá trình vẽ lại bản đồ giấy trên máy tính hoặc bàn số hóa nhằm tạo một bản vẽ dạng số của bản đồ đó. Hiện nay có rất nhiều phần mềm số hóa bao gồm Autocad, Mapinfo, Arcinfo, Microstation... Sau khi số hóa, tùy thuộc vào phần mềm số hóa mà dữ liệu vector sẽ được tổ chức trong các định dạng files khác nhau

như với Mapinfo sẽ được lưu trữ vào files*.TAB, với Microstation sẽ được lưu trữ vào files*.DGN.

2.3.3.6 Chỉnh sửa dữ liệu

Sau quá trình số hóa, dữ liệu được nhận chưa phải đã hoàn thiện và sử dụng được, các dữ liệu này được gọi là dữ liệu thô, cần phải qua một quá trình chỉnh sửa hợp lệ. Quá trình này bao gồm các công đoạn: lọc bỏ điểm dưa thừa (filter), làm trơn đường (smooth), loại bỏ các đối tượng trùng nhau và tạo các điểm giao.

2.3.3.7 Kiểm tra - Bổ sung đối tượng

Sau khi chỉnh sửa dữ liệu là quá trình kiểm tra tính đầy đủ của đối tượng và độ chính xác của dữ liệu sau khi số hóa. Quá trình này ảnh hưởng đến độ chính xác cũng như chất lượng của sản phẩm sau này. Kiểm tra độ chính xác của dữ liệu là kiểm tra mức độ sai số giữa dữ liệu raster và dữ liệu vector (là độ lệch giữa các đường vector và tâm đường raster), thông thường sai số này phải $< 0,1$ mm tính theo tỷ lệ bản đồ. Kiểm tra tính đầy đủ đối tượng nghĩa là kiểm tra và bổ sung đầy đủ các đối tượng cần thu nhận theo yêu cầu đề ra đối với từng loại bản đồ tài liệu. Khi thực hiện công tác này người kiểm tra phải nắm được toàn bộ các thông số đồ họa quy định cho từng đối tượng, sử dụng thành thạo các công cụ sửa chữa và số hóa đối tượng để khi gặp các lỗi phải tiến hành xử lý ngay.

2.3.3.8 Tiếp biên

Không giống như các bản đồ trên giấy, công tác tiếp biên với các mảnh lân cận phải thực hiện ngay sau khi thu nhận và chỉnh sửa dữ liệu, các đối tượng dạng vùng tô màu phải chưa được tạo (polygon) bởi vì sau khi đóng vùng và tô màu nền, các yếu tố dạng vùng rất khó tiếp biên với nhau.

2.3.3.9 Biên tập và trình bày bản đồ

Dựa vào mục đích - yêu cầu của bản đồ cần thành lập, một lần nữa các đối tượng trên bản đồ được kiểm tra, thay đổi ký hiệu thích hợp và bố trí vị trí các đối tượng nhằm đảm bảo tính tương quan về địa hình cũng như tính thẩm mỹ của bản đồ.

Hiện nay có rất nhiều phần mềm xử lý bản đồ, cách tổ chức và quản lý dữ liệu không gian và dữ liệu thuộc tính ở các phần mềm có khác nhau, nhưng quy trình biên tập chuyển từ bản đồ giấy thành bản đồ số nhìn chung là giống nhau. Với sự phát triển của công nghệ thông tin cùng với nhu cầu về tổ chức xây dựng cơ sở dữ liệu quốc gia, hiện nay việc thành lập bản đồ số thay thế bản đồ giấy là rất cần thiết và là nhiệm vụ cấp bách.

2.3.4 Giới thiệu MapInfo

MapInfo Professional là một sản phẩm phần mềm vẽ bản đồ được sản xuất bởi Công ty MapInfo. MapInfo Professional có khả năng kết hợp và hiển thị dữ liệu từ nhiều nguồn khác nhau có định dạng và hệ qui chiếu khác nhau trên một bản đồ. Phần mềm này có khả năng chồng phủ lên nhau các lớp raster và vector trên cùng một bản đồ; các lớp raster có thể bán trong suốt, để có thể phục vụ như là nhiều hơn chỉ là phông nền.

MapInfo phổ biến cả trong lĩnh vực kinh doanh và công cộng, nơi mà một người dùng thông thường có thể phân tích xây dựng lại các lớp dữ liệu bản đồ.

Chương 3 - CƠ SỞ DỮ LIỆU KHÔNG GIAN VÀ WEBGIS

3.1 CƠ SỞ DỮ LIỆU KHÔNG GIAN

3.1.1 Cấu trúc cơ sở dữ liệu không gian

Dữ liệu trong hệ thống thông tin địa lý là những dữ liệu luôn thay đổi và phức tạp. Chúng bao gồm những mô tả số của hình ảnh bản đồ, mối quan hệ logic giữa các hình ảnh đó, những dữ liệu thể hiện các đặc tính của hình ảnh và các hiện tượng xảy ra tại các vị trí địa lý xác định. Nội dung của CSDL được xác định bởi các ứng dụng khác nhau của hệ thống thông tin địa lý trong một hoàn cảnh cụ thể.

CSDL của hệ thống thông tin địa lý gồm hai phần cơ bản là dữ liệu không gian (dữ liệu bản đồ) và dữ liệu thuộc tính (dữ liệu phi không gian). Mỗi một loại dữ liệu có đặc trưng riêng và chúng khác nhau về yêu cầu lưu trữ, xử lý và hiển thị.

3.1.1.1 Dữ liệu thuộc tính

a. Khái niệm

Là những mô tả về đặc tính, đặc điểm và các hiện tượng xảy ra tại vị trí địa lý xác định mà chúng khó hoặc không thể biểu thị trên bản đồ được. Cũng như các hệ thống thông tin địa lý khác, hệ thống này có 4 loại dữ liệu thuộc tính:

b. Đặc tính của đối tượng

Liên kết chặt chẽ với các thông tin đồ thị, các dữ liệu này được xử lý theo ngôn ngữ hỏi đáp cấu trúc (SQL) và phân tích. Chúng được liên kết với các hình ảnh đồ thị thông qua các chỉ số xác định chung, thông thường gọi là mã địa lý và được lưu trữ trong cả hai mảng đồ thị và phi đồ thị. Hệ thống thông tin địa lý còn có thể xử lý các thông tin thuộc tính riêng rẽ và tạo ra các bản đồ chuyên đề trên cơ sở các giá trị thuộc tính. Các thông tin thuộc tính này cũng có thể được hiển thị như là các ghi chú trên bản đồ hoặc là các tham số điều khiển cho việc lựa chọn hiển thị các thuộc tính đó như là các ký hiệu bản đồ.

c. Dữ liệu tham khảo địa lý

Mô tả các sự kiện hoặc hiện tượng xảy ra tại một vị trí xác định. Không giống các thông tin đặc tính, chúng không mô tả về bản thân các hình ảnh bản đồ, thay vào đó chúng mô tả các danh mục hoặc các hoạt động như cho phép xây dựng các khu công nghiệp mới, nghiên cứu y tế, báo cáo hiểm họa môi trường... liên quan đến các vị trí địa lý xác định. Các thông tin tham khảo địa lý đặc trưng được lưu trữ và quản lý trong các tập tin độc lập và hệ thống không thể trực tiếp tổng hợp với các hình ảnh bản đồ trong CSDL của hệ thống. Tuy nhiên các bản ghi này chứa các yếu tố xác định vị trí của sự kiện hay hiện tượng.

d. Quan hệ không gian giữa các đối tượng

Quan trọng cho các chức năng xử lý của hệ thống thông tin địa lý. Các mối quan hệ này có thể đơn giản hay phức tạp như sự liên kết, khoảng cách tương thích, mối quan hệ topo giữa các đối tượng.

3.1.1.2 Dữ liệu không gian

a. Khái niệm:

Dữ liệu không gian là những mô tả số của hình ảnh bản đồ. Chúng bao gồm tọa độ, quy luật và các ký hiệu dùng để xác định hình ảnh cụ thể của bản đồ trong một khuôn dạng hiểu được của máy tính. Hệ thống thông tin địa lý dùng các dữ liệu không gian để tạo ra một bản đồ hay hình ảnh bản đồ trên màn hình hoặc trên giấy thông qua thiết bị ngoại vi như máy in, máy chiếu...

Dữ liệu không gian có hai mô hình lưu trữ: vector và raster

b. Dữ liệu dạng Vector:

Dữ liệu dạng vector là các điểm tọa độ (X,Y) hoặc là các quy luật tính toán tọa độ và nối chúng thành các đối tượng trong một hệ thống tọa độ nhất định.

Các kiểu đối tượng địa lý dạng vector:

- Kiểu điểm: 1 tọa độ (x, y) trong 2D hoặc 1 tọa độ (x, y, z) trong 3D, 0 chiều.
- Kiểu đường: danh sách các tọa độ $x_1y_1, x_2y_2 \dots x_ny_n$ hoặc là một hàm toán học, 1 chiều, tính được chiều dài.

- Kiểu vùng: tập các đường khép kín, 2 chiều, tính được chu vi và diện tích
- Kiểu bề mặt: chuỗi tọa độ xyz, hàm toán học, 3 chiều, tính được diện tích bề mặt, thể tích.

Những ưu điểm của cấu trúc Vector:

- Ít trường hợp dữ liệu bị dày bộ nhớ trong máy tính vì tổ chức dữ liệu Vector thường ở dạng nén, vì vậy có thể chứa được một lượng dữ liệu Vector rất lớn trong dữ liệu không gian.
- Các đối tượng riêng biệt được thể hiện một cách rõ ràng và liên tục bằng những đường nét rõ ràng.
- Các yếu tố không gian về mặt hình học thì dễ dàng được xác định.
- Có độ chính xác cao trong việc tính toán và xử lý các yếu tố không gian.

Nhược điểm:

- Nhược điểm lớn nhất của cấu trúc dữ liệu Vector là xử lý chòng xếp các lớp bản đồ rất khó thực hiện được, ngay cả những việc chòng xếp rất đơn giản của dữ liệu Raster.

c. *Dữ liệu Raster (ảnh đối tượng):*

Dữ liệu Raster là dữ liệu được tạo thành bởi các ô lưới có độ phân giải xác định. Loại dữ liệu này chỉ dùng cho mục đích diễn tả và minh họa chi tiết bằng hình ảnh cho các đối tượng của hệ thống. Các đặc điểm:

- Các điểm được xếp liên tiếp từ trái qua phải và từ trên xuống dưới.
- Mỗi một điểm ảnh (pixel) chứa một giá trị.
- Một tập các ma trận điểm và các giá trị tương ứng tạo thành một lớp (layer).
- Trong CSDL có thể có nhiều lớp.

Mô hình dữ liệu raster là mô hình dữ liệu GIS được dùng tương đối phổ biến trong các bài toán về môi trường, quản lý tài nguyên thiên nhiên.

Mô hình dữ liệu raster chủ yếu dùng để phản ánh các đối tượng dạng vùng là ứng dụng cho các bài toán tiến hành trên các loại đối tượng dạng vùng: phân loại; chia nhỏ. Các nguồn dữ liệu xây dựng nên dữ liệu raster có thể bao gồm [5]:

- Quét ảnh
- Ảnh máy bay, ảnh viễn thám
- Chuyển từ dữ liệu vector sang
- Lưu trữ dữ liệu dạng raster.
- Nén theo hàng (Run length coding).
- Nén theo chia nhỏ thành từng phần (Quadtree).
- Nén theo ngữ cảnh (Fractal).

Trong một hệ thống dữ liệu cơ bản raster được lưu trữ trong các ô (thường hình vuông) được sắp xếp trong một mảng hoặc các dãy hàng và cột. Nếu có thể, các hàng và cột nên được căn cứ vào hệ thống lưới bản đồ thích hợp.

Những ưu điểm cơ bản của cấu trúc Raster:

- Đơn giản và dễ tham khảo.
- Việc chia nhỏ các lớp bản đồ được thực hiện một cách thuận tiện.
- Dễ dàng vẽ ra được.
- Dễ thiết lập một bìa mặt liên tục bằng phương pháp nội suy.
- Đa số các tư liệu không gian thường được ghi ở dạng Raster như ảnh vệ tinh, ảnh máy bay chụp quét. Thông thường các tư liệu Raster đó dễ dàng nhập trực tiếp.

Những nhược điểm của cấu trúc dữ liệu Raster:

- Tài liệu thường bị tình trạng quá tải, làm tốn nhiều phần của bộ nhớ trong máy tính. Trong rất nhiều trường hợp, các yếu tố bản đồ không nhất thiết phải được gắn thuộc tính (code hóa) thành các ô lưới đặc trưng.
- Mỗi quan hệ về hình học giữa các yếu tố không gian thì khó vẽ và khó thiết lập được, ví dụ với hai bản đồ được xác định bằng hàng, cột thì mối liên hệ hình học giữa các đặc điểm của hai bản đồ đó là rất khó xác định.

- Các bản đồ Raster thường thô và kém vẻ đẹp hơn so với bản đồ vẽ bằng đường nét thanh của cấu trúc Vector. Trong bản đồ Raster, các yếu tố đường, sông ... ranh giới thường được biểu hiện bằng các pixel nên có dạng răng cưa.
- Việc chuyển đổi các thuộc tính không gian của cấu trúc Raster thì dễ bị nhiễu. Ví dụ một con đường khi quay đi một góc nào đó rồi quay lại đúng góc đó nhưng nó có thể bị biến đổi so với hình dạng ban đầu.
- Đối với phân tích không gian, hạn chế nhất của cấu trúc Raster là độ chính xác thường thấp so với mong muốn.

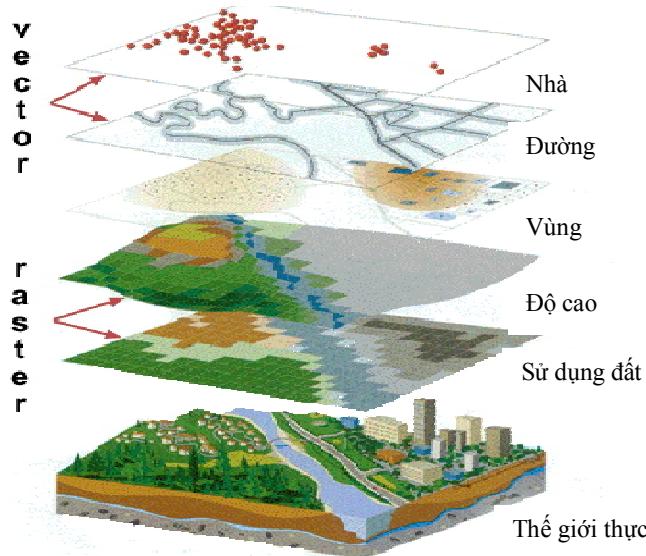
3.1.1.3 Cách tổ chức dữ liệu không gian

Để tiện phân tích và tổng hợp, dữ liệu không gian thường được tổ chức thành các lớp (layer/theme); cũng thường được gọi là các lớp dữ liệu chuyên đề (thematic layer).

Mỗi lớp dữ liệu thường biểu diễn 1 tính chất liên quan đến vị trí trên mặt đất. Ví dụ: lớp dữ liệu về ranh giới hành chánh, đường đi, các địa điểm quan trọng....

Mỗi lớp dữ liệu có thể có một hay nhiều kiểu đối tượng địa lý (điểm, đường, vùng). Trên một lớp dữ liệu, tại một vị trí không thể có cùng lúc hai giá trị riêng biệt. Ví dụ: trên lớp dữ liệu về trường học, tại một vị trí nào đó không thể vừa là trường A vừa là trường B.

Cách tổ chức dữ liệu thành các lớp chuyên đề cho phép thể hiện thế giới thực phức tạp một cách đơn giản nhằm giúp hiểu biết các quan hệ trong thiên nhiên.



Hình 3.1- Biểu diễn thế giới thực bằng các lớp chuyên đề

3.1.2 Cách thức lưu trữ - Quan hệ không gian topology

Topology là mối quan hệ logic giữa vị trí của các đối tượng, là một lãnh vực toán học.

Cấu trúc dữ liệu thuộc topology có lợi cho việc xử lý việc số hóa, xử lý lỗi; giảm dung lượng lưu trữ dữ liệu cho các vùng vì các ranh giới giữa những vùng nằm kề nhau được lưu trữ chỉ một lần; và cho phép chúng ta cấu trúc dữ liệu dựa trên các nguyên lý về tính kề cận (adjacency) và kết nối (connectivity) để xác định các quan hệ không gian. Phần lớn cấu trúc dữ liệu mang tính topology là mô hình dữ liệu vector kiểu cung/nút (arc/node).

- Cung: là 1 chuỗi các đoạn thẳng nối giữa các nút, có nút đầu và nút cuối.
- Nút: là nơi hai cung gặp nhau.
- Điểm: là các nút độc lập.
- Vùng : là chuỗi khép kín các cung.

Quan hệ không gian của các đối tượng trong các phần mềm GIS được xây dựng theo khuôn dạng thích hợp. Thường được lập thành 3 bảng (table) có quan hệ, tương ứng với 3 kiểu đối tượng: điểm, đường và vùng.

3.1.3 Hệ cơ sở dữ liệu không gian

3.1.3.1 Định nghĩa

Güting (1994) định nghĩa hệ cơ sở dữ liệu không gian như là một lớp của hệ cơ sở dữ liệu gồm có 3 đặc tính sau:

- Một hệ cơ sở dữ liệu không gian là một hệ cơ sở dữ liệu.
- Nó cung cấp các loại dữ liệu không gian trong mô hình dữ liệu và ngôn ngữ truy vấn.
- Nó hỗ trợ các loại dữ liệu không gian, lập chỉ mục không gian và các giải thuật phân tích không gian.

3.1.3.2 Các tính chất của hệ cơ sở dữ liệu không gian

CSDL không gian là một CSDL được tối ưu để lưu trữ và truy vấn dữ liệu liên quan đến các đối tượng trong không gian, bao gồm các điểm, đường thẳng và đa giác. Trong một bảng của CSDL không gian, ngoài các cột có kiểu dữ liệu kí tự và kiểu số, còn chứa các cột kiểu dữ liệu không gian (ví dụ như: kiểu hình học (Geometry) hoặc kiểu đối tượng (Feature)) để hỗ trợ xử lý tính toán và truy vấn thông tin không gian. Tổ chức Open Geospatial Consortium đã đưa ra đặc tả cho các kiểu đối tượng đơn và thiết lập các chuẩn đối với việc bổ sung các chức năng không gian cho các hệ thống CSDL.

3.1.3.3 Kiểu dữ liệu không gian

Thông thường, hệ cơ sở dữ liệu được thiết kế để quản lý và xử lý các dữ liệu dạng chuỗi, số, ngày, boolean hay các biểu thức. Không có qui định trong các hệ thống cơ sở dữ liệu quan hệ để hỗ trợ lưu trữ và xử lý dữ liệu không gian. Cơ sở dữ liệu đối tượng và quan hệ cho phép người dùng định nghĩa kiểu dữ liệu trừu tượng để mô tả cấu trúc cột trong cơ sở dữ liệu phức tạp. Một số hệ cơ sở dữ liệu đã sử dụng khả năng này để xác định kiểu dữ liệu không gian để quản lý và xử lý dữ liệu không gian.

Mô hình đối tượng hình học của OGC (hình PL.B.1) cung cấp một chuẩn về kiểu dữ liệu không gian. Tuy nhiên mỗi hệ cơ sở dữ liệu khác nhau thực thi chuẩn theo nhiều cách khác nhau.

Ngoài các dữ liệu về vector, các hệ cơ sở dữ liệu cũng định nghĩa các dữ liệu Raster (các ảnh vệ tinh, ảnh viễn thám,...) như là các kiểu dữ liệu trùu tượng trong cơ sở dữ liệu đối tượng và quan hệ. Như vậy các dữ liệu Vector và Raster có thể được quản lý và xử lý trong hệ cơ sở dữ liệu không gian.

3.1.4 Giới thiệu PostgreSQL (PostGIS)

3.1.4.1 Khái niệm

PostgreSQL là một hệ quản trị cơ sở dữ liệu đối tượng, quan hệ (ORDBMS) dựa trên Postgres phiên bản 4.21, được phát triển tại Khoa khoa học máy tính Đại học California tại Berkeley.

3.1.4.2 Các đặc điểm của PostgreSQL

- Hoạt động được trên nhiều hệ điều hành như: Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) và Windows.
- Lưu trữ (dạng nhị phân) các đối tượng có dữ liệu lớn như hình ảnh, âm thanh, hoặc video.
- Hỗ trợ các ngôn ngữ như C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC.
- Có sự thống nhất giữa những người phát triển (tái sử dụng những kỹ năng và các thư viện đã có)
- Có hầu hết các truy vấn SQL với các kiểu dữ liệu như INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL và TIMESTAMP.
- Tạo index giúp quá trình truy vấn đạt hiệu quả cao hơn
- Hỗ trợ các chức năng tìm kiếm tổng quát hóa có thể “gắn” các chỉ mục vào giúp quá trình tìm kiếm được thực hiện trên phạm vi rộng hơn như: compound, unique, partial, and functional indexes.

- Cho phép người dùng tạo các kiểu dữ liệu mới, các hàm, các thủ tục...
- PostgreSQL không quy định những hạn chế trong việc sử dụng mã nguồn của phần mềm nên có thể được dùng, sửa đổi và phổ biến bởi mọi người cho bất kỳ mục đích nào.
- Tốc độ: Dữ liệu từ cơ sở dữ liệu sẽ phục vụ tốt hơn dữ liệu đọc từ shape file. Cơ sở tối ưu hóa để phục vụ một khối lượng lớn dữ liệu lặp và dữ liệu không gian phù hợp với mô tả này hoàn hảo.
- Hỗ trợ đa người dùng: dữ liệu không gian có xu hướng là các dữ liệu tham chiếu, tức là được chia sẻ bởi nhiều người dùng. Lưu trữ dữ liệu trong hệ quản trị cơ sở dữ liệu sẽ cung cấp những lợi ích như: truy cập từ xa thông qua các chuẩn giao tiếp như: JDBC, ODBC, PERL/DBI..., làm tăng tính bảo mật cho dữ liệu như: một số người dùng có toàn quyền tuy nhiên một số người dùng chỉ có thể đọc hay không được nhìn thấy.
- Khả năng truy vấn: đây là lợi ích lớn nhất.
- Cho phép thực hiện các câu truy vấn truyền thống: Ví dụ như: Cho tôi biết tất cả các thông tin của chợ Xuân Khánh.
- Cho phép thực hiện các câu truy vấn không gian: Ví dụ như: Cho tôi biết tất cả các khách sạn cách Đại học Cần Thơ với khoảng cách là 5km.

3.1.4.3 Giới thiệu PostGIS

PostGIS là một CSDL không gian được tích hợp vào hệ quản trị cơ sở dữ liệu PostgreSQL. Nó hỗ trợ tất cả các hàm và đối tượng được định nghĩa trong chuẩn OpenGIS. Việc sử dụng những hàm tính toán không gian dựa trên PostGIS thuận lợi cho việc xử lý dữ liệu không gian và truy vấn hoàn toàn dựa trên các câu lệnh SQL.

PostGIS một CSDL không gian có những ưu điểm để xử lý các thông tin về hình dạng không gian như trả lời các truy vấn về những đối tượng ở gần một vị trí nào đó, những đối tượng nằm trong phạm vi hoặc ở vùng phụ cận của một đối

tượng khác, phạm vi của một vùng nơi mà có một hoạt động nào đó đang xảy ra là gì, những đối tượng nào nằm bên trong một đối tượng khác. . .

PostGIS có khả năng lưu trữ và thao tác với dữ liệu địa lý rất mạnh. Nó cung cấp những khả năng xử lý thông tin địa lý bên trong một môi trường CSDL. Những hàm SQL bao gồm buffer, intersection, within, distance... Những hàm này lấy dữ liệu hình học từ các cột trong bảng PostGIS và trả về những hình học mới hoặc những thông tin khác. Ví dụ như hàm distance sẽ tính toán khoảng cách giữa các đối tượng (feature) không gian, và hàm sẽ trả về một hình mới là một đa giác được làm vùng đậm tại một khoảng cách nào đó từ đối tượng nguồn.

PostGIS cho phép dễ dàng khi kết nối dữ liệu không gian với dữ liệu phi không gian trong một môi trường dữ liệu không gian và cung cấp đầy đủ sức mạnh của ngôn ngữ truy vấn cấu trúc (SQL) để thực hiện những phân tích khác, chẳng hạn như tính chi phí trung bình cho một vùng địa lý....

PostGIS còn hỗ trợ thêm chỉ mục không gian (spatial indexes) dựa trên GIST and R-Tree, đây là điểm riêng biệt nhất của PostGIS so với các hệ quản trị CSDL khác giúp PostGIS dễ dàng quản lý, lưu trữ, khai thác và tìm kiếm thông tin không gian.

3.1.4.4 Một số đối tượng theo chuẩn OGC trong PostGIS

PostGIS hỗ trợ các đối tượng và các hàm được định nghĩa chi tiết trong OGC "Simple Features for SQL". Với PostGIS ta có thể lưu trữ các điểm, đường, vùng, tập hợp điểm, tập hợp đường, tập hợp vùng và các thông tin hình học khác. PostGIS mở rộng chuẩn với sự hỗ trợ cho các tọa độ dạng 3DZ, 3DM và 4D.

OpenGIS định nghĩa 2 chuẩn để diễn tả các đối tượng không gian: Well-known Text (WKT) và Well-Known Binary (WKB). Cả WKT và WKB đều bao gồm thông tin về kiểu của đối tượng và tọa độ của nó.

PostGIS được hiện thực tuân theo đặc tả "OGC Simple Feature Specifications" cho chuẩn SQL. Bản đặc tả này định nghĩa những thuật toán và lược đồ SQL để thêm (insert), truy vấn (query), thao tác (manipulate) và xóa (delete) những đối

tượng không gian. Những tọa độ của các đối tượng không gian được lưu trữ trong các bảng. Mỗi một bảng có thể chứa một kiểu đối tượng hình học (geometry) là điểm (point, multipoint), đường (linestring, multilinestring), đa giác (polygon, multipolygon và geometry collection). Những tọa độ của mỗi đối tượng được lưu trữ trong một trường với một kiểu riêng biệt.

Kiểu mô tả một tập hợp tọa độ không gian của đối tượng là WKT (Well Known Text). Khi những đối tượng không gian được đưa vào cơ sở dữ liệu thì PostGIS sẽ chuyển chúng từ WKT sang WKB (Well Known Binary) để việc lưu trữ được tốt hơn.

Ví dụ về miêu tả WKT:

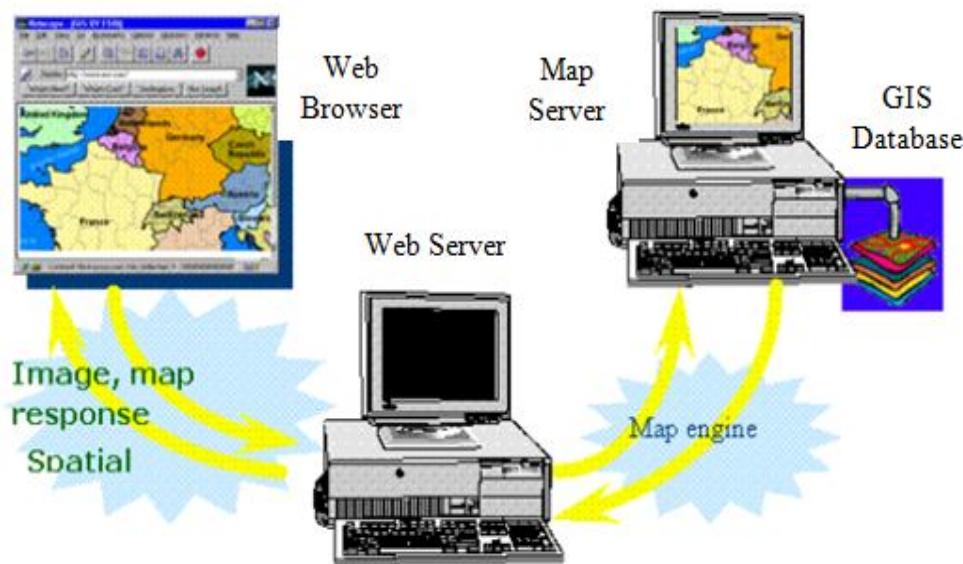
- POINT(30 10)
- LINESTRING(30 10, 10 30, 40 40)
- POLYGON((30 10, 10 20, 20 40, 40 40, 30 10))
- POLYGON((35 10, 10 20, 15 40, 45 45, 35 10),
(20 30, 35 35, 30 20, 20 30))
- MULTIPOINT((10 40), (40 30), (20 20), (30 10))
- MULTILINESTRING((10 10, 20 20, 10 40),
(40 40, 30 30, 40 20, 30 10))
- MULTIPOLYGON (((30 20, 10 40, 45 40, 30 20)),
((15 5, 40 10, 10 20, 5 10, 15 5)))
- MULTIPOLYGON(((40 40, 20 45, 45 30, 40 40)),
((20 35, 45 20, 30 5, 10 10, 20 35),
(30 20, 20 25, 20 15, 30 20)))

3.2 CÔNG NGHỆ WEBGIS

3.2.1 Giới thiệu về WebGIS

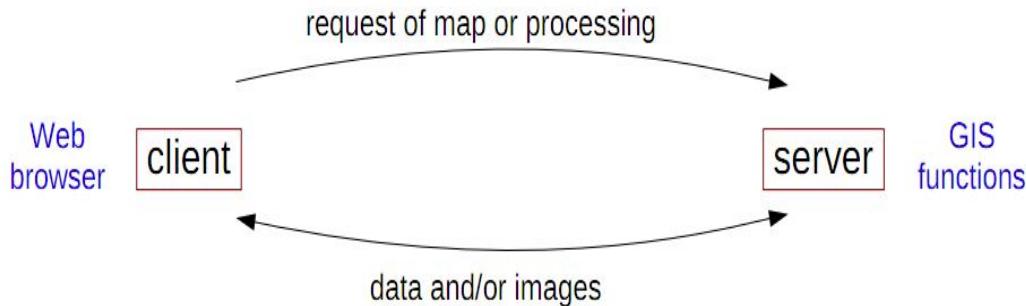
GIS có nhiều định nghĩa nên WebGIS cũng có nhiều định nghĩa. Nói chung, các định nghĩa của WebGIS dựa trên những định nghĩa đa dạng của GIS và có thêm các thành phần của Web (web component). Đây là một số định nghĩa về WebGIS.

WebGIS là hệ thống thông tin địa lý (Geographic Information System -GIS) được phân bố thông qua hệ thống mạng máy tính phục vụ cho việc tích hợp, phổ biến (disseminate) và giao tiếp với các thông tin địa lý được hiển thị trên World Wide Web (Edward, 2000, URL)



Hình 3.2 - Sơ đồ hoạt động của WebGIS

WebGIS là một hệ thống phức tạp cho phép truy cập trên mạng với những chức năng: thu nhận (capturing), lưu trữ (storing), tích hợp (integrating), xử lý (manipulating), phân tích (analyzing) và hiển thị dữ liệu theo vị trí tọa độ không gian mà không cần phải sử dụng các phần mềm GIS (theo Harder 1998). Trong cách thực hiện nhiệm vụ phân tích GIS, dịch vụ này gần giống như là kiến trúc Client-Server của Web.



Hình 3.3 - Kiến trúc Client- Server trong WebGIS

Xử lý thông tin địa lý được chia ra làm các nhiệm vụ ở phía server và phía client. Điều này cho phép người dùng có thể truy xuất, thao tác và nhận kết quả từ việc khai thác dữ liệu GIS từ trình duyệt web của họ mà không cần phải trả tiền cho phần mềm GIS.

Một client tiêu biểu là trình duyệt web và server bao gồm một web server có cung cấp chương trình phần mềm WebGIS. Client thường yêu cầu một ảnh bản đồ hay vài xử lý thông tin địa lý qua web đến server ở xa. Server chuyển đổi yêu cầu thành mã nội bộ và gọi những chức năng về GIS và bằng cách chuyển tiếp yêu cầu tới phần mềm WebGIS. Phần mềm này trả về kết quả, sau đó kết quả này được định dạng lại cho việc trình bày bởi trình duyệt hay những hàm từ các plugin. Server sau đó trả về kết quả cho client hiển thị hoặc gửi dữ liệu và các công cụ phân tích đến client để dùng ở phía client.

Phần lớn sự chú ý gần đây là tập trung vào việc phát triển các chức năng GIS có tiềm năng lớn trong việc làm cho thông tin địa lý trở nên hữu dụng và sẵn sàng tới số lượng lớn người dùng trên toàn thế giới. Thách thức lớn nhất của WebGIS là việc tạo ra một hệ thống phần mềm không phụ thuộc vào platform và chạy trên chuẩn giao thức mạng TCP/IP, có nghĩa là khả năng WebGIS được chạy trên bất kỳ trình duyệt Web của bất kỳ máy tính nào nối mạng Internet. Đối với vấn đề này các phần mềm GIS phải được thiết kế lại để trở thành ứng dụng WebGIS theo các kỹ thuật mạng Internet.

WebGIS có rất nhiều ưu điểm so với các phần mềm GIS desktop:

- Khả năng phân phối thông tin địa lý rộng rãi trên toàn thế giới.

- Người dùng Internet có thể truy cập đến các ứng dụng GIS mà không phải mua phần mềm.
- Đối với phần lớn người dùng không có kinh nghiệm về GIS thì việc sử dụng WebGIS sẽ đơn giản hơn việc sử dụng các ứng dụng GIS loại khác.
- WebGIS là chức năng bổ sung cho GIS hoạt động trong môi trường rộng hơn thông qua mạng và được ứng dụng ngày càng rộng rãi trong thương mại, quản lý nhà nước và giáo dục. Nhiều ứng dụng sẽ được chạy trên mạng nội bộ trong doanh nghiệp và cơ quan chính phủ như là một phương tiện phân phối và sử dụng dữ liệu không gian địa lý chung.
- Nhiều nghiên cứu hiện nay về WebGIS liên quan đến các ứng dụng xây dựng bản đồ tương tác, một trong những lĩnh vực quan trọng của sự đổi mới liên quan đến việc "chỉ trả cho sử dụng" của dịch vụ bản đồ và GIS.

3.2.2 Phân loại WebGIS

3.2.2.1 Phân loại theo kiến trúc

Internet theo triết lý dựa trên kiến trúc khách/chủ (Client/Server). Sự kết hợp giữa Client /Server sẽ cho ra các trang WebGIS có các chức năng khác nhau (theo Nguyễn Cao Tùng - Sách do TBI tài trợ).

- *Thuần Server*: Máy chủ (Server) sẽ đảm nhiệm tất cả các công việc bao gồm lưu trữ và phân tích dữ liệu. Tất cả các khách hàng sẽ sử dụng các dữ liệu hoặc các chức năng hiện có trên máy chủ và không được quyền thay đổi các thông tin này. Người dùng đưa các nhu cầu về thông tin và phân tích lên Máy chủ thông qua web và máy chủ sẽ xử lý thông tin rồi trả lại thông tin cho người dùng. Kiến trúc này thường áp dụng cho các trang WebGIS thương mại hoặc cộng đồng có số lượng người dùng lớn và không quan tâm đến khả năng xử lý GIS trên các máy của người dùng.

- *Thuần Client*: Máy khách được cung cấp các chức năng để xử lý các yêu cầu về GIS để giám bớt công việc cho máy chủ. Các chương trình con có chức năng xử

lý GIS (GIS applet) sẽ được phân phối đến máy tính của người dùng theo 2 cách: phân phối khi có yêu cầu và phân phối cài đặt cố định ở máy khách.

- *Kết hợp Server và Client*: hai kiến trúc nêu trên đều có ưu và nhược điểm. Kiến trúc thuần Server sẽ chịu ảnh hưởng lớn của chất lượng đường truyền; kiến trúc thuần Client lại phụ thuộc vào chất lượng của máy Client. Từ đó giải pháp kết hợp Server và Client ra đời nhằm tận dụng các ưu điểm của 2 kiến trúc nêu trên. Có thể kết hợp bằng cách dữ liệu lưu trên máy chủ, các chức năng xử lý đặt tại máy khách. Cũng có thể kết hợp bằng cách máy chủ cung cấp các chức năng, dữ liệu lưu ở máy khách. Hoặc cũng có thể kết hợp theo cách dữ liệu và chức năng vừa lưu ở máy chủ, vừa cung cấp các chức năng xử lý đơn giản cho máy khách. Phương án này có thể tối ưu hóa được khả năng xử lý và đáp ứng các nhu cầu đặc biệt của người dùng. Thông thường các công việc đòi hỏi phải có dữ liệu lớn và tính toán phức tạp thì giao cho server xử lý, các công việc đòi hỏi người dùng có quyền điều khiển cao thì giao cho Client. Giải pháp này đặc biệt hữu ích khi thỉnh thoảng Client mới cần liên lạc với Server để lấy dữ liệu.

3.2.2.2 Phân loại theo kỹ thuật

- *Thuần HTML*: Thuật ngữ “thuần HTML” dùng để chỉ những trang WebGIS được tạo ra bằng cách sử dụng các khả năng của HTML (HyperText Markup Language – ngôn ngữ siêu văn bản), không dùng các phần mở rộng từ phía máy chủ (Server side extensions) hoặc các Scripts. Các phần tử của nó chỉ sử dụng các liên kết đơn giản (simple link) và các hình ảnh. Các chức năng của nó được thực hiện thông qua cấu trúc liên kết HTML chỉ cho phép dùng các hình ảnh dạng raster, vì vậy chỉ có những bản đồ raster mới được phân phối trên nó. Ưu điểm của kỹ thuật này là: thứ nhất chỉ cần dùng các Server thấp để xử lý, vì Server ở đây làm việc như là một WebServer dùng phân phối các trang HTML và các hình ảnh. Thứ hai, không cần phần mềm GIS nào chạy trong lúc thực thi các yêu cầu của người dùng. Thứ ba, phía máy khách chỉ cần dùng trình duyệt Web chuẩn. Nhược điểm: chỉ dùng cho các nhiệm vụ có cấu trúc đơn giản.

- *HTML với các chương trình thực thi trên Server:* Trái ngược với giải pháp “thuần HTML”, đây là giải pháp tạo ra một trang HTML WebGIS do một chương trình đang chạy trên Server, chương trình này có khả năng đáp ứng đầy đủ các yêu cầu. Máy khách gửi một yêu cầu đến WebServer, chương trình trên máy chủ sẽ nhận được yêu cầu này và tạo ra kết quả riêng, kết quả này có thể là một bản đồ (raster) hoặc là một trang HTML. Các tham số có thể được gửi kèm đến chương trình, các tham số này được lấy từ các thành phần của trang HTML như text boxes, combo boxes, radio button,.. Khi chuyển đến máy chủ, tọa độ của điểm click trên bản đồ cũng được chuyển kèm theo. Vì vậy các chương trình trên máy chủ sẽ xử lý các hành động tương ứng với tọa độ điểm click. Người dùng có thể phóng to, thu nhỏ, dịch chuyển hoặc thực hiện các hành động tại các điểm click trên bản đồ chính xác. Kết nối giữa WebServer và các chương trình trả lời có thể thực hiện thông qua CGI (Common Gateway Interface) hoặc bất cứ giao diện nào được cung cấp mà giao tiếp được với WebServer.

- *Các giải pháp dựa trên JAVA:* Java là ngôn ngữ lập trình hướng đối tượng, đa luồng, đa mục đích và thích hợp để tạo ra các ứng dụng cho Internet và các mạng phân tán phức hợp khác. Những chương trình được tạo ra từ Java không quan tâm đến hệ máy mà bạn đang dùng. Mỗi kiểu máy tính và các hệ điều hành có phiên bản cụ thể gọi là máy ảo Java (Java Virtual Machine). Máy ảo Java giúp chuyển đổi từ trình Java sang dạng mà các loại máy tính và hệ điều hành cụ thể đều hiểu được. Quá trình chuyển đổi này hoàn toàn tự động và xuyên suốt đối với người dùng. Như vậy các ứng dụng được viết từ ngôn ngữ Java sẽ vận hành được trên bất cứ các loại máy nào trang bị Java Virtual Machine. Chương trình Java sẽ được tải về trình duyệt của máy khách khi cần thiết. Chương trình Java (applets) sẽ khởi động và có thể yêu cầu dữ liệu cần thiết từ máy chủ. Nói chung giải pháp dựa trên Java rất uyển chuyển bởi nó tránh được những giới hạn của HTML thuần. Ưu điểm của việc dùng Java là công việc có thể thực thi trên máy khách mà không cần phải truyền về máy chủ. Nhược điểm là việc sử dụng Java để giải quyết công việc hiện nay trên mạng là rất chậm, một số ứng dụng Java không hoạt động sau bức tường lửa.

3.2.2.3 Phân loại theo dịch vụ

Theo Claus Rinner (viện địa tin học, đại học tổng hợp Muenster, Đức) thì có 5 loại dịch vụ WebGIS sau:

- *Geodata Server*: Chỉ lưu trữ dữ liệu địa lý và cung cấp dữ liệu cho máy khách khi có yêu cầu thông qua Internet. Máy khách tải dữ liệu về và dùng các chương trình GIS trên máy khách để xử lý dữ liệu. Ưu điểm của dịch vụ này là dữ liệu có sẵn mà không cần phải số hóa.

- *Map Server*: Lưu trữ dữ liệu và cung cấp các bản đồ thông qua WWW, kèm theo một số chức năng như Zoom, Pan và một số tham số như hiển thị lớp, lựa chọn màu sắc. Ở đây máy chủ xử lý toàn bộ, máy khách chỉ hiển thị các bản đồ do máy chủ cung cấp (ảnh raster).

- *Online Retrieval System*: Phân phối các bản đồ chuyên đề và một số hàm phân tích. Khi sử dụng các hàm này, kết quả sẽ trả về cho máy khách, dữ liệu trả về ở đây là dữ liệu đã được phân tích.

- *Online GIS*: Lớn hơn Online Retrieval System vì nó có khả năng thực hiện các chức năng phân tích thực (real analysis functions) trên các tiến trình của dữ liệu đưa vào.

- *GIS Function Server*: Giống như Online GIS nhưng chỉ khác là GIS Function Server không chứa dữ liệu, nếu Online GIS chỉ trả về dữ liệu thì GIS Function Server trả về các chương trình phân tích nhỏ để máy khách có thể xử lý. GIS Function Server cung cấp các chức năng GIS giải quyết trên dữ liệu do người dùng Upload lên hoặc cung cấp các hàm phân tích cho máy khách download về. Như vậy việc phân tích không gian không còn giới hạn ở việc dữ liệu lưu trên máy chủ.

3.2.3 Mô hình xử lý và kiến trúc triển khai WebGIS

Kiến trúc web của hệ thống thông tin dữ liệu không gian cũng phải thỏa mãn kiến trúc 3 tầng dành cho mọi hệ thống thông tin dạng web cơ bản. Tùy thuộc vào từng công nghệ riêng biệt và phát triển ứng dụng của từng hãng mà chúng có khả năng phát triển, mở rộng thành kiến trúc đa tầng (n-tier) hay không.

Cơ sở dữ liệu không gian sẽ được dùng để quản lý và truy xuất dữ liệu không gian được đặt trên Data Server. Cleaning House là nơi được dùng để lưu trữ và duy trì những siêu dữ liệu về dữ liệu không gian tại những Data Server khác nhau. Dựa trên những thành phần quản lý dữ liệu, ứng dụng server và mô hình server được dùng cho ứng dụng hệ thống để tính toán thông tin không gian thông qua các hàm cụ thể tại Application Server. Tất cả kết quả tính toán của Application Server sẽ được gửi đến Web Server để thêm vào các gói HTML, gửi cho client và hiển thị;

3.2.3.1 Các bước xử lý

- Client gửi yêu cầu người dùng thông qua giao thức HTTP đến Web Server.
- Web Server nhận yêu cầu của người dùng gửi đến từ phía client, xử lý và chuyển tiếp yêu cầu đến ứng dụng trên Application Server có liên quan.
 - Application server (Spatial Specific Server) nhận các yêu cầu cụ thể đối với ứng dụng và gọi các hàm có liên quan để tính toán xử lý. Nếu có yêu cầu dữ liệu nó sẽ gửi yêu cầu dữ liệu đến Data Exchange center (Cleaning House).
 - Data exchange center nhận yêu cầu dữ liệu và tìm kiếm vị trí của những dữ liệu này sau đó gửi yêu cầu dữ liệu đến server chứa dữ liệu (Data Server) tương ứng cần tìm.
 - Data Server dữ liệu tiến hành truy vấn lấy ra dữ liệu cần thiết và trả dữ liệu này về cho Data Exchange Server.
 - Data Exchange Server nhận dữ liệu từ nhiều nguồn data server khác nhau nắm rải rác trên mạng. Sắp xếp dữ liệu lại theo logic của yêu cầu dữ liệu, sau đó gửi trả dữ liệu về cho Application Server.
 - Application Server nhận dữ liệu trả về từ các Data Exchange Server và đưa chúng đến các hàm cần sử dụng, xử lý chúng tại đây và kết quả được trả về cho Web Server.
 - Web Server nhận về kết quả xử lý, thêm vào các ngữ cảnh web (HTML, PHP..) để có thể hiển thị được trên trình duyệt và cuối cùng gửi trả kết quả về cho trình duyệt dưới dạng các trang web.

3.2.3.2 Các kiến trúc triển khai

* *Client side*: Client-side được dùng để hiển thị kết quả đến cho người dùng, nhận các điều khiển trực tiếp từ người dùng và tương tác với web server thông qua trình duyệt web. Các trình duyệt web sử dụng chủ yếu HTML để định dạng trang web.Thêm vào đó một vài plug-in, ActiveX và các mã Applet được nhúng vào trình duyệt để tăng tính tương tác với người dùng.

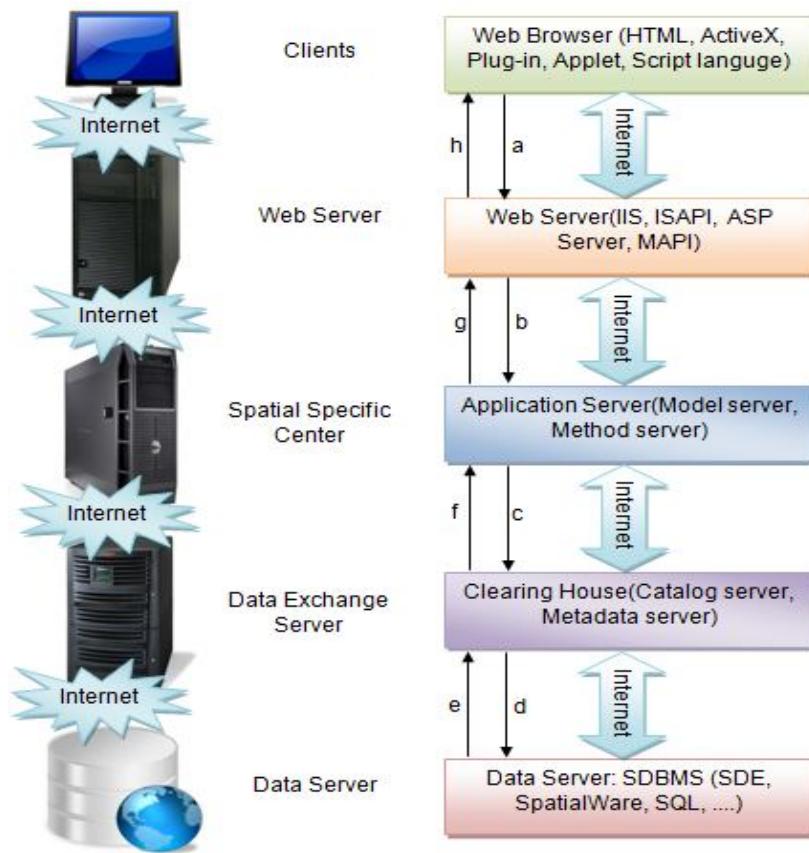
* *Server side*: Gồm có: Web server, Application server, Data server và Clearinghouse Server. Server-side có nhiệm vụ lưu trữ dữ liệu, xử lý tính toán và trả về kết quả (dưới dạng hiển thị được) cho client-side.

Web server: Web server được dùng để phục vụ cho các ứng dụng web, web server sử dụng nghị thức HTTP để giao tiếp với trình duyệt web ở phía client. Tất cả các yêu cầu từ phía client đối với ứng dụng web đều được web server nhận và thông dịch và sau đó gọi các chức năng của ứng dụng thông qua các giao tiếp mạng như MAPI, Winsock, namped pipe...

Application server: Đây là phần chương trình gọi các hàm xử lý GIS, gửi yêu cầu lấy dữ liệu đến Clearing House.

Data server: Data server là phần cơ bản của hầu hết các hệ thống thông tin với nhiệm vụ quản lý và điều khiển truy cập dữ liệu. Ban đầu, đa số GIS sử dụng File System để quản lý dữ liệu không gian và DBMS (Database Management System) để quản lý dữ liệu thuộc tính. Ngày nay có nhiều sản phẩm và giải pháp phần mềm thay thế để quản lý dữ liệu không gian và thuộc tính chung như SDE của ESRI, SpatialWare của MapInfo ... Nhìn chung các cơ sở dữ liệu sử dụng đều là các cơ sở dữ liệu quan hệ, và trong tương lai sẽ thay thế bằng cơ sở dữ liệu hướng đối tượng.

Clearing House: Clearing House được sử dụng để chứa các dữ liệu về không gian được quản lý bởi các data server. Clearing House đóng vai trò như một cuốn catalog, nó tìm kiếm trong catalog này các dữ liệu cần tìm.



Hình 3.4 - Kiến trúc WebGIS

Như vậy, có 2 chiến thuật xây dựng WebGis tương ứng với 2 kiểu triển khai, kiểu thứ nhất tập trung công việc chủ yếu cho phía server, kiểu kia ngược lại tập trung công việc cho phía client. Sự kết hợp 2 chiến thuật này hình thành nên kiểu thứ 3 là kết hợp.

3.2.3.3 Các chuẩn trao đổi WebGIS hiện nay

Tổ chức OGC đã đưa ra ba chuẩn dịch vụ truy cập thông tin địa lý mang tính chuẩn hóa cao là: Web Map Service (WMS), Web Feature Service (WFS) và Web Coverage Service (WCS). Ngoài ra còn có các chuẩn khác như GeoParser và GeoCoder. Trong đó, hai chuẩn WMS và WFS là hai chuẩn cơ bản được sử dụng rất nhiều nhằm cung cấp các dịch vụ biểu diễn các thông tin địa lý ra ảnh bản đồ và truy vấn các dữ liệu địa lý đó.

3.2.4 Giới thiệu GeoServer

GeoServer là một máy chủ mã nguồn mở với mục đích kết nối những thông tin địa lý có sẵn tới các Geoweb (trang Web địa lý) sử dụng chuẩn mở. Được bắt đầu bởi một tổ chức phi lợi nhuận có tên The Open Planning Project (TOPP), nhằm mục đích hỗ trợ việc xử lý thông tin không gian địa lý với chất lượng cao, đơn giản trong sử dụng, là phần mềm mã nguồn mở nhằm cung cấp và chia sẻ dữ liệu. Được kỳ vọng sẽ trở thành một phương thức đơn giản để kết nối những nguồn thông tin có sẵn từ Google Earth, NASA World Wind nhằm tạo ra các dịch vụ Webmap như Google Maps, Windows Live Local và Yahoo Maps.

Là một dự án mang tính cộng đồng, GeoServer được phát triển, kiểm thử và hỗ trợ bởi nhiều nhóm đối tượng và tổ chức khác nhau trên toàn thế giới. GeoServer là sự phối hợp các chuẩn hoạt động của Open Geospatial Consortium (OGC), dịch vụ bản đồ Web Map Service (WMS), Web Feature Service (WFS). GeoServer là thành phần nền tảng của Geospatial Web

3.2.4.1 Lịch sử phát triển

Dự án GeoServer được bắt đầu vào năm 2001 bởi The Open Planning Project (TOPP). Vào thời điểm đó, mọi Website bản đồ chỉ tập trung vào chức năng khởi tạo bản đồ, và không thể chia sẻ những dữ liệu đã được thực hiện trên bản đồ. TOPP nhận ra rằng những dữ liệu này tương đương với 'mã nguồn' của bản đồ, và nó vô cùng quan trọng trong việc tạo ra cấu trúc dữ liệu không gian mở, để người dùng có thể phân tích và mô hình hóa, việc không thể đổi với dữ liệu ảnh.

Ngay sau khi những đặc tính kỹ thuật của Web Feature Server (WFS) được đưa ra trong bản phác thảo của Open Geospatial Consortium (OGC), kế thừa giao thức chuẩn được phát triển bởi TOPP trong việc tạo ra những kiến trúc không gian thông tin mở, GeoServer trở thành phần mềm mã nguồn mở sớm nhất cung cấp các đặc điểm kỹ thuật trong việc tạo và chỉnh sửa dữ liệu không gian.

Vào năm 2003, GeoServer được bổ sung WFS 1.0, và bổ sung WFS 1.1 vào năm 2006. Cộng đồng phát triển GeoServer đưa thêm WMS 1.1.1 và WCS 1.0. Đó

là những thành phần đưa GeoServer trở thành chuẩn của máy chủ không gian địa lý nguồn mở.

Những đóng góp khác bao gồm Giao diện quản trị Web và hỗ trợ nhiều định dạng dữ liệu xuất ra. Như một GeoWeb mở rộng, GeoServer luôn không ngừng phát triển, từng bước hỗ trợ: Google Earth, NASA World Wind, Google Maps, Windows Live Local và Yahoo Maps trong các dịch vụ desktop truyền thống và trong nền tảng GIS.

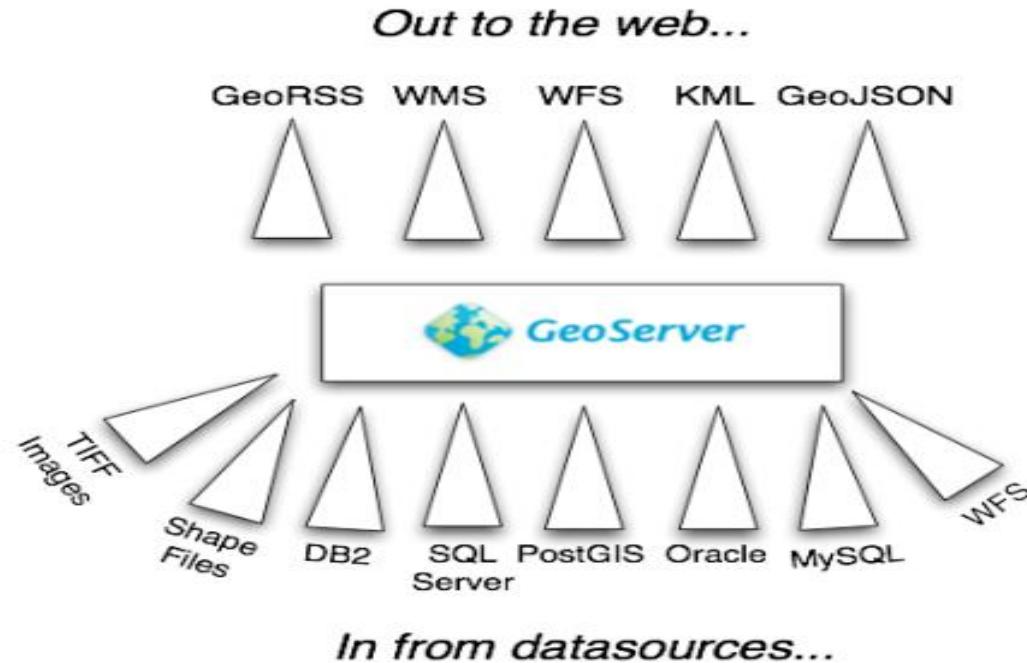
3.2.4.2 Chuẩn mở và khả năng chia sẻ dữ liệu không gian

Geoserver cho phép người dùng hiển thị thông tin không gian của mình về thế giới. Cung cấp chuẩn dịch vụ bản đồ (Web Map Service - WMS), GeoServer có thể tạo bản đồ và xuất ra nhiều định dạng, hỗ trợ rất nhiều style bản đồ. Tương thích với chuẩn Web Feature Service (WFS), GeoServer cho phép chia sẻ và chỉnh sửa dữ liệu đang được dùng để hiển thị bản đồ.

3.2.4.3 Các đặc trưng của Geoserver

GeoServer cho phép xuất dữ liệu linh hoạt dựa vào việc hỗ trợ các chuẩn KML, GML, Shapefile, GeoRSS, Portable Document Format, GeoJSON, JPEG, GIF, SVG, PNG....

GeoServer có thể truy xuất được nhiều định dạng dữ liệu, bao gồm PostGIS, Oracle Spatial, ArcSDE, DB2, MySQL, Shapefiles, GeoTIFF, GTOPO30 và nhiều loại khác. Bên cạnh đó, GeoServer còn có thể chỉnh sửa dữ liệu nhờ những thành phần xử lý của chuẩn Web Feature Server. GeoServer được xây dựng trong bộ GeoTools, được viết bởi ngôn ngữ Java.



Hình 3.5 – các định dạng dữ liệu được GeoServer hỗ trợ

MapServer cũng được coi là một sản phẩm mã nguồn mở, có nhiều chức năng tương tự. Sự khác biệt giữa hai sản phẩm là Mapserver được phát triển trong môi trường cũ, được viết bởi ngôn ngữ C và hoạt động nhờ CGI, còn GeoServer được viết bằng ngôn ngữ Java. GeoServer phân biệt với MapServer bởi nó có giao diện đồ họa, giúp đơn giản hơn trong việc cấu hình, và thực thi chức năng sửa đổi dựa vào Web Feature Server, cho phép chỉnh sửa thông tin không gian cả trên Web cũng như trên máy trạm Desktop. Ưu điểm lớn nhất của MapServer là tốc độ thực thi nhanh hơn GeoServer, nhưng từ phiên bản 1.6 trở đi của GeoServer thì tốc độ của hai sản phẩm đã tương đương.

3.2.4.4 Styled Layer Descriptor trong GeoServer

Về cơ bản, bản chất dữ liệu không gian không phải là thành phần trực quan, sinh động. Để nhìn thấy dữ liệu thì nó cần phải được định dạng kiểu dáng (style). Điều này có nghĩa là phải chỉ định màu sắc, độ rộng và các thuộc tính trực quan khác cho nó. Trong GeoServer, việc định kiểu này được thực hiện bằng cách sử dụng một ngôn ngữ được gọi là Styled Layer Descriptor (SLD). SLD là một ngôn ngữ dựa trên XML. Phần này sẽ giới thiệu về những gì có thể làm với SLD và cũng như GeoServer xử lý nó như thế nào.

Dữ liệu mà GeoServer có thể hỗ trợ thường là các hình dạng: điểm (points), đường (lines) và đa giác (polygons). Đường là dạng đơn giản nhất, chúng chỉ có một đường để định dạng kiểu (thuộc tính được biết như “stroke”). Đa giác, có một đường và một vùng bên trong (như thuộc tính “fill”), cả hai có thể được định dạng khác nhau. Điểm, có cả đường viền và vùng bên trong. Để lấp đầy vùng thì màu sắc có thể được quy định; cho nét, màu và độ dày cũng có thể được chỉ định.

Những kiểu định dạng nâng cao có thể không chỉ là màu sắc và độ dày. Chẳng hạn, điểm có thể được chỉ định với các hình dạng được hỗ trợ sẵn, như hình tròn, hình vuông, hình sao và thậm chí là các hình ảnh đồ họa hoặc văn bản. Đường có thể được định dạng là kiểu gạch ngang đứt khúc. Đa giác có thể được lấp đầy bằng một màu đồ họa hoặc hình ảnh tùy chọn. Ngoài ra, việc định kiểu dáng cho đối tượng không gian còn có thể dựa trên dữ liệu thuộc tính của nó thông qua bộ lọc trong GeoServer. Do đó, các đối tượng cùng loại có thể có các kiểu dáng khác nhau tùy thuộc vào giá trị thuộc tính của nó. Bên cạnh đó cũng có thể gán các nhãn (label) cho các đối tượng không gian. Các đối tượng cũng có thể được định kiểu tùy theo tỷ lệ phóng to hay thu nhỏ của bản đồ. Chẳng hạn, khi ở tỷ lệ lớn thì kích thước của đối tượng sẽ lớn hơn và ngược lại khi ở tỷ lệ nhỏ thì kích thước các đối tượng sẽ được thu nhỏ lại hoặc có thể không được hiển thị.

Mỗi lớp (featuretype) được đăng ký với GeoServer cần phải có ít nhất một SLD được kết hợp với nó. Mặc định GeoServer có sẵn một số SLD để kết hợp với các lớp dữ liệu chưa được định nghĩa kiểu, khi đó người dùng có thể tự định nghĩa các SLD cho mỗi lớp riêng biệt. Các tài liệu này được sử dụng để thay thế các tài liệu mặc định. Việc thêm các kiểu tự định nghĩa này được thực hiện thông qua giao diện quản trị của GeoServer. Dưới đây là một ví dụ đơn giản minh họa cho việc định dạng kiểu dáng cho một lớp dữ liệu có kiểu *point*:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0" ... >
  <NamedLayer>
    <Name>Simple point</Name>
    <UserStyle>
      <Title>Simple point</Title>
      <FeatureTypeStyle>
        <Rule>
          <PointSymbolizer>
            <Graphic>
              <Mark>
                <WellKnownName>star</WellKnownName>
              <Fill>
                <CssParameter name="fill">#FF0000</CssParameter>
              </Fill>
            </Mark>
            <Size>10</Size>
          </Graphic>
        </PointSymbolizer>
      </Rule>
    </FeatureTypeStyle>
  </UserStyle>
</NamedLayer>
</StyledLayerDescriptor>
```

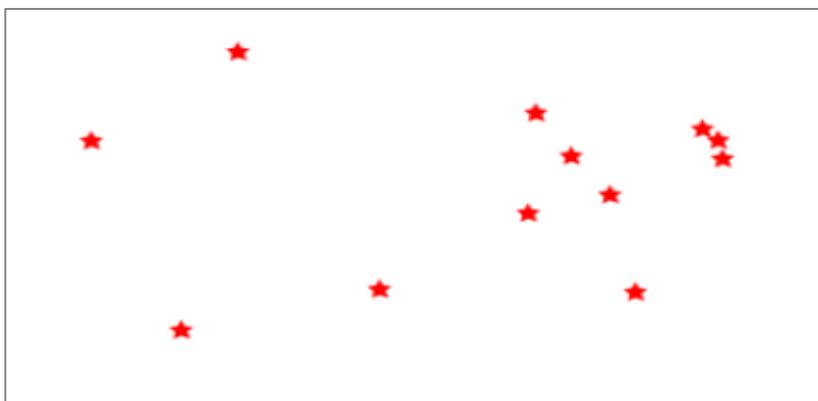
Trong tài liệu SLD trên cần quan tâm các thẻ sau:

- Thẻ “PointSymbolizer” được sử dụng để bắt đầu định kiểu dáng cho dữ liệu dạng điểm.
- Thẻ “WellKnownName” cho biết việc sử dụng các hình dạng được hỗ trợ sẵn để định kiểu cho dữ liệu, trong trường hợp này là hình sao (star) sẽ đại diện

cho một điểm. Ngoài ra, còn có nhiều hình dạng khác như hình vuông (square), hình tròn (circle), hình tam giác (triangle),...

- Thẻ “Fill” và “Size” dùng để tô màu và chỉ định kích thước cho điểm.

Kết quả:



Hình 3.6- Kết quả định dạng cho lớp kiểu điểm bằng SLD

3.2.4.5 Bộ lọc (Filtering) trong GeoServer

Bộ lọc trong GeoServer cho phép xác định những đối tượng thỏa mãn một tập các điều kiện được chỉ định trước. Cái này thường được sử dụng để giảm số lượng dữ liệu được trả về bởi các dịch vụ truy xuất dữ liệu trên các lớp bản đồ như dịch vụ WFS. Bộ lọc trong GeoServer được xây dựng dựa trên các đặc tả của OGC. Ngoài ra, bộ lọc trong GeoServer còn được sử dụng trong các SLD, điều này làm tăng tính linh động trong việc định dạng kiểu dáng cho dữ liệu không gian. Một ví dụ sử dụng bộ lọc trong tài liệu sau dùng để định dạng màu nền và độ rộng của những đường đi có thuộc tính loại đường (*loai_duong*) là tĩnh lô (*tinh_lo*).

```
<FeatureTypeStyle>
  <Rule>
    <Title>Tinh lo</Title>
    <ogc:Filter>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>loai_duong</ogc:PropertyName>
        <ogc:Literal>tinh lo</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:Filter>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#FFFA51</CssParameter>
        <CssParameter name="stroke-width">13</CssParameter>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

3.2.5 Thư viện Openlayers

Hiện nay, có nhiều thư viện hỗ trợ cho việc tương tác với bản đồ trên nền web, đại diện trong số đó là Mscross, Mapbuilder và OpenLayers. Các thư viện này hỗ trợ tốt việc hiện thị dữ liệu raster và vector. Tuy nhiên, do hỗ trợ nhiều điều khiển (controls) có khả năng tương tác linh hoạt, hỗ trợ hiển thị nhiều định dạng dữ liệu như WKT, GeoJSON, GML và hiện thị tốt trên nhiều trình duyệt nên OpenLayers sẽ được sử dụng trong đề tài để hiện thị bản đồ và tương tác với người dùng.

Openlayers là một thư viện thuần JavaScript hỗ trợ cho việc hiển thị dữ liệu bản đồ trên hầu hết các trình duyệt web mà không phụ thuộc vào phía server. OpenLayers cài đặt một tập các hàm API JavaScript hỗ trợ việc xây dựng các ứng dụng bản đồ trên nền web, cũng tương tự như các hàm API của Google Maps và MSN Virtual Earth, nhưng với sự khác biệt quan trọng – OpenLayers là phần mềm mã nguồn mở, được phát triển bởi cộng đồng phần mềm mã nguồn mở.

Có hai khái niệm quan trọng cần được hiểu rõ khi làm việc với OpenLayers để xây dựng bản đồ: ‘Map’, và ‘Layer’. ‘Map’ là nơi để lưu giữ các thông tin về hệ quy chiếu, đơn vị, vùng bao (box) ... của bản đồ. Bên trong bản đồ, dữ liệu được

hiển thị thông qua các ‘Layer’. Một ‘Layer’ là một nguồn dữ liệu, nó cho biết OpenLayers nên yêu cầu và hiển thị dữ liệu như thế nào. Trong một ‘Map’ có thể chứa một hoặc nhiều ‘Layer’ chồng lên nhau. Dữ liệu trong một ‘Layer’ có thể là dạng raster hoặc vector.

Ngoài ra, việc cung cấp một tập các điều khiển sẽ cho phép người phát triển có thể thao tác trên bản đồ một cách linh hoạt. Các điều khiển là các lớp trong OpenLayers, chúng cho phép định nghĩa các trạng thái, hành vi của bản đồ. Hoặc hiển thị các thông tin bổ sung cho người dùng. Các điều khiển chính là giao diện cho phép tương tác với bản đồ. Các điều khiển thường được sử dụng trên bản đồ, khi nó được khởi tạo:

- *Attribution*: điều khiển Attribution sẽ hiển thị quyền sở hữu được thiết lập trên mỗi lớp ở bên dưới góc phải trong bản đồ. Kiểu dáng và vị trí của điều khiển này có thể được chạy đè bằng lớp CSS ‘olControlAttribution’.
- *Navigation*: Cho phép người dùng dịch chuyển bản đồ theo các hướng khác nhau, cũng như cho phép phóng to/ thu nhỏ bản đồ. Điều khiển này là sự kết hợp của các điều khiển khác như: DragPan, ZoomBox.
- *DragFeature*: Cho phép dịch chuyển các đối tượng trên một lớp vector bằng cách nhấn giữ chuột trái trên đối tượng đó và di chuyển.
- *DrawFeature*: Cho phép vẽ các đối tượng điểm, đường, và đa giác trên một lớp vector khi nó được kích hoạt.
- *OverviewMap*: Tạo một bản đồ thu nhỏ của bản đồ chính, cho phép người dùng điều khiển phương hướng bản đồ được dễ dàng hơn khi nó được phóng to. Mặc định thì điều khiển này nằm ở phía dưới bên trái bản đồ.
- *ScaleLine*: Hiện thị tỷ lệ hiện tại của bản đồ. Mặc định được đặt ở góc dưới bên phải bản đồ.
- *SelectFeature*: Cho phép chọn những đối tượng trên một lớp vector. Điều khiển này có thể cho phép chọn nhiều đối tượng cùng một lúc bằng cách nhấn

giữ phím ‘Shift’ hoặc ‘Alt’ trong khi chọn đối tượng, mặt khác cũng có thể tạo một vùng bao trên tất cả các đối tượng muốn chọn.

Việc thêm một điều khiển vào một bản đồ được thực hiện tương đối đơn giản, ví dụ sau sẽ thêm điều khiển *OverviewMap* vào bản đồ.

```
var map = new OpenLayers.Map('map');
map.addControl(new OpenLayers.Control.OverviewMap());
```

Dòng đầu cho phép tạo ra một đối tượng ‘map’ với các điều khiển mặc định (Attribution , Navigation,...). Đối số của hàm xây dựng này là một phần tử *div* với giá trị thuộc tính *id* của thẻ *div* này là *map*. Dòng thứ hai chỉ ra cách để thêm một điều khiển *OverviewMap* vào bản đồ.

Bên cạnh đó, OpenLayers cũng cung cấp các lớp cho phép định dạng kiểu dáng các đối tượng không gian. Việc định dạng kiểu dáng này có thể áp dụng cho từng đối tượng hoặc toàn bộ các đối tượng vector trên một lớp. Các thuộc tính về kiểu dáng của đối tượng vector được định nghĩa thông qua đối tượng ‘style’. Đoạn mã sau minh họa cho việc định kiểu dáng cho các đối tượng kiểu điểm và đường.

```
var sketchSymbolizers = {
    "Point": {
        pointRadius : 4,
        graphicName : "circle",
        fillColor : "white",
        fillOpacity : 1,
        strokeWidth : 1,
        strokeOpacity : 1,
        strokeColor : "#333333" },
    "Line": {
        strokeWidth : 3,
        strokeOpacity : 1,
        strokeColor : "#51A9FF",
        strokeDashstyle : "dash" }};
var style = new OpenLayers.Style();
style.addRules([ new OpenLayers.Rule({
    symbolizer : sketchSymbolizers
}) ]);
```

Tuy nhiên, việc định dạng kiểu dáng này chỉ áp dụng được cho các đối tượng thuộc các lớp vector.

3.3 KIẾN TRÚC HƯỚNG DỊCH VỤ VÀ DỊCH VỤ WEB

3.3.1 Kiến trúc hướng dịch vụ

Kiến trúc hướng dịch vụ (SOA - Service Oriented Architecture) là một tập hợp linh hoạt các nguyên tắc được sử dụng trong suốt giai đoạn phát triển hệ thống. Một hệ thống dựa trên SOA sẽ đóng gói các chức năng như một bộ các dịch vụ (services) có khả năng tương tác, các dịch vụ này có thể được sử dụng trong nhiều hệ thống tách biệt nhau.

Dịch vụ được xem là yếu tố then chốt trong SOA. Có thể hiểu dịch vụ như là hàm chức năng (mô-đun phần mềm) thực hiện qui trình nghiệp vụ nào đó. Các dịch vụ trong SOA có kết nối mềm dẻo với nhau (nghĩa là một ứng dụng có thể ‘nói chuyện’ với một ứng dụng khác mà không cần biết các chi tiết kỹ thuật bên trong), có giao tiếp (dùng để gọi hàm dịch vụ) được định nghĩa rõ ràng và độc lập với nền tảng hệ thống, và có thể tái sử dụng. SOA là cấp độ cao hơn của phát triển ứng dụng, chú trọng đến qui trình nghiệp vụ và dùng giao tiếp chuẩn để giúp che đi sự phức tạp kỹ thuật bên dưới.

Thật ra triết lý SOA không hoàn toàn mới, DCOM và CORBA cũng có kiến trúc tương tự. Tuy nhiên, các kiến trúc cũ ràng buộc các thành phần với nhau quá chặt chẽ, chẳng hạn như các ứng dụng phân tán muôn làm việc với nhau phải đạt được 'thỏa thuận' về chi tiết tập hàm API, một thay đổi mã lệnh trong thành phần COM sẽ yêu cầu những thay đổi tương ứng đối với mã lệnh truy cập thành phần COM này.

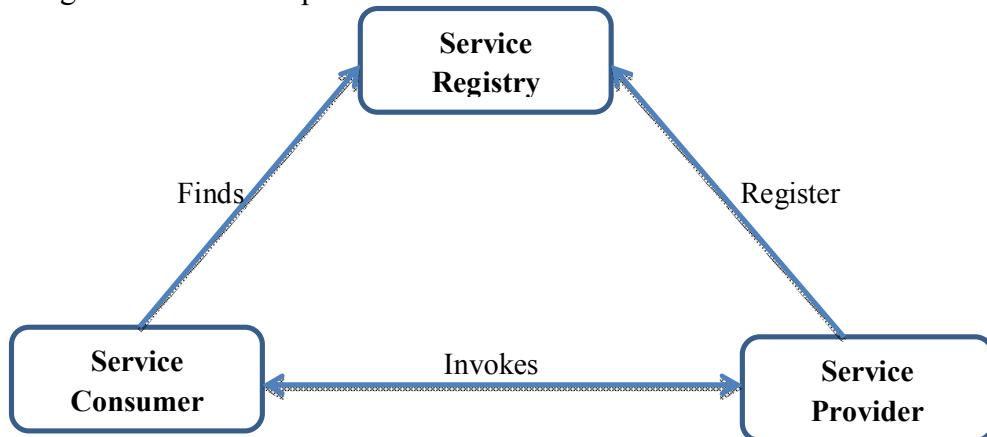
Ưu điểm chính của SOA là khả năng kết nối mềm dẻo và tái sử dụng. Thiết kế SOA sẽ tách riêng phần thực hiện dịch vụ (phần mềm) với giao tiếp gọi dịch vụ. Điều này tạo nên một giao tiếp nhất quán cho ứng dụng khách (client) sử dụng dịch vụ bất chấp công nghệ thực hiện dịch vụ. Thay vì xây dựng các ứng dụng đơn lẻ và đồ sộ, nhà phát triển sẽ xây dựng các dịch vụ tinh gọn có thể triển khai và tái sử dụng trong toàn bộ quy trình nghiệp vụ. Điều này cho phép tái sử dụng phần mềm

tốt hơn, cũng như tăng sự linh hoạt vì nhà phát triển có thể cài tiến dịch vụ mà không làm ảnh hưởng đến ứng dụng client sử dụng dịch vụ.

SOA dựa trên 2 nguyên tắc thiết kế quan trọng:

- Mô đun (module): Tách vấn đề lớn thành nhiều vấn đề nhỏ.
- Đóng gói: Che đi dữ liệu và logic trong từng mô đun (hay ‘hộp đen’) đối với truy cập từ ngoài.

Trong SOA có 3 thành phần chính:



Hình 3.7- Sự cộng tác trong một kiến trúc hướng dịch vụ.

Nhà cung cấp dịch vụ (service provider) cần cung cấp thông tin về dịch vụ của mình cho một dịch vụ lưu trữ thông tin dịch vụ (service registry). *Người sử dụng* dịch vụ (service consumer) thông qua service registry để tìm kiếm thông tin mô tả về dịch vụ cần tìm và sau đó là xây dựng kênh giao tiếp với phía nhà cung cấp.

SOA cung cấp giải pháp để giải quyết các vấn đề tồn tại của các hệ thống hiện nay như: phức tạp, không linh hoạt và không ổn định. Một hệ thống triển khai theo mô hình SOA có khả năng dễ mở rộng, liên kết tốt. Đây chính là cơ sở và nền tảng cho việc tích hợp, tái sử dụng lại những tài nguyên hiện có.

3.3.2 Dịch vụ web

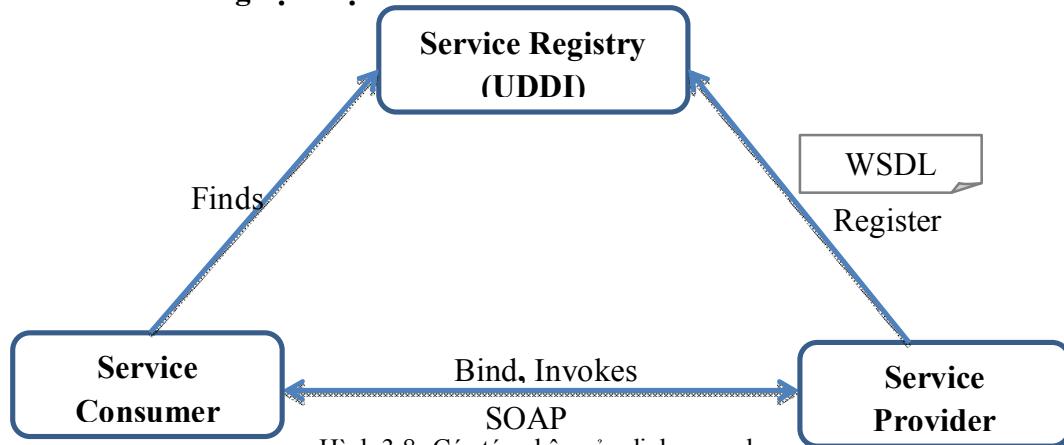
Dịch vụ web (Web Service) là một công nghệ được sử dụng rộng rãi để hiện thực mô hình SOA, nó cung cấp cơ chế cho phép trao đổi, tương tác giữa các ứng dụng với nhau dựa trên nền tảng XML. Dịch vụ web sử dụng giao thức web chuẩn để giao tiếp với nhau. Điều này làm cho dịch vụ web có thể tương tác với các dịch

vụ web khác dễ dàng tạo nên sự đồng nhất giữa các môi trường không đồng nhất. Giá trị cơ bản của dịch vụ web dựa trên việc cung cấp các phương thức theo chuẩn trong việc truy nhập đối với hệ thống đóng gói. Các phần mềm được viết bởi những ngôn ngữ lập trình khác nhau và chạy trên những nền tảng khác nhau có thể sử dụng dịch vụ web để chuyển đổi dữ liệu thông qua mạng Internet theo cách giao tiếp tương tự bên trong một máy tính. Tuy nhiên, công nghệ xây dựng dịch vụ web không nhất thiết phải là các công nghệ mới, nó có thể kết hợp với các công nghệ đã có như XML, SOAP, WSDL, UDDI... Với sự phát triển và lớn mạnh của Internet, dịch vụ web thật sự là một công nghệ đáng được quan tâm để giảm chi phí và độ phức tạp trong tích hợp và phát triển hệ thống.

▪ Đặc điểm của dịch vụ web

- Dịch vụ web cho phép các client và server trong các môi trường khác nhau có thể tương tác được với nhau. Ví dụ, khi đặt web server cho ứng dụng trên một máy chủ chạy hệ điều hành Linux, trong khi người dùng sử dụng máy tính chạy hệ điều hành Windows, ứng dụng vẫn có thể xử lý bình thường mà không cần có các yêu cầu đặc biệt để tương thích giữa hai hệ điều hành này.
- Phần lớn các kỹ thuật dịch vụ web được phát triển từ các chuẩn đã được công nhận, ví dụ như XML.
- Một dịch vụ web thường bao gồm nhiều mô đun và có thể công bố lên mạng Internet.

▪ Các tác nhân trong dịch vụ web



Hình 3.8- Các tác nhân của dịch vụ web.

- *Nhà cung cấp dịch vụ*: dùng ngôn ngữ định nghĩa dịch vụ web (Web Service Description Language - WSDL) để mô tả dịch vụ mà mình có thể cung cấp cho *nha dang ky dich vu*.
- *Nhà đăng ký dịch vụ*: lưu thông tin các dịch vụ được cung cấp bởi nhà cung cấp dịch vụ. Thành phần chính của nhà đăng ký dịch vụ là kho tích hợp đặc tả (Universal Discovery, Description and Integrate Repositories - UDDI).
- *Khách hàng dịch vụ*: là người sử dụng dịch web. Bằng cách sử dụng WSDL để đặc tả nhu cầu sử dụng và gửi cho *nha dang ky dich vu*. Bằng việc sử dụng UDDI và chức năng tìm kiếm của *nha dang ky dich vu*, khách hàng có thể tìm thấy *nha cung cap dich vu* thích hợp. Ngay sau đó khách hàng dịch vụ thiết lập một kênh giao tiếp với *nha cung cap dich vu* bằng cách sử dụng giao thức truy cập đối tượng (Simple Object Access Protocol - SOAP) để truy xuất dịch vụ.
- **Hoạt động của dịch vụ web**: đầu tiên nhà cung cấp dịch vụ sẽ công bố dịch vụ của mình lên nhà đăng ký dịch vụ. Khi có nhu cầu khách hàng dịch vụ sẽ tìm kiếm các dịch vụ trên nhà đăng ký dịch vụ. Kết quả trả về là một dịch vụ web mà sau đó khách hàng sẽ kết nối đến để thực hiện gọi các chức năng.

3.3.2.1 Web Service Description Language (WSDL)

WSDL là một ngôn ngữ mô tả web service dựa trên nền tảng XML. WSDL sử dụng các thẻ để định nghĩa các thuộc tính của dịch vụ web. Nó giúp định nghĩa cho một dịch vụ web và cho phép đặc tả cách mà một dịch vụ web có thể được truy cập. WSDL được sử dụng để mô tả cách mà một dịch vụ web phải làm việc như thế nào.

Các thuật ngữ trong WSDL:

Types: kiểu dữ liệu được sử dụng bởi dịch vụ web

Message: thông điệp được sử dụng bởi dịch vụ web

PortType: những thao tác (operations) được thực thi bởi dịch vụ web

Binding: giao thức giao tiếp được sử dụng bởi dịch vụ web

3.3.2.2 Simple Object Access Protocol (SOAP)

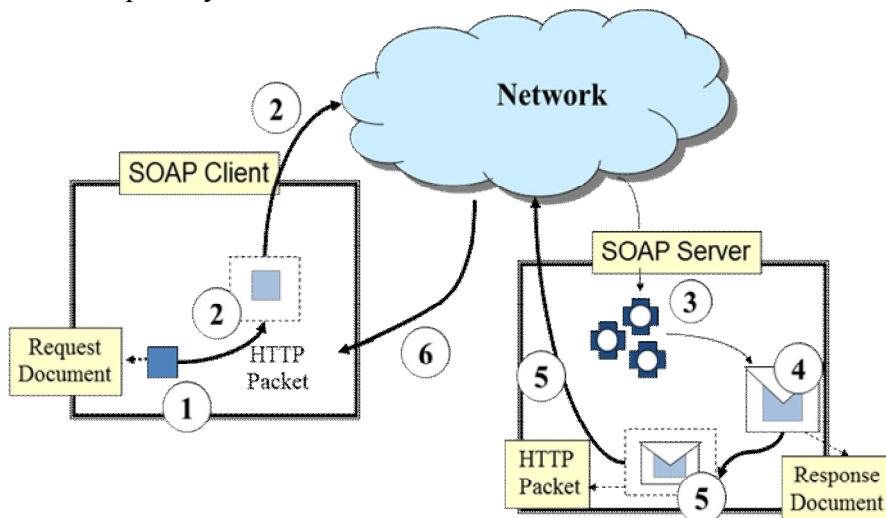
Theo W3C, SOAP là một giao thức hạng nhẹ (lightweight protocol) cho việc trao đổi thông tin trong một môi trường phân tán. Là một giao thức dựa trên XML,

nó cho phép các thành phần và ứng dụng giao tiếp với nhau thông qua giao thức HTTP. SOAP cung cấp một cách thức giao tiếp giữa những ứng dụng chạy trên những môi trường, công nghệ và ngôn ngữ lập trình khác nhau. Mặc dù SOAP thường được sử dụng với HTTP, nhưng nó cũng có thể được sử dụng với những giao thức khác như SMTP.

3.3.2.3 Quá trình xử lý của một SOAP

Quá trình xử lý của một SOAP có thể mô tả như sau:

1. Khi có nhu cầu SOAP client sẽ tạo ra một yêu cầu bao gồm các tham số cần thiết để gọi một phương thức trên SOAP server
2. Yêu cầu này sẽ được gói vào trong một HTTP packet và được gửi đến SOAP server thông qua mạng
3. SOAP server sẽ nhận các yêu cầu này sau đó phân tích và xử lý yêu cầu
4. Kết quả xử lý sẽ được gói vào trong một SOAP Envelope và được xem như kết quả trả về
5. Kết quả này sẽ được gói vào HTTP packet và truyền về client. Client nhận và phân tích kết quả này

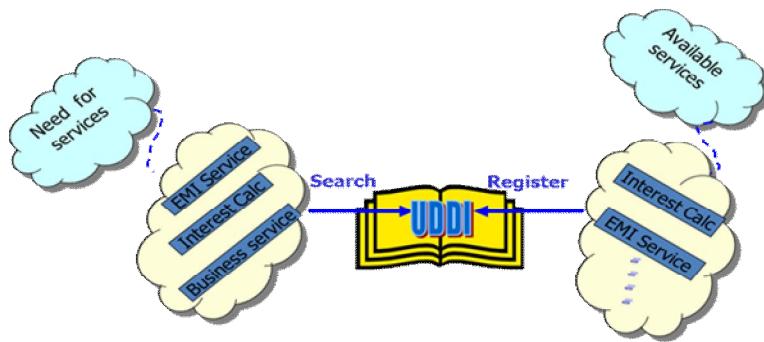


Hình 3.9- Quá trình xử lý của SOAP

3.3.2.4 Universal Description, Discovery and Integration (UDDI)

UDDI là một dịch vụ lưu trữ và tìm kiếm các thông tin về dịch vụ web. Nó được xem là cầu nối giữa nhà cung cấp dịch vụ và người sử dụng dịch vụ. Đôi với

nha cung cấp dịch vụ thì nó là nơi để họ đăng ký và quảng bá các thông tin về dịch vụ web của họ. Đối với người sử dụng dịch vụ, đây là nơi mà họ có thể tìm thấy các dịch vụ mà họ cần, chọn lọc các dịch vụ để tìm các dịch vụ đáp ứng nhu cầu của họ.



Hình 3.10- Đăng ký và sử dụng các dịch vụ

3.4 HIỆN TRẠNG ỨNG DỤNG CÔNG NGHỆ GIS TẠI VIỆT NAM

Công nghệ thông tin phát triển rất mạnh mẽ, nhiều kỹ thuật tiên tiến trước đây chỉ có ở các nước phát triển thì hiện nay đã có mặt tại Việt Nam. Đối với sản phẩm công nghệ thông tin chuyên nghiệp như hệ thống thông tin địa lý, tuy là các nhà cung cấp đang cố gắng đưa ra các sản phẩm dễ dùng, tiện lợi nhưng thực chất vẫn có khoảng cách giữa các nhà chuyên môn và các sản phẩm công nghệ nói trên. Ở Việt Nam hệ thống thông tin địa lý ứng dụng chủ yếu cho các lĩnh vực số hóa. Sau giai đoạn số hóa người ta cần đến hệ thống thông tin địa lý ở các chức năng phân tích khác để giải quyết cho bài toán ứng dụng.

Nhiều công ty nước ngoài đã xâm nhập vào thị trường Việt Nam trong lĩnh vực này. Tuy nhiên giá các sản phẩm phần mềm của họ rất cao tạo một khoảng cách ngày càng xa cho việc ứng dụng GIS vào cuộc sống hàng ngày.

Công nghệ GIS với những lợi thế mà nó mang lại phương pháp quản lý hiệu quả hơn, mọi sự vật, đối tượng từ những thông tin không gian đến những thông tin phi không gian tất cả đều được quản lý thống nhất trên cùng hệ thống. Mọi truy xuất đều thể hiện trực quan hơn trên bản đồ giấy thay cho những dòng văn bản đơn thuần. Chính vì thế GIS ngày càng trở nên quen thuộc hơn cho người dùng, nó được

ứng dụng trong nhiều lĩnh vực từ đơn giản đến phức tạp và chi phí đòi hỏi đầu tư ngày một thấp hơn. Có thể nói rằng GIS ngày một tự khẳng định tầm quan trọng, được các nước phát triển xem như một mũi nhọn trong lĩnh vực công nghệ thông tin và được đưa vào chương trình giảng dạy.

Chương 4 - CHUẨN OPENGIS

4.1 TỔ CHỨC OPENGIS OGC

OGC – Open Geospatial Consortium là một tổ chức xây dựng các chuẩn với tính chất đồng tâm, tự nguyện, có tính toàn cầu và phi lợi nhuận. OGC dẫn dắt việc phát triển các chuẩn cho các dịch vụ trên cơ sở vị trí và không gian địa lý. OGC hoạt động với chính quyền, các nhà công nghiệp GIS và các Viện nghiên cứu để tạo ra các giao tiếp ứng dụng mở cho các hệ thống thông tin địa lý và các công nghệ chính yếu khác có liên quan.

Ngày nay, OGC là một tổ chức quốc tế của 419 công ty, các tổ chức chính phủ và các trường đại học tham gia trong quá trình tìm tiếng nói chung để phát triển các đặc tả giao tiếp cho cộng đồng. Chúng ta thường gọi đó là các đặc tả OpenGIS (OpenGIS Specifications). Các đặc tả OpenGIS hỗ trợ các giải pháp đồng vận hành, tích hợp làm cho dữ liệu địa lý luôn sẵn sàng phục vụ cho Web, các dịch vụ trên nền tảng định vị, các dịch vụ không dây và phù hợp với các xu hướng chính của Công nghệ Thông tin. Các đặc tả sẽ tăng cường sức mạnh cho các nhà phát triển công nghệ nhằm biến các dịch vụ và thông tin không gian phức tạp trở nên dễ dàng truy cập và hữu ích với hầu hết các loại ứng dụng. Trong phạm vi đề tài này chúng em sẽ giới thiệu 2 đặc tả của OCG đó là Styled Layer Descriptor Implementation Specification (Phụ lục A) và OpenGIS Abstract Specification for SQL (Phụ lục B).

Số thành viên tham gia vào OGC càng tăng với mọi thành phần: tư nhân, nhà nước và các Trung tâm nghiên cứu dưới nhiều hình thức khác nhau. Đối với tổ chức Nhà nước Chính phủ thì việc tham gia vào OGC là cơ hội nắm bắt được các xu thế phát triển của công nghệ GIS đồng thời là nơi có thể nhận được các tư vấn, hỗ trợ cần thiết để hoạch định các chính sách phát triển GIS và các kinh nghiệm quý báu từ các dự án GIS trên toàn thế giới theo chuẩn OpenGIS nhằm thực thi hiệu quả khả năng đồng vận hành, tích hợp dữ liệu giữa các hệ thống khác nhau.

4.2 TỔNG QUAN VỀ OPENGIS

4.2.1 Các khái niệm

Chuẩn OpenGIS: viết tắt của Open Geodata Interoperability Specification

Dữ liệu địa lý “geodata”: Thông tin xác định vị trí địa lý và những nét tiêu biểu của các đặc tính tự nhiên hoặc nhân tạo và những ranh giới của trái đất. Geodata đại diện cho những khái niệm trừu tượng của những thực thể thế giới thực, những con đường, trường học, cơ quan, bệnh viện... là dữ liệu mô tả các hiện tượng một cách trực tiếp hay gián tiếp kết hợp với vị trí liên quan đến bề mặt trái đất đã được thu thập dưới dạng số.

Xử lý thông tin địa lý “geoprocessing”: là việc tính toán dạng kỹ thuật số sử dụng geodata, lập mô hình, phiên dịch và sử dụng thông tin trái đất bao gồm hệ thống thông tin địa lý, hệ thống thông tin đất đai, việc xử lý và tạo ảnh trái đất, việc chứa data trong tất cả các loại cơ sở dữ liệu, những phương thức khảo sát dạng kỹ thuật số, sự định hướng, khí tượng học, địa chấn học...

“Interoperable geoprocessing ”: chỉ ra khả năng của một hệ thống kỹ thuật số có thể:

- Trao đổi thông tin không gian về trái đất, về các đối tượng ở trên, bên trên và dưới bề mặt trái đất.
- Chạy các phần mềm có khả năng thao tác các thông tin như vậy qua mạng một cách hợp tác.

4.2.2 Đặc tả OpenGIS (OpenGIS Specification)

4.2.2.1 Khái niệm

Đặc tả OpenGIS là một đặc tả toàn diện của bộ khung phần mềm cho các truy cập phân tán đến geodata và những tài nguyên geoprocessing. Đặc tả này cung cấp cho các nhà phát triển phần mềm trên thế giới một khuôn mẫu giao tiếp chung cẩn kẽ để viết các phần mềm hoạt động chung với các phần mềm dạng OpenGIS khác.

OpenGIS là một hệ thống khung quy định chuẩn của phần mềm cho phép truy cập và xử lý thông tin địa lý Geodata, cho phép các nhà phát triển phần mềm xây dựng các phần mềm có sử dụng chung tài nguyên dữ liệu địa lý. Bộ khung OpenGIS gồm:

- Cách thức chung mô tả các hiện tượng, các đối tượng trên trái đất bằng toán học và các khái niệm toán học.
- Mô hình khung quy định khả năng truy cập, quản lý, thao tác, trình bày và chia sẻ dữ liệu địa lý (geodata) giữa cộng đồng Công nghệ thông tin.
- Mô hình khung quy định phương pháp sử dụng dữ liệu địa lý mở (Open Geodata) và các dịch vụ của OpenGIS về mặt kỹ thuật và tổ chức.

Các nhà phát triển xây dựng những hệ thống có giao tiếp thích ứng sẽ tạo ra những phần mềm và những ứng dụng có thể kiểm soát một phạm vi rộng các kiểu geodata và các hàm geoprocessing. Người sử dụng hệ thống này có thể chia sẻ một không gian dữ liệu tiềm năng rộng lớn qua mạng, dù dữ liệu được sản sinh tại các thời điểm khác nhau bởi các nhóm không liên quan sử dụng các hệ thống sản xuất khác nhau cho những mục đích khác nhau và thật sự có thể đang hiện hữu dưới sự điều khiển chính của hệ thống được sử dụng cho việc sản xuất của họ. Geodata được tổ chức trong các hệ thống có giao tiếp thích ứng OpenGIS Specification sẽ có thể được truy xuất bởi các phần mềm có giao tiếp thích ứng OpenGIS Specification khác.

OpenGIS Specification cung cấp một bộ khung cho những người phát triển phần mềm để tạo ra những phần mềm cho phép những người dùng của họ truy nhập vào xử lý dữ liệu địa lý từ những nguồn đa dạng qua một giao tiếp chung bên trong một nền tảng công nghệ thông tin mở.

4.2.2.2 Ưu điểm

Đối với nhà phát triển ứng dụng: có thể dễ dàng hơn trong việc xây dựng các phần mềm truy cập Geodata và tài nguyên Geoprocessing; sửa đổi các ứng dụng

cho phù hợp với nhu cầu cụ thể của người dùng. Có thể sử dụng lại mã nguồn của Geoprocessing.

Đối với nhà quản lý thông tin: Linh hoạt hơn trong việc phân phối hoặc truy cập dữ liệu Geodata, cung cấp khả năng của Geoprocessing tới tận khách hàng

Đối với người dùng cuối: là người được hưởng lợi tối ưu nhờ truy cập nhanh chóng và dễ dàng đến một hệ thống thông tin rộng lớn có nhiều ứng dụng và khai thác thông tin gắn với không gian địa lý chính xác.

4.2.2.3 Đặc điểm

OpenGIS Specification thiết lập một nền tảng công nghệ chung trên đó ngành công nghiệp phần mềm có thể xây dựng những thành phần phần mềm và ứng dụng geoprocessing có các đặc điểm sau:

- Interoperable (hoạt động liên hợp): OpenGIS Specification cung cấp những giao diện chuẩn đối với geodata và những dịch vụ geoprocessing. Những giao diện này hỗ trợ trong những hệ thống độc lập và các mạng: sự truy cập geodata khách/chủ phân tán, thao tác geoprocessing ngang hàng phân tán.
- Supportive of Information Communities (Sự khuyến khích của cộng đồng thông tin) - OpenGIS Specification tối ưu hóa việc chia sẻ dữ liệu trong một cộng đồng người dùng và những nhà sản xuất.
- Ubiquitous (Thường gặp) - OpenGIS Specification cung cấp những phương tiện cho tất cả các ứng dụng để khai thác những dịch vụ của OpenGIS qua những giao diện và giao thức chuẩn.
- Reliable (Tin cậy) – Geoprocessing phân tán yêu cầu ở một mức cao khả năng điều khiển và sự toàn vẹn.
- Easy to use (Dễ sử dụng) – Phần mềm theo nền OpenGIS Specification sẽ sử dụng những qui tắc và thủ tục đúng, logic cho việc sử dụng geodata và các dịch vụ geoprocessing.

- Portable (Khả chuyên) – OpenGIS Specification độc lập về môi trường phần mềm, nền tảng, phần cứng và mạng.
- Cooperative (Hợp tác) – OpenGIS Specification hỗ trợ tính toán dùng chung và tài nguyên dùng chung. Công nghệ OpenGIS có thể dễ dàng kết hợp với công nghệ thông tin khác.
- Scalable (Biến đổi được) – Phần mềm theo nền OpenGIS Specification thường gồm có những thành phần phần mềm geoprocessing có thể được cấu hình cho bất kỳ ứng dụng geoprocessing nào hoặc môi trường tính toán chuẩn, bất chấp kích thước cơ sở dữ liệu.
- Extensible (Dễ mở rộng) – OpenGIS Specification có thể đồng hóa giữa những phần mềm geoprocessing và kiểu geodata mới, và có thể điều khiển những công nghệ mới mà OpenGIS Specification phụ thuộc trên nó, như những nền tính toán phân tán khi chúng sẵn có.
- Compatible (Tương thích) – OpenGIS Specification giữ gìn sự đầu tư của những người dùng trong dữ liệu và phần mềm. Nó tương thích và không gối chồng lên nhau.
- Implementable (Có thể cài đặt) – Mục đích quan trọng nhất là các công nghệ được ghi rõ trong OpenGIS Specification phải có thể cài đặt được.

4.2.2.4 Phân loại

OpenGIS Specification được phát triển và phát hành thành nhiều thành phần. Nó bao gồm đặc tả trừu tượng (Abstract Specification) và đặc tả cài đặt (Implementation Specification).

OpenGIS Abstract Specification: Ủy ban kỹ thuật OGC đã phát triển một kiến trúc hỗ trợ tầm nhìn của mình về công nghệ không gian địa lý và khả năng tương tác dữ liệu được gọi là OpenGIS Abstract Specification. Abstract Specification cung cấp nền tảng về khái niệm cho hầu hết các hoạt động phát triển đặc tả OGC. Các giao thức và giao diện mở được xây dựng và tham chiếu với các Abstract Specification, vì thế cho phép khả năng tương tác giữa các bộ phận khác

nhau và các hệ thống xử lý không gian khác nhau. Mục đích của Abstract Specification là tạo ra một mô hình khái niệm đầy đủ, đủ để tạo ra các OpenGIS Implementation Specifications. OpenGIS Abstract Specification được chia thành những chủ đề sau:

- Topic 0 - Overview: Giới thiệu tổng quan về Abstract Specification
- Topic 1 - Feature Geometry: cung cấp những cấu trúc hình học cho các dạng đối tượng hình học
- Topic 2: Spatial referencing by coordinates: Cung cấp hệ thống tham chiếu không gian mà nhờ đó các đối tượng liên quan với các vị trí trên trái đất theo qui tắc đo đạc.
- Topic 3 - Locational Geometry Structures: cung cấp mô hình trừu tượng cho công nghệ được sử dụng rộng rãi trên nền GIS. Topic này hỗ trợ cho việc đặc tả các đối tượng hình học đơn giản và hệ qui chiếu không gian của chúng.
- Topic 4 - Stored Functions and Interpolation: cung cấp mô hình trừu tượng cho công nghệ được sử dụng rộng rãi trên nền GIS. Topic này cần thiết để hỗ trợ cho đặc tả miền bao phủ ở topic 6.
- Topic 5 - Features : Như ISO 19101, “Một đối tượng là một sự trừu tượng của một hiện tượng trong thế giới thực”. Một đối tượng là đối tượng địa lý nếu nó gắn liền với một vị trí tương đối trên trái đất.
- Topic 6 - Schema for coverage geometry and functions: Tiêu chuẩn này định nghĩa một lược đồ khái niệm về các đặc tính không gian của miền bao phủ.
- Topic 7 - Earth Imagery: cung cấp một mô hình tham khảo cho việc xử lý phân phối mở các hình ảnh địa lý.
- Topic 8 - Relationships Between Features: Giới thiệu trừu tượng về mối quan hệ của các thực thể trong thế giới thực. Sự trừu tượng này mô hình hóa các mối quan hệ giữa các đối tượng (feature) được giới thiệu trong topic 5.

- Topic 10 - Feature Collections: Một Feature Collections là một đối tượng trừu tượng bao gồm: Feature Instances, Feature Schema, và Project Schema.
- Topic 11 - Metadata: ISO 19115 đã thay thế cho OGC Abstract Specification topic 9 và 11.
- Topic 12 - The OpenGIS Service Architecture: Tiêu chuẩn này cung cấp một khuôn khổ cho các nhà phát triển để tạo ra phần mềm cho phép người dùng truy cập và xử lý dữ liệu địa lý từ nhiều nguồn khác nhau trên một giao diện tính toán chung trong một môi trường công nghệ thông tin mở.
- Topic 13 - Catalog Services: Bao gồm các dịch vụ truy cập thông tin địa lý
- Topic 14 - Semantics and Information Communities: cho phép các nhóm người như: nhà sinh thái, kỹ sư dân dụng quản lý hiệu quả về ngữ nghĩa các dữ liệu địa lý và giúp họ có được lợi ích tối đa từ các dữ liệu địa lý của nhau mặc dù khác biệt về ngữ nghĩa.
- Topic 15 - Image Exploitation Services: Mô tả các chuyên mục và phân loại các dịch vụ khai thác hình ảnh cần thiết để hỗ trợ việc sử dụng các hình ảnh và miền bao phủ liên quan.
- Topic 16 - Image Coordinate Transformation Services: Bao gồm các dịch vụ chuyển đổi tọa độ hình ảnh.
- Topic 17 - Location Based Mobile Services: bản nháp cho các dịch vụ dựa vào định vị. Chưa được thông qua.
- Topic 18 - Geospatial Digital Rights Management Reference Model (GeoDRM RM): tài liệu này là một mô hình tham khảo cho các chức năng quản lý quyền kỹ thuật số (DRM) cho nguồn tài nguyên không gian địa lý.
- Topic 20: Observations and Measurements: tiêu chuẩn này định nghĩa một lược đồ khái niệm cho việc quan sát, và cho các tính năng liên quan đến việc lấy mẫu khi quan sát.

- Topic Domain 1 - Telecommunications Domain: mô hình miền cho mạng viễn thông.

OpenGIS Implementation Specifications: Implementation Specifications thì khác hơn so với Abstract Specification. Nó được viết cho những người hướng về kỹ thuật nhiều hơn và chi tiết hơn về cấu trúc giao diện giữa các thành phần phần mềm. Một đặc tả giao diện được xem như là chi tiết nếu được triển khai bởi hai kỹ sư phần mềm hoàn toàn không biết nhau mà thành phần kết quả của mỗi người có thể tương tác với nhau bằng giao diện đó. Hiện tại có hơn 50 Implementation Specification được phát hành trên trang web của OGC.

Trong phần phụ lục của luận văn này chúng em sẽ trình bày nội dung của OpenGIS Abstract Specification Simple Features for SQL (đặc tả này định nghĩa lược đồ SQL hỗ trợ cho việc lưu trữ, truy xuất, truy vấn và cập nhật các đối tượng không gian thông qua ODBC API) và Styled Layer Descriptor Implementation Specification (đặc tả này như một chuẩn để định dạng kiểu cho việc thể hiện các đối tượng trên bản đồ)

4.3 WEB MAP SERVICE (WMS)

Web Map Service là một dịch vụ giúp tạo ra các bản đồ dựa trên các dữ liệu địa lý. Bản đồ ở đây được hiểu như một cách thể hiện trực quan của dữ liệu địa lý còn bản thân bản đồ không được xem là dữ liệu. Các bản đồ này được hiển thị dưới các định dạng ảnh như PNG, GIF, JPEG hoặc các định dạng thành phần đồ họa vector như SVG (Scalable Vector Graphics), WebCGM (Web Computer Graphics Metafile). Một WMS sẽ hỗ trợ ba thao tác (operation), trong đó hai thao tác đầu là bắt buộc cho mọi WMS:

- GetCapabilities (bắt buộc): Client nhận về một mô tả thông tin về WMS, các tham số được chấp nhận và hỗ trợ, bảng mô tả thường dưới dạng file XML.

Tham số request	Bắt buộc/ Tùy chọn	Mô tả
VERSION=version	Tùy chọn	Phiên bản yêu cầu
SERVICE=WMS	Bắt buộc	Loại dịch vụ
REQUEST=GetCapabilities	Bắt buộc	Tên yêu cầu
UPDATESEQUENCE=string	Tùy chọn	Dãy số/chuỗi để điều khiển cache

Bảng 4.1- Bảng mô tả các tham số trong GetCapabilities request URL

Ví dụ: <http://localhost:8080/geoserver/ows?service=wms&version=1.1.1&request=GetCapabilities>

- GetMap (bắt buộc): Client nhận về một ảnh bản đồ phù hợp với tham số mà client gửi lên server. Bảng sau mô tả các yêu cầu của GetMap.

Tham số request	Bắt buộc/ Tùy chọn	Mô tả
VERSION=version	Bắt buộc	Phiên bản yêu cầu.
REQUEST=GetMap	Bắt buộc	Tên yêu cầu.
LAYERS=layer_list	Bắt buộc	Danh sách các lớp bản đồ được phân cách bởi dấu phẩy.
STYLES=style_list	Bắt buộc	Danh sách kiểu của mỗi lớp trong request.
SRS=namespace:identifier	Bắt buộc	Hệ quy chiếu không gian.
BBOX=minx,miny,maxx,maxy	Bắt buộc	Góc dưới trái và trên phải của hình chữ nhật bao theo đơn vị của SRS.
WIDTH=output_width	Bắt buộc	Chiều rộng của ảnh (pixel)
HEIGHT=output_height	Bắt buộc	Chiều cao của ảnh (pixel)
FORMAT=output_format	Bắt buộc	Định dạng trả về của bản đồ
TRANSPARENT=TRUE FALSE	Tùy chọn	Độ trong suốt của nền bản đồ
BGCOLOR=color_value	Tùy chọn	Màu nền
EXCEPTIONS=exception_format	Tùy chọn	Định dạng ngoại lệ trả về của WMS
TIME=time	Tùy chọn	Giá trị thời gian của lớp
ELEVATION=elevation	Tùy chọn	Giá trị độ cao của lớp
Các chiều mẫu khác	Tùy chọn	Giá trị của các chiều

Bảng 4.2- Bảng mô tả các yêu cầu của GetMap

Ví dụ:

`http://localhost:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=ws_cantho:quanhuyen&styles=ct_quanhuyen&bbox=524847.24,1097030.63,592626.076,1142071.805&width=512&height=340&srs=EPSG:4326`

- GetFeatureInfo (không bắt buộc): Client hỏi thông tin về đặc điểm nào đó (của đối tượng) xuất hiện trên bản đồ. Trình duyệt web phía client có thể thông qua Web Map Service thực hiện các chức năng này bằng cách gửi các yêu cầu dưới dạng một URL. Nội dung của chuỗi URL này phụ thuộc vào công việc được yêu cầu (chỉ ra bởi tham số Request Type).

Tham số request	Bắt buộc / tùy chọn	Mô tả
VERSION=version	Bắt buộc	Phiên bản yêu cầu.
REQUEST=GetFeatureInfo	Bắt buộc	Tên yêu cầu.
<map_request_copy>	Bắt buộc	Phân copy lại các tham số của yêu cầu GetMap.
QUERY_LAYERS=layer_list	Bắt buộc	Danh sách các lớp cần truy vấn, được phân cách bởi dấu phẩy.
INFO_FORMAT=output_format	Tùy chọn	Định dạng thông tin của feature (kiểu MIME).
FEATURE_COUNT=number	Tùy chọn	Số lượng feature cần lấy thông tin (default = 1).
X=pixel_column	Bắt buộc	Hoành độ của điểm cần lấy thông tin.
Y= pixel_column	Bắt buộc	Tung độ của điểm cần lấy thông tin.
EXCEPTIONS=exception_format Tùy chọn	Tùy chọn	Tung độ của điểm cần lấy thông tin.
Vendor-specific parameters	Tùy chọn	Định dạng của biệt lê (default=application/vnd.ogc.se_xm)

Bảng 4.3- Bảng mô tả các yêu cầu của GetFeatureInfo.

Ví dụ:

`http://localhost:8080/geoserver/wms?bbox=524847.24,1097030.63,592626.076,1142071.805&styles=&format/jpeg&info_format=text/plain&request=GetFeatureInfo&layers=ws_cantho:quanhuyen&query_layers=ws_cantho:quanhuyen&width=512&height=340&x=100&y=100`

4.4 WEB FEATURE SERVICE (WFS)

WFS cung cấp các giao tiếp thông thường đến cơ sở dữ liệu GIS như là thêm, xóa, sửa, truy vấn, cập nhật các đối tượng địa lý. Nó sử dụng giao thức HTTP để tính toán và phân phối thông tin. WFS là một cách phân phối các đặc trưng địa lý thông qua một dịch vụ web đến với ứng dụng phía người dùng (client) hoặc một trình duyệt (browser). Người dùng (client) có thể yêu cầu dữ liệu có chọn lọc để phục vụ cho phạm vi nhu cầu của mình. WFS là một sự chuẩn hóa của việc phân phối dữ liệu vector đến đông đảo người dùng. Người dùng có thể nhập vào dữ liệu dạng vector khi yêu cầu thông tin và WFS sẽ đáp ứng yêu cầu đó. Đây là đại diện chuẩn điển hình của kỹ thuật web động.

Để hỗ trợ cho các giao tác thêm, xóa, sửa và truy vấn trên dữ liệu, WFS hỗ trợ các giao tác sau:

- GetCapabilities: WFS bắt buộc phải mô tả khả năng của nó qua giao tác này. Nó chỉ ra các kiểu đối tượng mà nó hỗ trợ và các giao tác được hỗ trợ trên mỗi kiểu feature.
- DescribeFeatureType: Khi có yêu cầu, WFS phải có khả năng mô tả cấu trúc của bất kỳ kiểu đối tượng nào mà nó hỗ trợ.
- GetFeature: WFS phải có khả năng đáp ứng các yêu cầu lấy và thể hiện các đối tượng. Client có thể chỉ rõ ra các thuộc tính nào của đối tượng mà nó muốn lấy kèm theo các câu tạo ràng buộc trên chúng.
- Transaction: WFS có khả năng hỗ trợ các yêu cầu thêm, xóa, sửa các đối tượng.
- LockFeature: WFS có khả năng đáp ứng các yêu cầu khóa các thể hiện thuộc tính trong khoảng thời gian thực hiện giao tác.

4.5 WEB COVERAGE SERVICE (WCS)

Đây là mở rộng của WMS cho phép truy cập đến các dữ liệu kiểu “phủ trùm” thể hiện các giá trị và thuộc tính của các vị trí không gian hơn là chỉ đơn thuần thể hiện bản đồ đã được tổng hợp (dạng hình ảnh, pictures). WCS cung cấp một giao

diện cho phép yêu cầu các vùng địa lý bao phủ trên web bằng cách sử dụng các nền tảng độc lập. Vùng bao phủ là những đối tượng hay hình ảnh trong một khu vực địa lý, trong khi các giao diện WMS hoặc các công thông tin bản đồ như Google Map chỉ trả kết quả là một hình ảnh mà người dùng cuối không thể chỉnh sửa hoặc phân tích không gian. Các thành viên trong OGC đã định nghĩa và bảo trì các đặc tả WCS. Geoserver có thể xem là một tham khảo cho việc cài đặt của chuẩn này.

Các WCS cho phép truy vấn và thu hồi các vùng bao phủ. Một WCS mô tả các hoạt động khám phá, truy vấn và chuyển đổi dữ liệu. Client sẽ tạo ra hàng loạt các yêu cầu và gửi nó tới một máy chủ web bằng cách sử dụng các tính năng HTTP. Các tính năng máy chủ web sau đó thực thi các yêu cầu. Các đặc điểm kỹ thuật WCS sử dụng HTTP như là nền tảng tính toán phân tán, mặc dù điều này không phải là một yêu cầu khó khăn.

Có hai kiểu mã hóa được định nghĩa cho các hoạt động WCS:

- XML (tuân theo chuẩn HTTP POST / SOAP)
- Từ khoá - cặp giá trị (tuân theo chuẩn HTTP GET / Gọi thủ tục từ xa)

Một số phần mềm hỗ trợ cho WCS:

- GeoServer - server (cung cấp WCS)
- APOLLO - server và client.
- GDAL - client (read).
- Geomatica Web Server Suite - client và server.
- GeoMedia - client (read WCS) và server (GeoMedia WebMap).
- GvSIG - client (read WCS).
- MapServer – server.
- ArcGIS Server - server và client.

Các phương thức được đặc tả trong WCS:

- GetCapabilities: WCS cho phép máy khách yêu cầu siêu dữ liệu. Phương thức này bắt buộc phải có nhằm đặc tả khả năng thực hiện của máy chủ, thường bao gồm mô tả hiện có trên máy chủ
- DescribeCoverage: phương thức bắt buộc nhằm cho phép máy khách yêu cầu mô tả đầy đủ một hoặc nhiều bao phủ được phục vụ bởi một máy chủ cụ thể. Máy chủ trả về một tài liệu XML mô tả một cách đầy đủ bao phủ được chỉ định.
- GetCoverage: phương thức bắt buộc nhằm cho phép máy khách yêu cầu một bao phủ bao gồm phạm vi thuộc tính tại một vị trí địa lý được chọn. Máy chủ sẽ trả về dữ liệu đáp ứng từ bao phủ được chọn và mã hóa bao phủ theo định dạng biết trước.

Chương 5 - XÂY DỰNG HỆ THỐNG WEBGIS

Dựa trên những lý thiết đã tìm hiểu, trong chương này chúng em sẽ tiến hành xác định phân tích các yêu cầu của hệ thống. Sau đó sẽ đưa ra giải pháp phù hợp đi đến thiết kế và xây dựng hệ thống đáp ứng các yêu cầu đã đặt ra.

5.1 ĐẶC TẢ YÊU YÊU CẦU HỆ THỐNG

5.1.1 Hiển thị bản đồ với các thông tin chi tiết

- *Mục đích:* cho phép hiển thị bản đồ của toàn thành phố Cần Thơ với các thông tin như ranh giới hành chính quận/huyện, ranh giới hành chính phường/xã, đường đi, các cơ quan hành chính, bệnh viện, trường học...
- *Tiền điều kiện:* Không
- *Cách xử lý:* mặc nhiên khi người dùng truy cập vào hệ thống thì chức năng này sẽ được thực thi.

5.1.2 Phóng to, thu nhỏ bản đồ

- *Mục đích:* cho phép người dùng phóng to/thu nhỏ bản đồ.
- *Tiền điều kiện:* Không
- *Cách xử lý:* có 3 cách để thực hiện chức năng này.
 - *Cách 1:* Khi người dùng nhấn vào biểu tượng phóng to/thu nhỏ thì bản đồ sẽ được phóng to/thu nhỏ tương ứng với số lần nhấn của người dùng. Trong trường hợp này, mặc định vị trí được chọn để phóng to/thu nhỏ là ở trung tâm của vùng được hiển thị trên bản đồ.
 - *Cách 2:* Khi người dùng nhấp đúp vào một vị trí trên bản đồ thì bản đồ sẽ được phóng to tại vị trí của con trỏ.
 - *Cách 3:* Sử dụng nút cuộn trên chuột để phóng to/thu nhỏ bản đồ. Khi người dùng cuộn lên hoặc cuộn xuống thì bản đồ sẽ được phóng to/ thu nhỏ tại vị trí con trỏ chuột.

5.1.3 Dịch chuyển bản đồ

- *Mục đích:* cho phép người dùng dịch chuyển bản đồ theo một hướng nào đó trên vùng hiển thị.
- *Tiền điều kiện:* Không
- *Cách xử lý:* Bản đồ sẽ dịch chuyển theo vị trí con trỏ trong vùng hiển thị khi người dùng nhấn giữ chuột trái và di chuyển.

5.1.4 Xem thông tin các địa điểm trên bản đồ

- *Mục đích:* cho phép người dùng xem thông tin chi tiết (tên, địa chỉ...) của một đối tượng trên bản đồ.
- *Tiền điều kiện:* các địa điểm đã được hiển thị lên bản đồ thông qua các chức năng tìm kiếm địa điểm hoặc khi bản đồ được phóng to đến mức 6 (zoom-level = 6).
- *Cách xử lý:* khi người dùng click lên một địa điểm trên bản đồ, một popup được hiển thị để trình bày các thông tin về địa điểm được chọn.

5.1.5 Tìm và hiển thị đường đi ngắn nhất giữa hai điểm

- *Mục đích:* cho phép tìm đường đi giữa 2 điểm đã chọn trước trên bản đồ
- *Tiền điều kiện:* người dùng phải chọn 2 điểm trước thông qua chức năng tìm đường hay tìm địa điểm.
- *Cách xử lý:*
 - Để thực hiện chức năng này người dùng phải tìm **2** điểm bắt đầu và điểm kết thúc trên bản đồ.
 - Cách 1: nhập tên của hai địa điểm để hệ thống xác định 2 điểm và hiển thị trên bản đồ
 - Cách 2: người dùng chọn trực tiếp 2 điểm trên bản đồ bằng cách nhấp chọn chức năng chọn điểm, sau đó click chọn hai vị trí trên bản đồ.
 - Cách 3: từ popup hiển thị thông tin của một địa điểm trên bản đồ, chọn từ đây (điểm bắt đầu) hoặc đến đây (điểm kết thúc).
 - Sau khi xác định được 2 điểm sẽ có các trường hợp:

- Nếu cả 2 điểm mà người dùng chọn đều nằm trên đường đi có trên bản đồ thì hệ thống sẽ tiến hành việc tìm đường.
- Nếu có một điểm mà người dùng chọn không nằm trên một đường đi trên bản đồ:
 - + Đầu tiên hệ thống sẽ tìm địa chỉ của điểm được chọn. Nếu tồn tại, điểm đó sẽ được ánh xạ đến điểm gần nhất trên đường đi (đường đi này được lấy từ địa chỉ của điểm đó).
 - + Ngược lại hệ thống sẽ tìm ra đường đi gần nhất với điểm được chọn, sau đó sẽ ánh xạ đến điểm gần nhất trên đường đi đó.
- Sau khi hệ thống xác định được 2 điểm thì tiến hành tìm đường và hiển thị kết quả cho người dùng. Ngoài chỉ đường trên bản đồ hệ thống cũng chỉ đường cho người dùng bằng văn bản. Trên đường đi từ A đến B hệ thống sẽ chỉ ra những đoạn nào sẽ đi qua cùng với độ dài của mỗi đoạn đường và tổng độ dài đường đi.
- Sau khi đã hiển thị kết quả đường đi của 2 điểm trên bản đồ, người dùng có thể nhấn và giữ chuột để di chuyển điểm A hay B để thay đổi vị trí muốn tìm đường và hệ thống sẽ tiếp tục tìm và hiển thị kết quả đường đi mới trên bản đồ.

5.1.6 Tìm và hiển thị các địa điểm xung quanh vị trí được chọn

- *Mục đích:* cho phép tìm và hiển thị địa điểm: trường học, bệnh viện, chợ ... trong các quận/huyện xung quanh một vị trí mong muốn đã chọn với khoảng cách xác định trước.
- *Tiền điều kiện:* người dùng đã chọn xem thông tin tại một vị trí mong muốn: trường học, chợ, bệnh viện, ủy ban, ...
- *Cách xử lý:*
 - Người dùng chọn chức năng tìm xung quanh tại một đối tượng thông qua popup hiển thị thông tin, nhập vào bán kính giới hạn phạm vi và tên địa điểm tìm.

- Hệ thống hiển thị kết quả bao gồm kết quả theo bản đồ và kết quả văn bản.
 - Kết quả theo bản đồ: hệ thống sẽ hiển thị tất cả các địa điểm được tìm thấy trên bản đồ dưới dạng một icon mặc định.
 - Kết quả văn bản: bao gồm số lượng và danh sách địa điểm được tìm thấy cùng với các thông tin chi tiết như tên, địa chỉ... của các địa điểm được tìm thấy.

5.1.7 Tìm kiếm và hiển thị các địa điểm theo vùng

- *Mục đích:* cho phép tìm kiếm và hiển thị các địa điểm trong những vùng được chọn trước. Vùng ở đây có thể là một hoặc nhiều quận/huyện hay phường/xã.
- *Tiền điều kiện:* Không
- *Cách xử lý:*
 - Người dùng chọn vùng giới hạn cần tìm các địa điểm.
 - Người dùng nhập vào tên địa điểm hoặc chọn chủ đề muốn tìm kiếm.
 - Hệ thống sẽ hiển thị kết quả theo bản đồ và kết quả theo văn bản
 - Kết quả theo bản đồ: hệ thống sẽ hiển thị tất cả các địa điểm được tìm thấy trên bản đồ dưới dạng một icon mặc định.
 - Kết quả văn bản: bao gồm số lượng và danh sách địa điểm được tìm thấy cùng với các thông tin chi tiết như tên, địa chỉ...

5.1.8 Tìm và hiển thị địa điểm theo chủ đề

- *Mục đích:* cho phép tìm và hiển thị địa điểm theo từng chủ đề.
- *Tiền điều kiện:* Không
- *Cách xử lý:*
 - Người dùng chọn chủ đề (trường, bệnh viện, ...) muốn xem.
 - Hệ thống hiển thị kết quả bao gồm kết quả theo bản đồ và kết quả văn bản.
 - Kết quả theo bản đồ: hệ thống sẽ hiển thị tất cả các địa điểm được tìm thấy trên bản đồ dưới dạng một icon mặc định.
 - Kết quả văn bản: bao gồm số lượng và danh sách địa điểm được tìm thấy cùng với các thông tin chi tiết như tên, địa chỉ...

5.1.9 Tìm và hiển thị địa điểm trên lộ trình

- *Mục đích:* cho phép tìm và hiển thị địa điểm (trường học, chợ, bệnh viện, ủy ban.....) dọc theo lộ trình giữa 2 điểm đã chọn trước trên bản đồ.
- *Tiền điều kiện:* hệ thống đang hiển thị bản đồ và hướng dẫn chỉ đường giữa 2 điểm đã được chọn trước đó.
- *Cách xử lý:*
 - Người dùng chọn chức năng tìm địa điểm trên lộ trình
 - Người nhập tên địa điểm cần tìm trên lộ trình và khoảng cách giới hạn cần tìm.
 - Hệ thống hiển thị kết quả bao gồm kết quả theo bản đồ và kết quả văn bản.
 - Kết quả theo bản đồ: hệ thống sẽ hiển thị tất cả các địa điểm được tìm thấy trên bản đồ dưới dạng một icon mặc định.
 - Kết quả văn bản: bao gồm số lượng và danh sách địa điểm được tìm thấy cùng với các thông tin chi tiết như tên, địa chỉ...

5.1.10 Đo khoảng cách đường chim bay giữa hai điểm bất kỳ

- *Mục đích:* cho phép người dùng đo khoảng cách đường chim bay giữa hai điểm do người dùng chọn.
- *Tiền điều kiện:* Không
- *Cách xử lý:*
 - Người dùng chọn chức năng đo khoảng cách. Sau đó chọn hai điểm trên bản đồ cần đo khoảng cách.
 - Hệ thống hiển thị kết quả là độ dài đường chim bay giữa hai điểm.

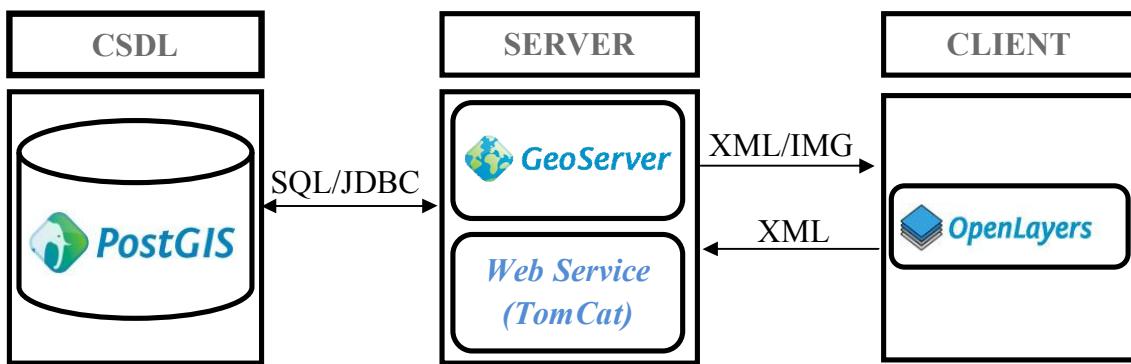
5.1.11 Tìm và hiển thị các con đường theo tên

- *Mục đích:* cho phép tìm và hiển thị các con đường theo tên đường.
- *Tiền điều kiện:* Không
- *Cách xử lý:*
 - Người nhập vào tên con đường muốn tìm.

- Hệ thống hiển thị tất cả các con đường được tìm thấy bằng văn bản, cho phép người dùng chọn để xem vị trí của từng con đường trên bản đồ.

5.2 MÔ HÌNH TỔNG THỂ CỦA HỆ THỐNG

Để đáp ứng các yêu cầu mô hình tổng thể của hệ thống gồm các phần sau:



Hình 5.1- Mô hình tổng thể của hệ thống

Diễn giải mô hình

PostGIS: Nơi lưu trữ dữ liệu không gian và phi không gian. Các dữ liệu này được lưu trữ dưới dạng các bảng. Ngoài ra PostGIS còn lưu trữ các hàm được định nghĩa để hỗ trợ xây dựng dịch vụ.

GeoServer: Cung cấp các dịch vụ dữ liệu bản đồ như WMS, WFS, WCS. Cho phép tạo ra các bản đồ nền dựa trên dữ liệu địa lý và các style được định nghĩa.

Web Service: Truy xuất dữ liệu không gian, phi không gian từ CSDL, sau đó xử lý và tạo ra các dịch vụ đáp ứng các yêu cầu của hệ thống như tìm đường đi, địa điểm.

OpenLayers: Chồng lớp bản đồ. Nhận và hiển thị thông tin từ dịch vụ WMS và WebService.

5.3 PHƯƠNG PHÁP LÀM VIỆC VÀ PHÂN CHIA CÔNG VIỆC

5.3.1 Phân chia công việc

Trong quá trình làm việc chúng em đều cùng nhau tìm hiểu về mặt lý thuyết cũng như giải thuật cài đặt. Mọi vấn đề này sinh thì cả hai thành viên cùng nghiên

cứu, đưa ra hướng giải quyết sau đó chia nhau thực hiện. Dưới đây là bảng phân chia công việc cho quá trình phát triển hệ thống:

STT	Công việc	Thực hiện
1	Tìm hiểu yêu cầu	Hoàng, Pha
2	Phân tích thiết kế hệ thống	Hoàng, Pha
3	Xây dựng dữ liệu	Hoàng, Pha
4	Cấu hình GeoServer	Pha
5	Định dạng bản đồ	Hoàng
6	Xây dựng thủ tục trong PostGIS	Pha
7	Xây dựng dịch vụ đường đi	Hoàng
8	Xây dựng dịch vụ tìm kiếm địa điểm	Hoàng, Pha
9	Xây dựng ứng dụng phía Client	Hoàng, Pha
10	Viết báo cáo	Hoàng, Pha

Hình 5.2- Bảng phân chia công việc

5.3.2 Sử dụng SVN

Do trong quá trình học môn Phát triển phần mềm mã nguồn mở, chúng em được học về Subversion, hệ thống quản lý phiên bản cho phần mềm mã nguồn mở. Thấy đây là cách tốt nhất để làm việc nhóm nên chúng em quyết định tìm hiểu và sử dụng SVN cho luận văn này. Việc này đem lại cho chúng em một số thuận lợi trong quá trình thực hiện như:

- Quản lý được phiên bản của dự án, không lẫn lộn giữa các phiên bản, có thể tự cập nhật, so sánh, và kết hợp tài nguyên mới vào phần tài nguyên cũ.
- Các thành viên có thể làm việc trực tuyến cùng nhau.
- Tích hợp code: update, commit, add, delete dễ dàng.
- Có thể theo dõi công việc, tiến độ lẫn nhau.
- Tiết kiệm được nhiều thời gian không phải cấu hình lại hệ thống nhiều lần, không còn tình trạng tích hợp code bằng cách copy/paste thủ công như các đề tài làm theo nhóm của những niên luận trước đây.

5.4 XÁC ĐỊNH CÁC GIẢI PHÁP VÀ XÂY DỰNG DỊCH VỤ

5.4.1 Xây dựng dữ liệu bản đồ bằng phần mềm MapInfo

5.4.1.1 Lấy ảnh từ google Map

Để tiến hành số hóa bản đồ điều đầu tiên là cần phải có một bản đồ nền. Bản đồ nền có thể là một bản đồ bằng giấy hoặc một tập tin định dạng ảnh trên máy tính. Đối với bản đồ giấy thì việc số hóa đòi hỏi phải sử dụng bàn số hóa hoặc phải chuyển sang định dạng ảnh mới có thể số hóa bằng các phần mềm chuyên dụng trên máy tính. Hiện nay, có một nguồn dữ liệu bản đồ được cộng đồng đánh giá cao về mức độ chính xác cũng như là chất lượng đó là nguồn bản đồ từ Google Map. Các bản đồ này sau khi được thu thập về sẽ được lưu dưới định dạng ảnh. Để lấy ảnh từ Google Map cần phải sử dụng các phần mềm chuyên dụng khác, và ở đây chúng em sử dụng phần mềm “*Easy Google Maps Downloader*” cho việc thu thập ảnh vì tính dễ sử dụng của nó so với các phần mềm khác. Bên cạnh đó, do mức độ chi tiết của bản đồ nền cần sử dụng phải nhiều, cho nên ảnh bản đồ lấy về cần có độ phân giải cao, việc này kéo theo dung lượng của ảnh lấy về là rất lớn (có thể lên đến hàng Gigabyte trên một ảnh cho khu vực thành phố Cần Thơ). Giải pháp là cần phải chia khu vực thành phố Cần Thơ thành những khu vực nhỏ hơn để tiến hành thu thập dữ liệu. Sau đó tiến hành ghép các phần bản đồ đã đăng ký này thành bản đồ nền hoàn chỉnh.

5.4.1.2 Số hóa các đặc trưng cần thiết

Sử dụng các kiến thức về Mapinfo và cơ sở toán học về bản đồ tiến hành vẽ bản đồ trong phạm vi thành phố Cần Thơ hay còn gọi là số hóa bản đồ. Các đối tượng cần thể hiện trên bản đồ bao gồm: ranh giới chung của thành phố Cần Thơ, ranh giới giữa các vùng (quận/huyện, phường/xã...), hệ thống đường đi, các cơ quan nhà nước ... cùng với các thông tin chi tiết của nó. Thực tế, để đơn giản trong quá trình tổ chức dữ liệu và xử lý thì các hệ thống GIS thường phân bản đồ thành các lớp để thể hiện các đối tượng có quan hệ với nhau về tính chất địa lý, ví dụ: lớp

ranh giới hành chính quận/huyện, lớp đường đi, lớp các cơ quan nhà nước, lớp bệnh viện... trong phạm vi của đề tài này bản đồ sẽ được phân ra thành các lớp sau:

- Lớp ranh giới hành chính các quận/huyện.
- Lớp ranh giới hành chính các phường/xã.
- Lớp giao thông. Lớp này tập trung mô hình hóa hệ thống đường đi trong quận Ninh Kiều.
- Các lớp địa điểm: cơ quan, trường học, bệnh viện, chợ, bến tàu/xe, khách sạn, công ty, giải trí, đền chùa, truyền thông, ngân hàng, công viên, cầu, thư viện, bưu điện.

Bên cạnh đó trong quá trình số hóa cũng đồng thời nhập dữ liệu thuộc tính cho các lớp trên (nếu có), chẳng hạn lớp đường có các thuộc tính như: mã đường, tên đường, loại đường... Các dữ liệu này sẽ được lưu lại dưới các định dạng của phần mềm Mapinfo (*.tab, *.dat...). Thông tin để tạo ra mỗi lớp bản đồ được lưu trữ thành nhóm 4 tập tin cùng tên, ví dụ lớp ranh giới hành chính quận huyện sẽ được vẽ ra từ bốn tập tin là quanhuyen.TAB, quanhuyen.DAT, quanhuyen.MAP, quanhuyen.ID.

5.4.2 Chuyển dữ liệu sang HQTCSDL PostgreSQL (PostGIS)

Việc lưu trữ dữ liệu trong các tập tin sẽ gặp khó khăn khi xử lý và cập nhật dữ liệu trong quá trình phát triển cũng như vận hành hệ thống. Cho nên việc tổ chức dữ liệu vào một hệ quản trị cơ sở dữ liệu (HQTCSDL) là điều cần thiết.

Hiện nay có rất nhiều HQTCSDL bao gồm cả thương mại và nguồn mở đều hỗ trợ trong quản lý dữ liệu không gian như SQL Server 2008, DB2, Oracle Spatial, PostgreSQL (PostGIS)... Trong đó, PostGIS là một HQTCSDL nguồn mở, miễn phí và có hầu hết các khả năng để thao tác với dữ liệu không gian tương tự các phần mềm thương mại khác, cho nên lựa chọn PostGIS để tổ chức dữ liệu là giải pháp phù hợp nhất.

Do không có công cụ để chuyển đổi dữ liệu trực tiếp từ định dạng tập tin của MapInfo thành các bảng dữ liệu trong PostGIS, nên việc chuyển đổi này được thực

hiện thông qua hai bước. Đầu tiên, từ định dạng tập tin sẽ được chuyển sang định dạng Shapefile bằng chức năng chuyển đổi của phần mềm MapInfo, sau đó các tập tin này sẽ được chuyển thành các bảng trong PostGIS thông qua công cụ “*PostGIS Shapefile and DBF Loader*”. Kết quả cụ thể của các bảng như sau:

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã trường học
3	ten	character varying	254			Tên trường học
4	diachi	character varying	254			Địa chỉ trường học
5	sdt	character varying	15			Số điện thoại trường học
6	the_geom	geometry				Dữ liệu không gian trường học

Bảng 5.1- Bảng ‘truong’: trường học

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã cơ quan
3	ten	character varying	254			Tên cơ quan
4	diachi	character varying	254			Địa chỉ cơ quan
5	sdt	character varying	15			Số điện thoại cơ quan
6	the_geom	geometry				Dữ liệu không gian cơ quan

Bảng 5.2- Bảng ‘coquan’: các cơ quan hành chính

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã bệnh viện
3	ten	character varying	254			Tên bệnh viện
4	diachi	character varying	254			Địa chỉ bệnh viện
5	sdt	character varying	15			Số điện thoại bệnh viện
6	the_geom	geometry				Dữ liệu không gian bệnh viện

Bảng 5.3- Bảng ‘benhvien’: các bệnh viện, trạm y tế

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã bến
3	ten	character varying	254			Tên bến
4	diachi	character varying	254			Địa chỉ bến
5	the_geom	geometry				Dữ liệu không gian bến

Bảng 5.4- Bảng ‘ben’: các bến tàu, bến xe, bến đò.

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã bưu điện
3	ten	character varying	254			Tên bưu điện
4	diachi	character varying	254			Địa chỉ bưu điện
5	sdt	character varying	15			Số điện thoại bưu điện
6	the_geom	geometry				Dữ liệu không gian bưu điện

Bảng 5.5- Bảng ‘buudien’: các bưu điện

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã cầu
3	ten	character varying	254			Tên cầu
4	diachi	character varying	254			Địa chỉ cầu
6	the_geom	geometry				Dữ liệu không gian cầu

Bảng 5.6- Bảng ‘cau’: các cầu.

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã chợ
3	ten	character varying	254			Tên chợ
4	diachi	character varying	254			Địa chỉ chợ
5	sdt	character varying	15			Số điện thoại chợ
6	the_geom	geometry				Dữ liệu không gian chợ

Bảng 5.7- Bảng ‘cho’: các siêu thị, chợ, cửa hàng

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã công ty
3	ten	character varying	254			Tên công ty
4	diachi	character varying	254			Địa chỉ công ty
5	sdt	character varying	15			Số điện thoại công ty
6	the_geom	geometry				Dữ liệu không gian công ty

Bảng 5.8- Bảng ‘congty’: các công ty, nhà máy.

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã công viên
3	ten	character varying	254			Tên công viên
4	diachi	character varying	254			Địa chỉ công viên
6	the_geom	geometry				Dữ liệu không gian công viên

Bảng 5.9- Bảng ‘congviен’: các công viên

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã đèn/chùa
3	ten	character varying	254			Tên đèn/chùa
4	diachi	character varying	254			Địa chỉ đèn/chùa
5	sdt	character varying	15			Số điện thoại đèn/chùa
6	the_geom	geometry				Dữ liệu không gian đèn/chùa

Bảng 5.10- Bảng ‘denchua’: các đèn, chùa, miếu, nhà thờ

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã giải trí
3	ten	character varying	254			Tên giải trí
4	diachi	character varying	254			Địa chỉ giải trí
5	sdt	character varying	15			Số điện thoại giải trí
6	the_geom	geometry				Dữ liệu không gian giải trí

Bảng 5.11- Bảng ‘giaitri’: các khu vui chơi, giải trí, quán cafe

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã khách sạn
3	ten	character varying	254			Tên khách sạn
4	diachi	character varying	254			Địa chỉ khách sạn
5	sdt	character varying	15			Số điện thoại khách sạn
6	the_geom	geometry				Dữ liệu không gian khách sạn

Bảng 5.12- Bảng ‘khachsan’: các khách sạn, nhà nghỉ

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã ngân hàng
3	ten	character varying	254			Tên ngân hàng
4	diachi	character varying	254			Địa chỉ ngân hàng
5	sdt	character varying	15			Số điện thoại ngân hàng
6	the_geom	geometry				Dữ liệu không gian ngân hàng

Bảng 5.13- Bảng ‘nganhang’: các ngân hàng, máy ATM

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	character varying	10			Mã thư viện
3	ten	character varying	254			Tên thư viện
4	diachi	character varying	254			Địa chỉ thư viện
5	sdt	character varying	15			Số điện thoại thư viện
6	the_geom	geometry				Dữ liệu không gian thư viện

Bảng 5.14- Bảng ‘thuvien’: các thư viện, nhà sách

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma_duong	character varying	10			Mã đường đi
3	ten_duong	character varying	30			Tên đường đi
4	loai_duong	character varying	15			Loại đường đi
5	mot_chieu	integer				Là đường một chiều
6	the_geom	geometry				Dữ liệu không gian của đường đi
7	nut_nguon	integer				Nút bắt đầu của đường đi
8	nut_dich	integer				Nút kết thúc của đường đi
9	chieu_dai	double precision				Chiều dài của đường đi

Bảng 5.15- Bảng ‘giaothong’: các đường đi

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Số thứ tự của điểm giao
2	the_geom	geometry				Dữ liệu không gian của điểm giao

Bảng 5.16- Bảng ‘dinh’: các điểm giao giữa các đường đi và các đầu mút

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	integer				Mã quận/huyện
3	ten	character varying	50			Tên quận/huyện
4	the_geom	geometry				Dữ liệu không gian của quận/huyện

Bảng 5.17- Bảng ‘quanhuyen’: địa giới hành chính của các quận huyện

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	gid	serial		x	x	Mã số khóa chính
2	ma	integer				Mã xã/phường
3	ten	character varying	30			Tên xã/phường
4	ma_huyen	integer				Mã quận/huyện
5	the_geom	geometry				Dữ liệu không gian của xã/phường

Bảng 5.18- Bảng ‘xaphuong’: địa giới hành chính của các xã phường

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	f_table_catalog	character varying	256	x	x	Tên catalog
2	f_table_schema	character varying	256	x		Tên schema
3	f_table_name	character varying	256	x		Tên bảng không gian
4	f_geometry_column	character varying	256	x		Tên cột chứa dữ liệu không gian
5	coord_dimension	integer		x		Số chiều của dữ liệu không gian
6	srid	integer		x		Chỉ số hệ thống tham chiếu không gian
7	type	character varying	30	x		Kiểu dạng vector

Bảng 5.19- Bảng ‘geometry_columns’: lưu thông tin các bảng dữ liệu không gian

STT	Tên Trường	Kiểu	Kích thước	Not null	Khóa chính	Diễn giải
1	srid	integer		x	x	Chỉ số hệ thống tham chiếu không gian
2	auth_name	character varying	256			Tên chuẩn/ tác giả
3	auth_srid	integer	256			Chỉ số hệ thống tham chiếu theo chuẩn
4	srtext	character varying	2048			Định dạng WKT của hệ thống tham chiếu
5	proj4text	character varying	2048			Chuỗi định nghĩa tọa độ của hệ thống tham chiếu thư viện Proj4

Bảng 5.20- Bảng ‘spatial_ref_sys’: lưu thông tin các hệ qui chiếu không gian.

Trong các bảng dữ liệu trên, thì bảng “*dinh*” là bảng được tạo ra sau khi dữ liệu đã được chuyển vào trong CSDL PostGIS. Bảng này được tạo ra bằng một thủ tục “*assign_vertex_id*” được xây dựng trong PostGIS. Thủ tục này sẽ tìm ra các điểm giao giữa các đường đi và thêm các điểm giao này vào bảng “*dinh*”, đồng thời cập nhật các giá trị cho các trường “*nut_nguon*” và “*nut_dich*” tương ứng trong bảng “*giaothong*” để tạo mối quan hệ topology giữa bảng “*dinh*” và bảng cạnh (trong trường hợp này là bảng “*giaothong*”).

Trong khi đó, các bảng “*geometry_columns*” và “*spatial_ref_sys*” là các bảng có sẵn trong CSDL PostGIS để lưu các thông tin về các bảng dữ liệu không gian.

5.4.3 Tạo kết nối giữa Geoserver và CSDL PostGIS

Sau khi xây dựng dữ liệu cho hệ thống, thì tiến hành tạo các nối kết giữa các bảng trong PostGIS với GeoServer. Cho phép các dịch vụ phía GeoServer truy xuất dữ liệu và tạo ra các lớp bản đồ. Đồng thời, cũng biên tập các tài liệu SLD để định dạng kiểu dáng cho dữ liệu phía server. Do yêu cầu truy xuất dữ liệu từ GeoServer để tạo bản đồ nền phía client, nên cần biên tập các tài liệu SLD cho các lớp “*giaothong*”, “*quanhuyen*”, “*xaphuong*”, “*benhvien*”, “*coquan*”. Phía client sẽ khai thác các dịch vụ này để lấy về các bản đồ dưới định dạng raster đã được định dạng kiểu dáng rồi hiển thị lên trình duyệt.

Các bước tiến hành cấu hình để kết nối GeoServer với CSDL PostGIS, và định dạng kiểu dáng cho dữ liệu:

- Tạo Workspace: là nơi để chứa tất cả các lớp trong hệ thống.
- Tạo kho chứa (Store): tạo các kết nối đến các nguồn dữ liệu, trong trường hợp ứng dụng sử dụng nhiều nguồn dữ liệu khác nhau thì cần tạo các kho chứa tương ứng với các nguồn đó. Hiện tại, chỉ sử dụng một nguồn dữ liệu từ PostGIS do đó chỉ cần tạo kết nối đến CSDL PostGIS.
- Tạo lớp (Layer): tiến hành tạo các lớp vector: quận/ huyện, phường/xã, đường đi, cơ quan, trường học, bệnh viện....
- Tạo kiểu (Style): xây dựng tập các tài liệu SLD để định dạng kiểu dáng cho các lớp dữ liệu.

5.5 BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT GIỮA HAI ĐIỂM

5.5.1 Giới thiệu

Tìm đường đi ngắn nhất từ hai điểm chọn trước trên bản đồ là yêu cầu cơ bản trong xây dựng hệ thống GIS. Khi đó thuật toán kinh điển giải quyết cho vấn đề này là Dijkstra. Thuật toán Dijkstra của nhà khoa học máy tính người Hà Lan Edsger Dijkstra, là một thuật toán giải quyết bài toán tìm đường đi ngắn nhất từ một đỉnh đến các đỉnh khác trong một đồ thị có hướng không có cạnh mang trọng số âm.

5.5.2 Mô hình hệ thống giao thông thành đồ thị có hướng

Trước khi áp dụng được thuật toán Dijkstra và giải quyết vấn đề tìm đường đi ngắn nhất giữa hai đỉnh trên đồ thị, thì cần phải mô hình hệ thống giao thông trong thành phố Cần Thơ thành một đồ thị có hướng với các đỉnh là các điểm giao giữa các con đường, và các cạnh là đường nối giữa các điểm giao đó. Khi đó trọng lượng của cạnh chính là chiều dài của đường đi. Bên cạnh đó cũng cần phải quan tâm đến chiều của đường đi, việc này đã được thực hiện trong quá trình số hóa dữ liệu từ định dạng ảnh (raster) sang định dạng vector. Khi đó, chiều của một đường đi được tính theo thứ tự vẽ các điểm trên đường đi đó, nghĩa là điểm đầu tiên được vẽ được xem như là nút nguồn, điểm cuối cùng được vẽ xem như là điểm đích, và chiều của đường đi chính là chiều đi từ điểm nguồn đến điểm đích. Tuy nhiên, trên thực tế hệ thống giao thông bao gồm cả đường một chiều và đường hai chiều. Do đó, để giải quyết vấn đề này thì cần phải thêm trường dữ liệu thuộc tính “số chiều” cho đường đi. Do đó trong khi mô hình hóa đồ thị, nếu đường nào có số chiều là hai thì sẽ được thêm vào đồ thị hai lần theo chiều ngược nhau. Như vậy, tìm đường đi ngắn nhất giữa hai điểm trên bản đồ bây giờ được chuyển thành bài toán tìm đường đi ngắn nhất giữa hai đỉnh trên đồ thị có hướng.

Nhưng không phải lúc nào người dùng cũng chọn điểm bắt đầu và cuối trùng với hai đỉnh của đồ thị, mà là hai điểm bất kỳ trên bản đồ. Trong trường hợp này thì đầu tiên cần phải tìm địa chỉ của hai điểm hiện tại (trong trường hợp điểm chọn trùng với các địa điểm như cơ quan, trường học...), nếu tồn tại địa chỉ thì ánh xạ

điểm được chọn lên đường đi có trong địa chỉ đó và tìm ra điểm gần với điểm được chọn nhất trên đường đi đó. Ngược lại, sẽ tìm hai đường đi bất kỳ gần nhất với hai điểm do người dùng chọn, sau đó tìm một điểm gần nhất trên đường gần nhất với hai điểm chọn đó. Lúc này sẽ chuyển sang tìm đường đi giữa hai điểm gần nhất vừa tìm. Có các trường hợp cho hai điểm gần nhất đó:

- Hai điểm này nằm trên hai đỉnh của đồ thị. Vấn đề tìm đường lúc này đơn giản là tìm từ đỉnh đầu đến đỉnh cuối.
- Khi điểm gần nhất nằm trên đường đi mà không trùng với đỉnh đầu cũng như đỉnh cuối của đường đi đó thì cần phải thêm vào đồ thị một nút mới (nút ảo) xen vào giữa đường đi và tách đường đi này thành hai đường đi mới có thuộc tính (chiều, tên đường) giống như đường cũ. Lúc này đường đi sẽ được tìm giữa các nút mới thêm vào.

Lưu ý: nút ảo như đề cập ở trên là nút mới được thêm vào trong trường hợp một đoạn đường được tách làm hai hoặc ba đoạn mới. Nút này chỉ có tác dụng khi mô hình hóa đồ thị (tức trong quá trình thực thi chương trình), mà không ảnh hưởng đến CSDL hiện có của hệ thống. Do đó, khi chương trình kết thúc cũng đồng nghĩa với việc các nút ảo này sẽ bị mất.

5.5.3 Đánh giá

Độ phức tạp của thuật toán Dijkstra được cài đặt phía dịch vụ là $O(n^2)$ (với n là số đỉnh) trong trường hợp tìm đường đi từ đỉnh nguồn đến tất cả các đỉnh còn lại trong đồ thị. Tuy nhiên, ở đây do yêu cầu của bài toán chỉ cần tìm đường đi từ điểm bắt đầu đến điểm kết thúc, do đó để cải tiến tốc độ thì chỉ cần xét đến điểm kết thúc của đường đi là ta có thể dừng giải thuật. Cách giải quyết này không phải luôn luôn làm giảm được thời gian thực hiện giải thuật, tuy nhiên phần lớn trường hợp thì mang lại hiệu quả thời gian tính toán tốt hơn.

5.6 TÌM KIẾM ĐỊA ĐIỂM

Tìm kiếm địa điểm là một trong những yêu cầu chức năng cho phép người dùng tìm kiếm thông tin các địa điểm theo tên hoặc theo vị trí của nó. Để hỗ trợ một

cách hiệu quả cho quá trình tìm kiếm, thì kiếm địa điểm được chia ra làm nhiều dạng tìm kiếm như sau:

- Tìm kiếm địa điểm theo tên: người dùng sẽ nhập vào tên của địa điểm cần tìm. Khi đó hệ thống cần nhận biết chuỗi tên do người dùng nhập vào là có dấu hay không có dấu. Nếu chuỗi có dấu thì chỉ cần so sánh chuỗi nhập với tên của các địa điểm hiện có trong dữ liệu. Ngược lại, nếu chuỗi là không dấu thì tiến hành chuyển dữ liệu của trường tên trong dữ liệu thành không dấu, rồi so sánh với chuỗi do người dùng nhập vào. Lưu ý rằng tất cả các phép so sánh là so sánh gần đúng và không phân biệt chữ hoa hay thường.
- Tìm kiếm địa điểm theo chủ đề (lớp): khi người dùng chọn chủ đề muốn tìm (ví dụ: bệnh viện), thì hệ thống lúc này chỉ cần truy xuất vào bảng chứa của chủ đề cần tìm.

Ngoài ra, tìm kiếm địa điểm theo tên hay theo chủ đề còn được kết hợp với phạm vi giới hạn tìm kiếm, trong trường hợp người dùng đã chọn trước vùng giới hạn. Nghĩa là, khi người dùng có chọn vùng (một hoặc nhiều quận/huyện hay phường/xã) tìm kiếm, thì trong quá trình tìm, hệ thống chỉ xét đến những địa điểm có vị trí (dữ liệu không gian) nằm trong những vùng do người dùng đã chọn trước đó, ngược lại hệ thống sẽ tìm trên toàn phạm vi thành phố Cần Thơ.

- Tìm kiếm địa điểm theo lộ trình: chức năng này chỉ được thực hiện khi đường đi ngắn nhất giữa hai điểm được hiển thị trên bản đồ. Kết hợp với tên của địa điểm cần tìm và khoảng cách giới hạn từ địa điểm cần tìm đến đường đi. Khi đó hệ thống sẽ tìm ra các địa điểm nằm dọc đường đi có tên gần đúng với chuỗi nhập và có khoảng cách từ điểm đó đến đường đi nhỏ hơn hoặc bằng khoảng cách do người dùng nhập vào. Khoảng cách giới hạn mặc định là 100 m.
- Tìm kiếm địa điểm xung quanh một vị trí: cho phép tìm kiếm các địa điểm xung quanh một điểm được chọn trước trong giới hạn về bán kính tìm kiếm được người dùng xác định trước. Hệ thống sẽ tìm các điểm có tên gần đúng với chuỗi nhập và khoảng cách từ điểm đó đến vị trí chọn trước nhỏ hơn hoặc bằng giá trị mà người dùng nhập vào. Mặc định khoảng cách này có giá trị là 200 m.

5.7 XÂY DỰNG TẬP DỊCH VỤ TRUY XUẤT BẢN ĐỒ

Phần này sẽ trình bày các dịch vụ cần được xây dựng để đáp ứng các yêu cầu của hệ thống:

- Tìm đường đi ngắn nhất giữa hai điểm chọn trước trên bản đồ.
 - *Đầu vào:* tọa độ hai điểm bất kỳ trên bản đồ.
 - *Đầu ra:* danh sách các đoạn đường, cùng với độ dài của đường đi ngắn nhất.
- Tìm gần đúng theo tên các địa điểm trên bản đồ.
 - *Đầu vào:* tên địa điểm cần tìm.
 - *Đầu ra:* danh sách các địa điểm có tên gần đúng với tên cần tìm.
- Tìm gần đúng theo tên các địa điểm trong một hoặc nhiều vùng (quận/huyện, xã/phường) chọn trước.
 - *Đầu vào:* tên địa điểm cần tìm và danh sách các vùng được chọn.
 - *Đầu ra:* danh sách các địa điểm có tên gần đúng với tên cần tìm trong vùng chọn trước.
- Tìm các địa điểm theo chủ đề
 - *Đầu vào:* tên chủ đề được chọn
 - *Đầu ra:* danh sách các địa điểm thuộc chủ đề đó
- Tìm các địa điểm xung quanh một vị trí được chọn trước trên bản đồ với một khoảng cách xác định.
 - *Đầu vào:* chọn vị trí trên bản đồ, tên địa điểm và khoảng cách giới hạn vùng để tìm.
 - *Đầu ra:* danh sách các địa điểm xung quanh vị trí được chọn.
- Tìm kiếm các địa điểm dọc theo lộ trình (đường đi ngắn nhất giữa hai điểm) với một khoảng cách giới hạn xác định
 - *Đầu vào:* danh sách các đoạn đường của lộ trình, tên địa điểm và khoảng cách giới hạn vùng được chọn (được tính từ điểm đó đến lộ trình).
 - *Đầu ra:* danh sách các địa điểm xung quanh lộ trình được chọn.
- Tìm kiếm các địa điểm theo chủ đề (lớp) trong một hoặc nhiều vùng (quận/huyện, xã/phường) chọn trước.

- *Đầu vào:* chủ đề và danh sách vùng được chọn.
- *Đầu ra:* danh sách các địa điểm theo chủ đề trong vùng được chọn.
- Tìm kiếm thông tin của một địa điểm trên bản đồ.
 - *Đầu vào:* địa điểm được chọn.
 - *Đầu ra:* Thông tin của địa điểm được chọn (tên, địa chỉ, số điện thoại nếu có).
- Tìm gần đúng theo tên các con đường trên bản đồ.
 - *Đầu vào:* tên con đường cần tìm
 - *Đầu ra:* Danh sách các con đường có tên gần đúng với tên cần tìm.

5.8 GIẢI PHÁP PHÍA CLIENT

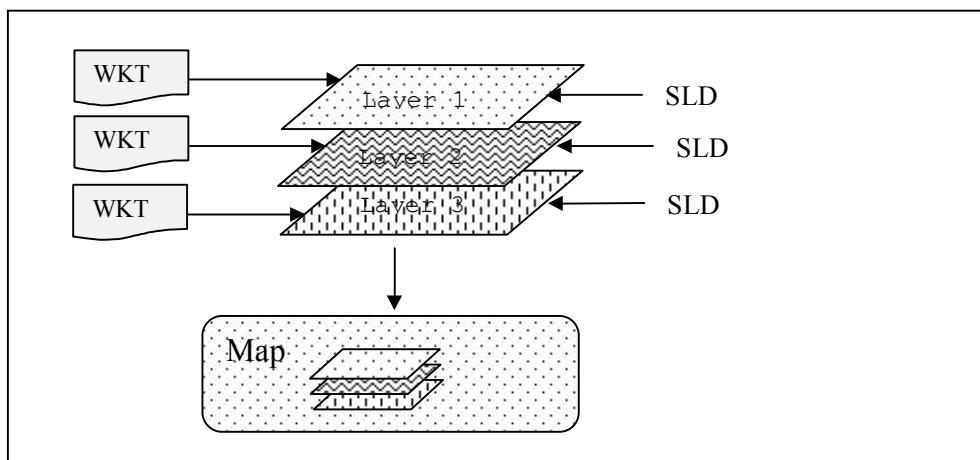
Phần này sẽ trình bày các giải pháp tích hợp bản đồ và khai thác thông tin thuộc tính để phát triển hệ thống. Ở đây, thư viện OpenLayers sẽ được dùng cho việc hiển thị bản đồ.

Các lớp bản đồ sẽ được chồng lên nhau, trong đó có một lớp định dạng ảnh (raster) được đặt làm lớp nền (base layer), các lớp vector chồng trên. Để có được lớp bản đồ từ máy khách, client sẽ gửi yêu cầu GetMap với các tham số theo yêu cầu và theo mục đích cần hiển thị. Cài đặt cụ thể của các lớp được thực hiện như sau:

- *Lớp nền:* được yêu cầu ngay từ khi hệ thống được khởi động để hiện thị bản đồ nền. Dữ liệu của lớp này có được do truy xuất dịch vụ WMS từ GeoServer. Trên bản đồ nền sẽ thể hiện các thông tin cơ bản như: ranh giới, tên các quận/huyện; ranh giới và tên các xã/phường; các địa điểm quan trọng như cơ quan nhà nước, bệnh viện.
- *Lớp điểm chọn (vector):* dùng để chứa điểm bắt đầu và điểm kết thúc của đường đi do người dùng chọn.
- *Lớp đường đi (vector):* lớp này được sử dụng để hiện thị đường đi ngắn nhất giữa hai điểm được chọn sau khi phân tích tập tin kết quả XML trả về từ dịch vụ tìm đường đi.

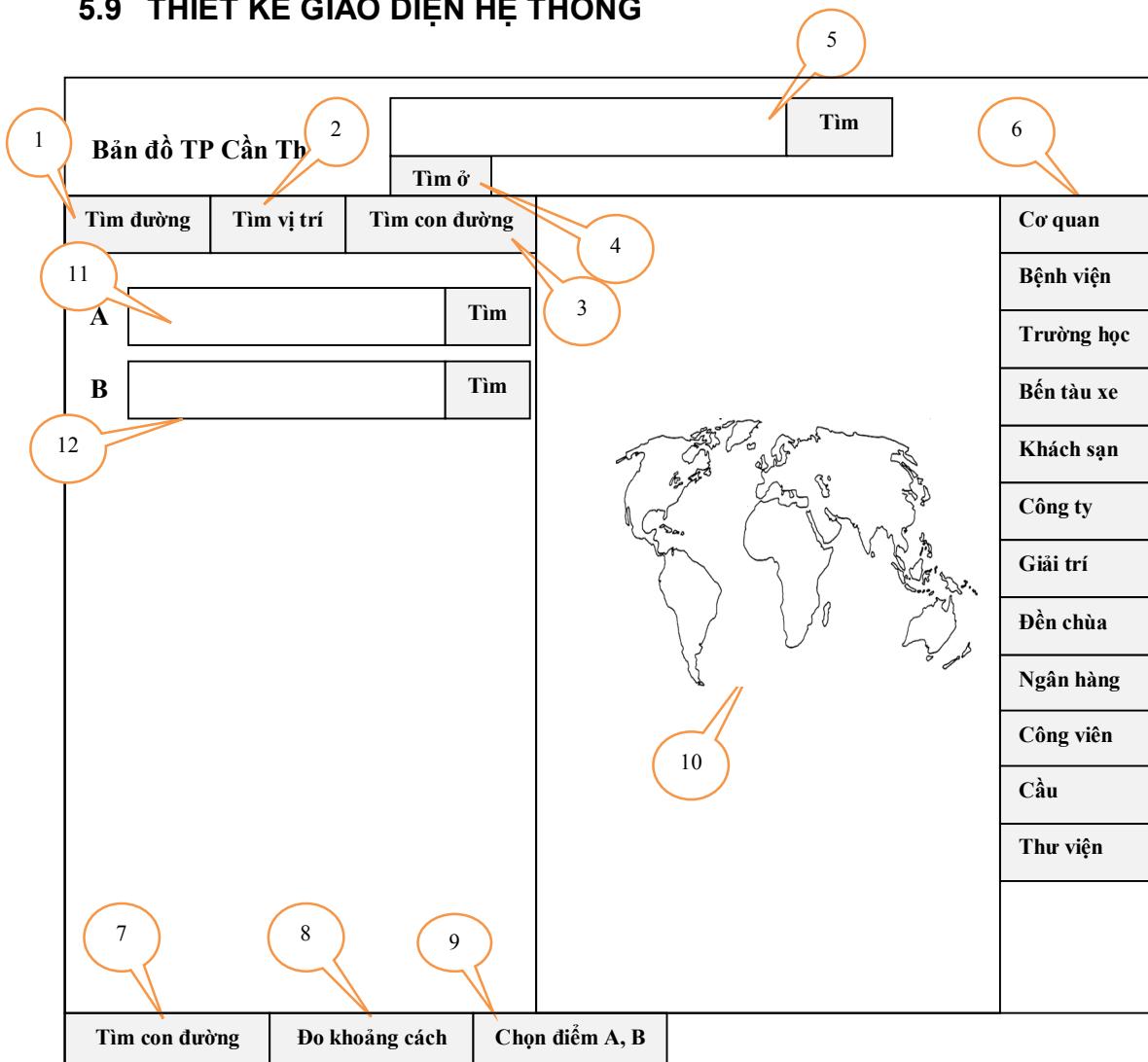
- *Lớp địa điểm (vector)*: lớp này sẽ hiện thị các địa điểm được tìm thấy khi gọi các dịch vụ tìm địa điểm.
- *Lớp con đường (vector)*: lớp này sẽ hiện thị kết quả của chức năng tìm kiếm con đường theo tên.

Việc chồng lớp bản đồ được thực hiện khá đơn giản, các lớp bản đồ được chia ra thành các đối tượng riêng biệt và được chồng lên nhau tạo thành bản đồ hoàn chỉnh.



Hình 5.3- Cơ chế hiển thị và chồng lớp dữ liệu

5.9 THIẾT KẾ GIAO DIỆN HỆ THỐNG



Hình 5.4- Thiết kế giao diện

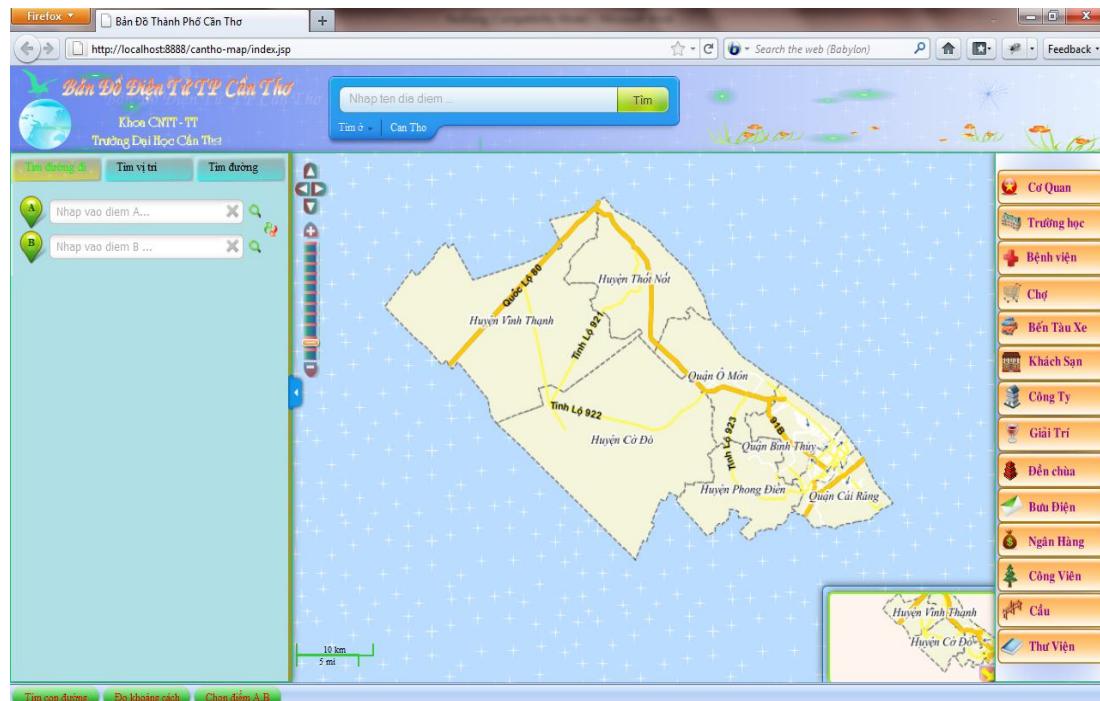
Nội dung của giao diện gồm 12 thành phần chính:

- (1): Cho phép người dùng tìm kiếm đường đi ngắn nhất giữa hai địa điểm được chọn trong các mục (11) và (12).
- (2): Hiển thị danh sách các địa điểm được tìm thấy cùng với các thông tin chi tiết của từng địa điểm.
- (3): Hiển thị danh sách các con đường được tìm thấy thông qua chức năng tìm con đường.

- (4): Cho phép người dùng chọn vùng giới hạn (quận/huyện hoặc phường/xã).
- (5): Cho phép tìm kiếm các địa điểm theo tên địa điểm và vùng giới hạn.
- (6): Cho phép tìm các địa điểm theo chủ đề.
- (7): Cho phép người dùng tìm các con đường theo tên.
- (8): Chức năng đo khoảng cách đường chim bay giữa hai điểm được chọn.
- (9): Cho phép người dùng chọn điểm bắt đầu và điểm kết thúc để thực hiện chức năng tìm đường đi ngắn nhất.
- (10): Hiển thị bản đồ tương ứng với các thao tác từ các mục (1), (2), (3), (5), (6), (7), (8), (9).
- (11): Chọn điểm bắt đầu để tìm đường đi.
- (12): Chọn điểm kết thúc để tìm đường đi.

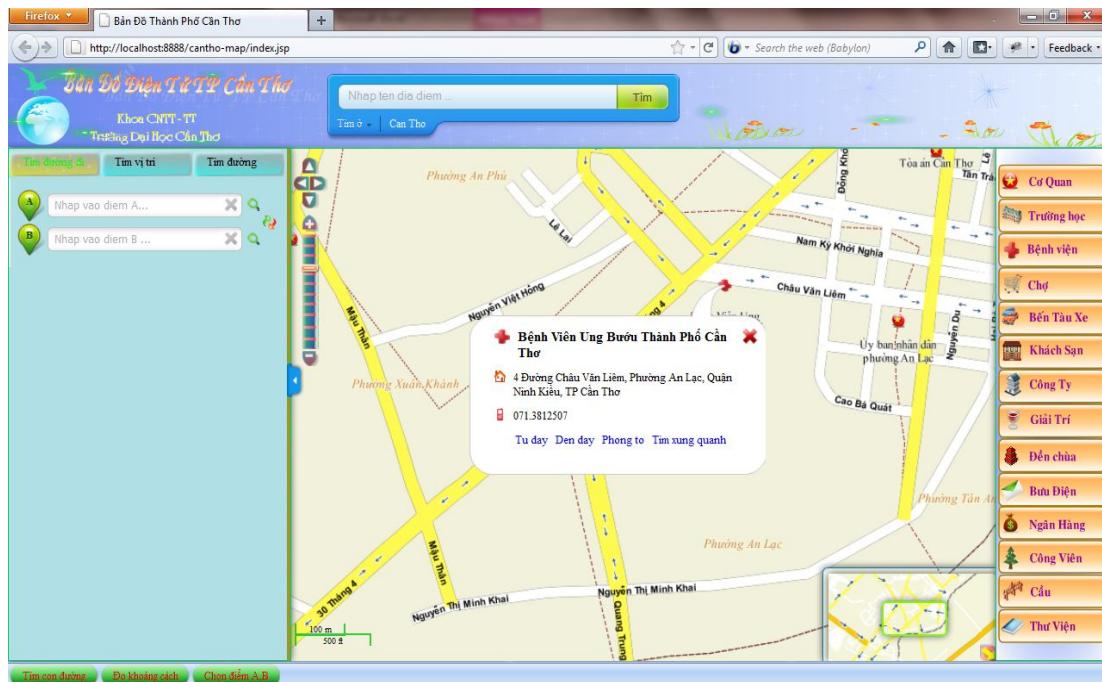
5.10 GIỚI THIỆU GIAO DIỆN HỆ THỐNG

Giao diện của hệ thống khi thực hiện các chức năng:

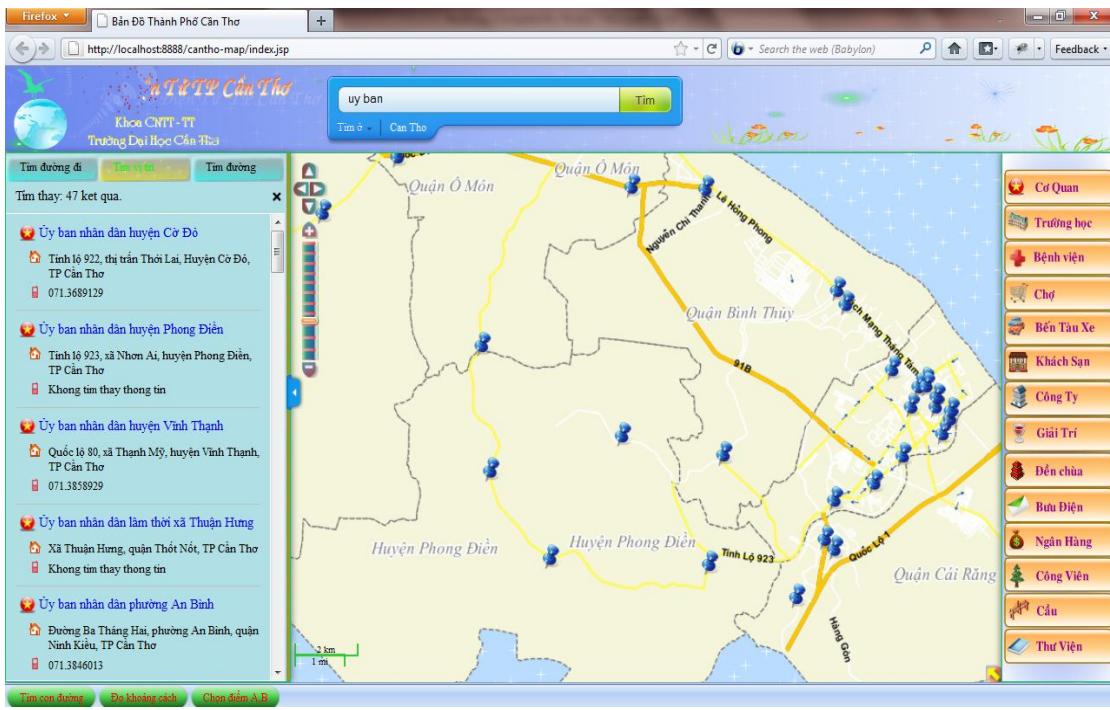


Hình 5.5- Giao diện chính của hệ thống

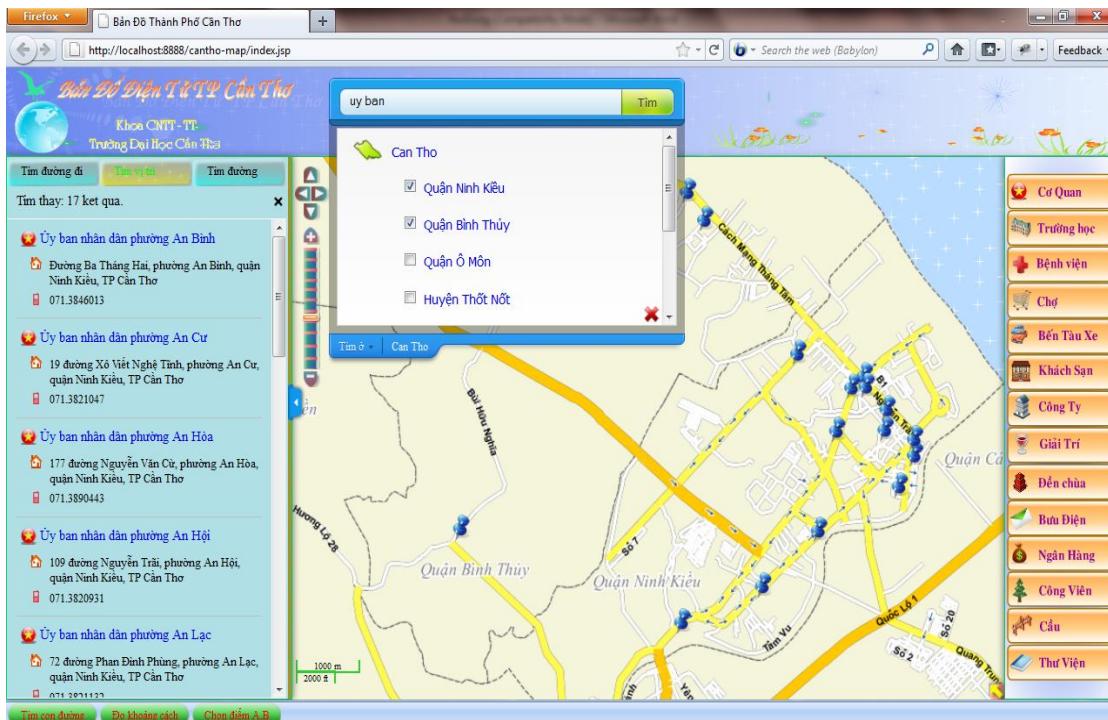
Chương 5- XÂY DỰNG HỆ THỐNG WEBGIS



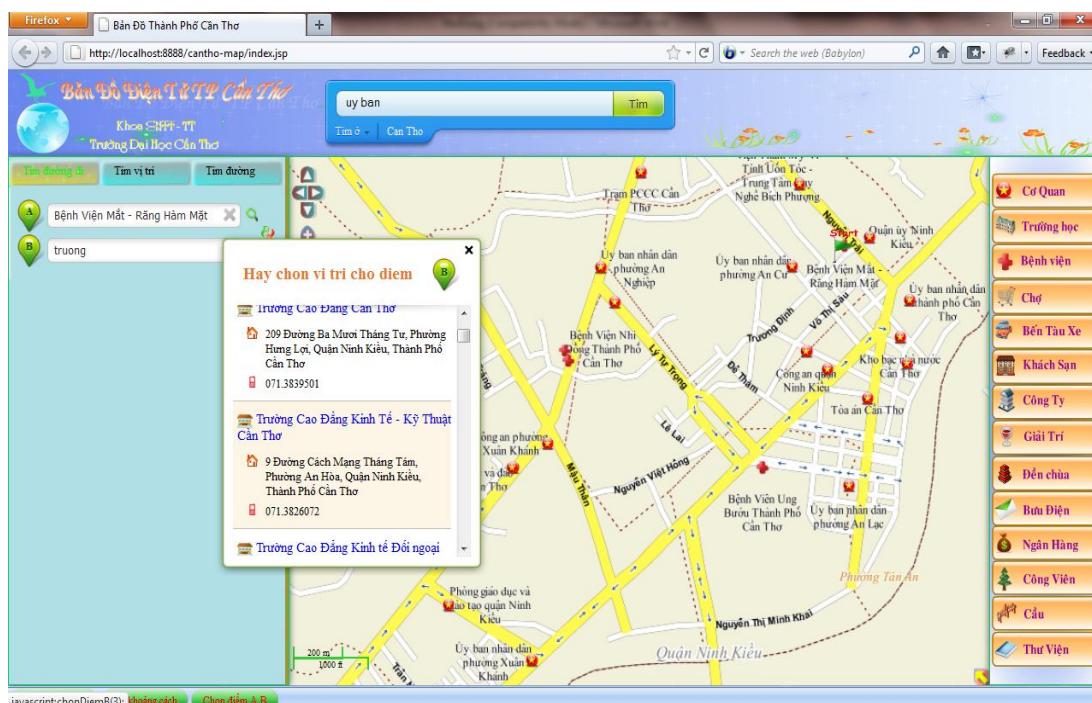
Hình 5.6- Hiển thị thông tin của một địa điểm được chọn



Hình 5.7- Tìm địa điểm theo tên

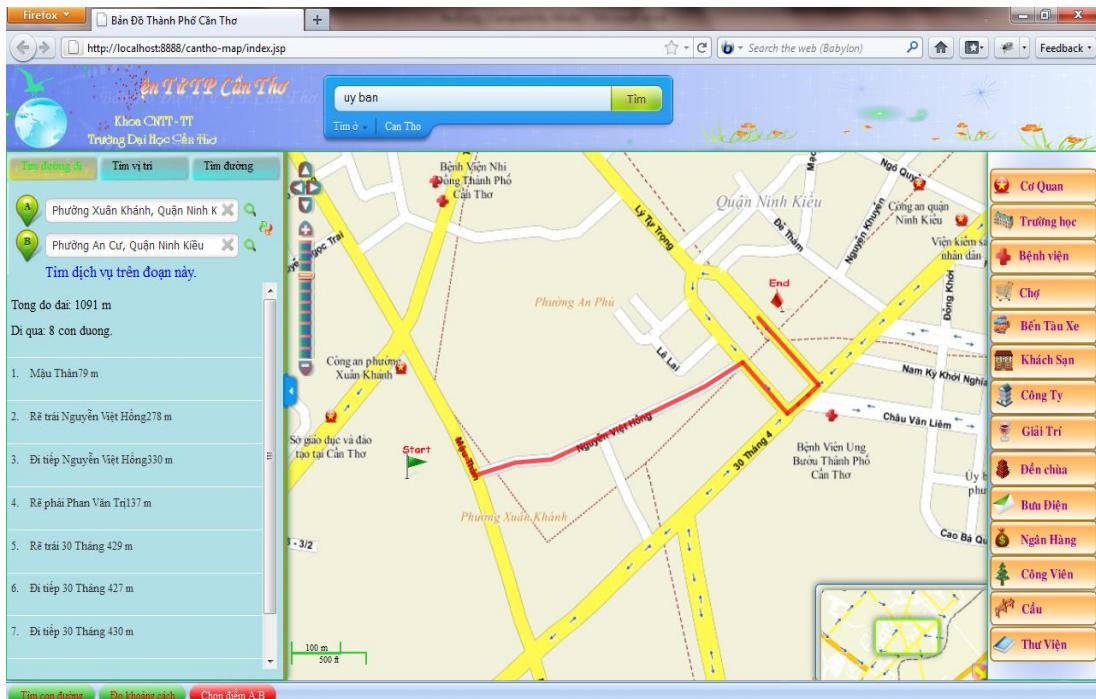


Hình 5.8- Chức năng tìm địa điểm theo vùng

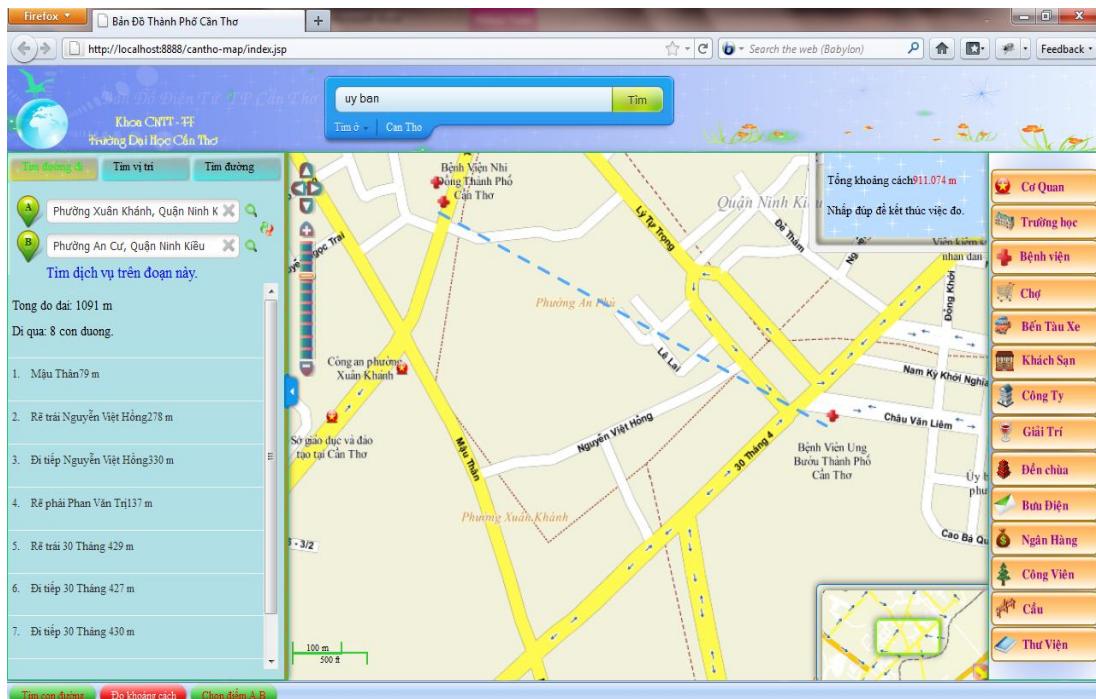


Hình 5.9- Chức năng chọn điểm bắt đầu và kết thúc của đường đi

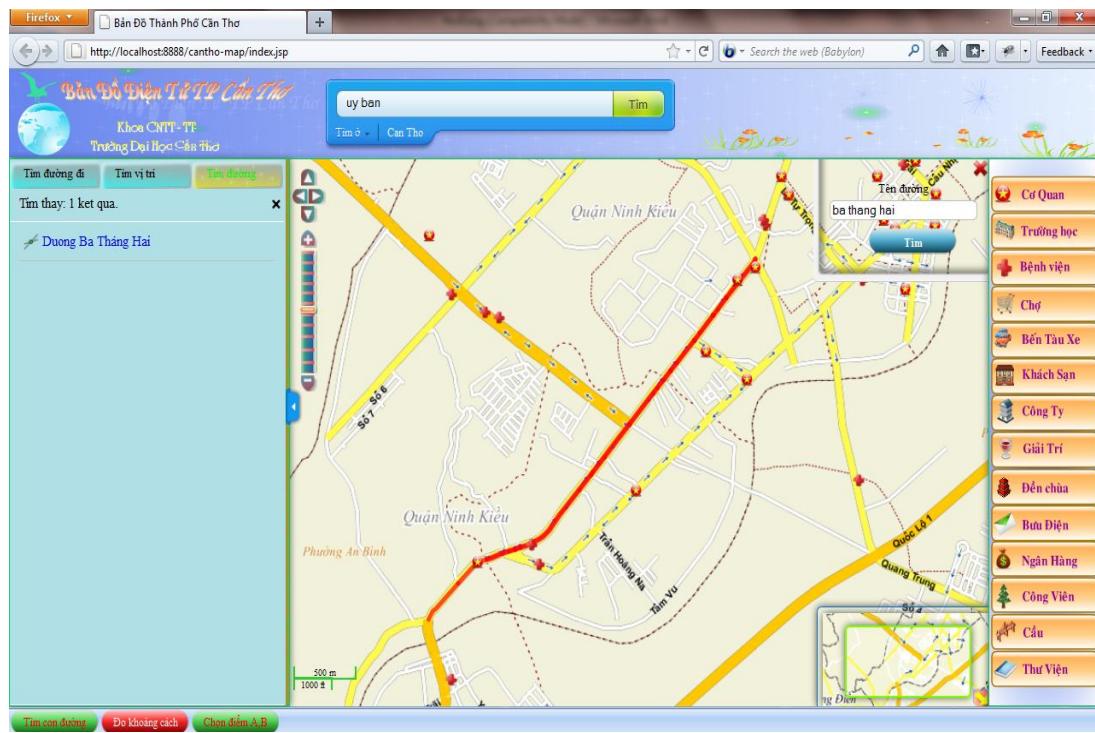
Chương 5- XÂY DỰNG HỆ THỐNG WEBGIS



Hình 5.10- Chức năng tìm đường đi



Hình 5.11- Đo khoảng cách đường chim bay



Hình 5.12- Chức năng tìm kiếm con đường

Chương 6 - KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

6.1 KẾT QUẢ ĐẠT ĐƯỢC

Qua quá trình nghiên cứu chúng em đã tiếp thu được nhiều kiến thức bổ ích về hệ thống thông tin địa lý theo công nghệ WebGIS. Chúng em cơ bản đã hoàn thành xong các vấn đề cốt yếu mà đề tài đã đặt ra:

- Làm chủ được công nghệ để phát triển WebGIS theo chuẩn OpenGIS.
- Xây dựng cơ sở dữ liệu địa lý dùng Mapinfo.
- Thao tác tốt trên hệ quản trị cơ sở dữ liệu không gian PostgreSQL (PostGIS).
- Khai thác được phần mềm mã nguồn mở GeoServer để tạo được dịch vụ web bản đồ phía máy chủ.
- Xây dựng được webservice với các dịch vụ:
 - Tìm kiếm đường đi sử dụng giải thuật Dijkstra.
 - Các dịch vụ tìm kiếm địa điểm.
- Gọi webservice dùng công nghệ Ajax.
- Hiển thị được bản đồ dùng Openlayers với các chức năng như: phóng to/thu nhỏ, cho biết tỉ lệ bản đồ, dịch chuyển bản đồ ở mức tổng quan và mức chi tiết.
- Định dạng lại các đối tượng trên bản đồ (con đường, cơ quan, bệnh viện, trường học....) bằng SLD với màu sắc và hình dạng hợp lý.
- Các chức năng cơ bản của một bản đồ số đều được thực hiện tốt:
 - Tìm kiếm địa điểm theo tên địa điểm.
 - Tìm kiếm địa điểm theo một hay nhiều quận huyện của Cần Thơ
 - Tìm kiếm địa điểm theo một hay nhiều phường xã thuộc một quận huyện của Cần Thơ.
 - Tìm kiếm địa điểm theo chủ đề.

- Tìm kiếm địa điểm theo tên và bán kính cho trước.
 - Tìm kiếm địa điểm theo bán kính trên đường đi đã tìm được.
 - Tìm kiếm con đường theo tên.
 - Tìm kiếm đường đi ngắn nhất với hai điểm cho trước.
 - Đo khoảng cách đường chim bay.
- Nâng vững hơn các kiến thức về: Java, JavaScript, CSS, HTML, JQuery.
 - Website hỗ trợ tìm kiếm đường đi, địa điểm trực quan, tiện dụng, tương đối đẹp và dễ sử dụng.
 - Nâng cao tính làm việc theo nhóm, khả năng tìm kiếm và đọc tài liệu tiếng anh.
 - Sử dụng được SVN để có thể dễ dàng quản lý phiên bản, tích hợp code giữa các thành viên trong nhóm và có thể làm việc trực tuyến, theo dõi tiến độ, công việc lẫn nhau.

6.2 HƯỚNG PHÁT TRIỂN

Trong tương lai hệ thống cần quản lý thêm các thông tin khác như đường cấm, đường đang thi công, sửa chữa để có thể ứng dụng được vào thực tế nhiều hơn. Thu thập thêm nhiều dữ liệu thuộc tính cho các địa điểm để làm cho hệ thống có thêm nhiều ý nghĩa về mặt thông tin, ví dụ như trọng tải của cầu, thông tin mô tả tổng quan về các địa điểm nếu có...

Bên cạnh đó cũng cần tìm hiểu thêm các kỹ thuật tìm kiếm khác để làm cho sự tìm kiếm theo chuỗi được linh hoạt và mềm dẻo hơn. Ngoài ra, cũng cần tìm hiểu các công nghệ trên nền tảng di động để có thể phát triển hệ thống trên điện thoại di động, PDA.

Hiện tại thì thuật toán tìm đường Dijkstra cho thời gian thực thi tương đối tốt trên tập dữ liệu 1303 đỉnh và 1896 cạnh. Tuy nhiên, do hạn chế về mặt thời gian thực hiện đề tài, nên chúng em không thể nghiên cứu kỹ các kỹ thuật tìm đường để có sự so sánh và mang lại kết quả tối ưu nhất về mặt thời gian. Dự định trong tương

lại có thể thay thuật toán Dijkstra bằng các thuật toán khác chẳng hạn như A* với hy vọng tốc độ thực hiện sẽ nhanh hơn.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] **Đào Ngọc Cảnh.** *Giáo trình hệ thống thông tin địa lý.* Tủ sách Đại Học Cần Thơ, 2003
- [2] **Lâm Quang Đốc.** *Bản đồ học.* NXB Đại Học Sư Phạm.
- [3] **Lê Đức Thắng.** *Giáo trình toán rời rạc 2.* Khoa CNTT & TT Đại Học Cần Thơ, 2009
- [4] **Nguyễn Văn An.** *Xây dựng dịch vụ web vẽ bản đồ trực tuyến.* Luận văn thạc sĩ, 2010
- [5] **Nguyễn Văn Kiệt.** *Xây dựng dịch vụ Web Feature Service theo chuẩn OpenGIS hỗ trợ hiển thị và các dịch vụ nối kết dữ liệu phi địa lý.* Luận văn thạc sĩ, 2010
- [6] **ThS. Nguyễn Đức Bình – ThS. Hoàng Hữu Cải – KS. Nguyễn Quốc Bình.** *Xây dựng bản đồ số hoá với MapInfo 6.0.* Tháng 3/2003

Tiếng Anh

- [7] **Open GIS Consortium, Inc.** *OpenGIS Simple Features Specification For SQL,* Revision 1.1
- [8] **GeoServer Project Steering Committee.** *GeoServer User Manual.* Release 2.0.2
- [9] **Refractons Research Inc.** *PostGIS 1.5.2 Manual*
- [10] **Steve Graham, Doug Davis, Simeon Simeonov, Glen Daniels, Peter Brittenham, Yuichi Nakamura, Paul Fremantle, Dieter König, Claudia Zentner.** *Building Web Services with Java,* Second Edition.
- [11] **Open GIS Consortium, Inc.** *Styled Layer Descriptor Implementation Specification,* Version 1.0

Website

- [12] Open Geospatial consortium, Inc, <http://www.opengeospatial.org/ogc>
- [13] GeoServer, <http://docs.geoserver.org/>
- [14] Openlayers, <http://www.openlayers.org/>
- [15] PostGIS, <http://postgis.refractions.net/>

- [16] <http://renaud.waldura.com/doc/java/dijkstra/>
- [17] <http://www.vietbando.com/maps/>
- [18] <http://gismoitruongct.vn>

PHỤ LỤC A - STYLED LAYER DESCRIPTOR IMPLEMENTATION SPECIFICATION

Đặc tả này quy định các định dạng của ngôn ngữ bản đồ, tạo kiểu dáng cho bản đồ địa lý theo định nghĩa của người dùng. Sau đây là nội dung chính của đặc tả này.

1. Layer

a. Layer Root Element

Một tài liệu SLD được định nghĩa như một chuỗi các layer đã được định kiểu trước. Phần tử gốc *StyledLayerDescriptor* được định nghĩa bằng đoạn XML Schema sau:

```
<xs:element name="StyledLayerDescriptor">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="sld:NamedLayer"/>
      <xs:element ref="sld:UserLayer"/>
    </xs:choice>
    <xs:attribute name="version" type="xs:string"
      use="required"/>
  </xs:complexType>
</xs:element>
```

Thuộc tính *version* cho biết version SLD của tài liệu, nhằm tạo điều kiện tương thích ngược với các tài liệu tĩnh được lưu trữ trong các phiên bản khác nhau của các đặc tả SLD. Các layer được định kiểu có thể tương ứng với các layer được đặt tên (*NamedLayer*) hoặc các layer được định nghĩa bởi người dùng (*UserLayer*).

b. Named Layers

Một layer được định nghĩa như là một tập các đối tượng, có thể là tiềm tàng của nhiều loại đối tượng khác nhau. Một layer được đặt tên là một layer có thể được truy xuất từ một OpenGIS Webserver sử dụng kiểu tên well-known.

Cơ chế đặc tả các lớp *named-layer* trong SLD được định nghĩa bởi đoạn XML-Schema sau:

```
<xs:element name="NamedLayer">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Name"/>
      <xs:element ref="sld:LayerFeatureConstraints" minOccurs="0"/>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="sld:NamedStyle"/>
        <xs:element ref="sld:UserStyle"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Name" type="xs:string"/>
```

Phần tử “*Name*” được xác định kiểu tên well-known của lớp đang được tham chiếu và bắt buộc phải có. Tất cả các well-known có thể có được xác định khả năng của tài liệu cho máy chủ.

Phần tử “*LayerFeatureConstraints*” là tùy chọn trong một *NameLayer* và cho phép người dùng xác định trên đối tượng của kiểu đối tượng được chọn bởi *name-layer* đang được tham chiếu. Nó thật sự là một bộ lọc cho phép lọc các đối tượng được hiển thị trong layer-name.

c. User defined layers

Ngoài việc sử dụng các lớp được đặt tên, người dùng có thể tự định nghĩa các lớp theo ý mình. Đoạn XML Schema bên dưới minh họa cho các lớp người dùng định nghĩa:

Trong một lớp được định nghĩa từ một tập hợp nhiều đối tượng, phần tử “UserLayer” phải cung cấp phương tiện để xác định các đối tượng được sử dụng. Tất cả các đối tượng được thừa nhận là lấy về từ một WFS hay một WCS

```
<xs:element name="UserLayer">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Name" minOccurs="0"/>
      <xs:element ref="sld:RemoteOWS" minOccurs="0"/>
      <xs:element ref="sld:LayerFeatureConstraints"/>
      <xs:element ref="sld:UserStyle" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

2. User-Defined Styles

Đoạn XML-Schema bên dưới minh họa cho phần tử *UserStyle* SLD:

```

<xs:element name="UserStyle">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Name" minOccurs="0"/>
      <xs:element ref="sld>Title" minOccurs="0"/>
      <xs:element ref="sld:Abstract" minOccurs="0"/>
      <xs:element ref="sld:IsDefault" minOccurs="0"/>
      <xs:element ref="sld:FeatureTypeStyle" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Title" type="xs:string"/>
<xs:element name="Abstract" type="xs:string"/>
<xs:element name="IsDefault" type="xs:string"/>

```

Ở mức độ ngữ nghĩa thì một UserStyle cũng giống như một NamedStyle được sử dụng trong ngữ cảnh của một WMS. Trong ý nghĩa đó một kiểu NamedStyle có thể được dùng như một tham chiếu đến một UserStyle ẩn mà UserStyle này đã được lưu trữ bên trong của một máy chủ bản đồ.

3. FeatureTypeStyles

FeatureTypeStyle định nghĩa các kiểu được áp dụng cho một loại đối tượng duy nhất của một lớp. Phần tử này cũng có thể được tái sử dụng bên ngoài phạm vi của WMS và các lớp. Nó được định nghĩa như sau:

```

<xs:element name="FeatureTypeStyle">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Name" minOccurs="0"/>
      <xs:element ref="sld>Title" minOccurs="0"/>
      <xs:element ref="sld:Abstract" minOccurs="0"/>
      <xs:element ref="sld:FeatureTypeName" minOccurs="0"/>
      <xs:element ref="sld:SemanticTypeIdentifier" minOccurs="0"
        maxOccurs="unbounded"/>
      <xs:element ref="sld:Rule" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SemanticTypeIdentifier" type="xs:string"/>

```

Phản tử FeatureTypeStyle cho phép phân chia rõ ràng việc xử lý các layer và việc xử lý các đối tượng. Khái niệm layer là duy nhất trong WMS và SLD nhưng đối tượng thì được sử dụng chung, chẳng hạn như trong WFS và GML. Do đó sự phân chia rõ ràng này là quan trọng.

4. Rule

“Rules” được sử dụng để qui định cách thức hiển thị dữ liệu dựa trên thuộc tính của đối tượng không gian và tỷ lệ bản đồ. *Rule* là một chỉ thị định nghĩa kiểu dáng duy nhất. Nó được áp dụng cho toàn bộ một lớp, hoặc có thể có các bộ lọc luận lý được kết hợp với nó để có thể định nghĩa kiểu dáng theo điều kiện. Những điều kiện lọc này có thể dựa trên dữ liệu thuộc tính của đối tượng hoặc dựa trên tỷ lệ phóng to thu nhỏ của bản đồ. Trong một *FeatureTypeStyle* có thể có nhiều *rule*, mỗi *rule* sẽ qui định kiểu dáng cho một số đối tượng thỏa điều kiện lọc của *rule* đó. Định dạng của *rule* được định nghĩa trong lược đồ sau:

```

<xs:element name="Rule">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Name" minOccurs="0"/>
      <xs:element ref="sld>Title" minOccurs="0"/>
      <xs:element ref="sld:Abstract" minOccurs="0"/>
      <xs:element ref="sld:LegendGraphic" minOccurs="0"/>
      <xs:choice minOccurs="0">
        <xs:element ref="ogc:Filter"/>
        <xs:element ref="sld:ElseFilter"/>
      </xs:choice>
      <xs:element ref="sld:MinScaleDenominator" minOccurs="0"/>
      <xs:element ref="sld:MaxScaleDenominator" minOccurs="0"/>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="sld:LineSymbolizer"/>

```

5. Symbolizer

Symbolizer được nhúng bên trong các rule mà rule này gom nhóm các điều kiện cho việc định kiểu các đối tượng. Một *Symbolizer* mô tả bằng cách nào một đối tượng được xuất hiện trên bản đồ. Một *Symbolizer* không chỉ mô tả hình dạng mà còn mô tả các thuộc tính đồ họa như màu sắc, độ trong suốt. Một ký hiệu được thu được bằng cách xác định một trong các kiểu *Symbolizer*, sau đó cung cấp các tham số để ghi đè mặc định của nó. Hiện tại có năm *Symbolizer* được định nghĩa: *Line*, *Polygon*, *Point*, *Text*, *Raster*

- **Line Symbolizer**

Ví dụ:

```
<LineSymbolizer>
  <Geometry>
    <ogc:PropertyName>centerline</ogc:PropertyName>
  </Geometry>
  <Stroke>
    <CssParameter name="stroke">#0000ff</CssParameter>
    <CssParameter name="stroke-width">2</CssParameter>
  </Stroke>
</LineSymbolizer>
```

Format: một *LineSymbolizer* dùng để định kiểu cho một “*stroke* (nét)” hay một loại hình học tuyến tính ví dụ như là một chuỗi các đoạn thẳng. Nó được định nghĩa đơn giản sau đây:

```
<xs:element name="LineSymbolizer">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Geometry" minOccurs="0"/>
      <xs:element ref="sld:Stroke" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Geometry: Phần tử *Geometry* của *LineSymbolizer* dùng để định nghĩa các đối tượng hình học tuyến tính được định kiểu. Phần tử *Geometry* là tùy chọn, nếu vắng mặt thì thuộc tính “*default*” của kiểu đối tượng được sử dụng bên trong *FeatureStyleType*. Thuộc tính *default* thì tùy thuộc vào hệ thống. Thường thì một đối tượng chỉ có duy nhất một *geometry*. Định dạng của một *Geomerty* như sau:

```

<xs:element name="Geometry">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ogc:PropertyName"/>  </xs:sequence>
    </xs:complexType>
  </xs:element>

```

Stroke: Phần tử *Stroke* của *LineSymbolizer* bao đóng các tham số đồ họa của các đối tượng hình học tuyến tính. Định dạng của các phần tử *Stroke* như sau:

```

<xs:element name="Stroke">
  <xs:complexType>
    <xs:sequence>
      <xs:choice minOccurs="0">
        <xs:element ref="sld:GraphicFill"/>
        <xs:element ref="sld:GraphicStroke"/>
      </xs:choice>
      <xs:element ref="sld:CssParameter" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

▪ Polygon Symbolizer

Một *Polygon Symbolizer* được dùng để vẽ, tô nền và tô biên cho một đa giác. Nó được định nghĩa như sau:

```
<xs:element name="PolygonSymbolizer">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Geometry" minOccurs="0"/>
      <xs:element ref="sld:Fill" minOccurs="0"/>
      <xs:element ref="sld:Stroke" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Fill: Phần tử *Fill* cho phép tô nền cho một đối tượng hình học. Nó được định nghĩa như sau:

```
<xs:element name="Fill">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:GraphicFill" minOccurs="0"/>
      <xs:element ref="sld:CssParameter" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

▪ Point Symbolizer

Format: Một *PointSymbolizer* được dùng để vẽ “*Graphic*” tại một điểm. Nó được định nghĩa đơn giản như sau:

```

<xs:element name="PointSymbolizer">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Geometry" minOccurs="0"/>
      <xs:element ref="sld:Graphic" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Graphic: Một *Graphic* là một “biểu tượng đồ họa” với một màu sắc, kích thước và hình dạng. Một *Graphic* có thể định nghĩa một “bức hình nhỏ” và có thể là *raster* hay *vector*. Định nghĩa ở mức cao của một đối tượng đồ họa như sau:

```

<xs:element name="Graphic">
  <xs:complexType>
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="sld:ExternalGraphic"/>
        <xs:element ref="sld:Mark"/>
      </xs:choice>
    <xs:sequence>
      <xs:element ref="sld:Opacity" minOccurs="0"/>
      <xs:element ref="sld:Size" minOccurs="0"/>
      <xs:element ref="sld:Rotation" minOccurs="0"/>
    </xs:sequence>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

▪ Text Symbolizer

Format: Một *TextSymbolizer* được dùng để định kiểu cho các nhãn văn bản. Nó được định nghĩa như sau:

```
<xs:element name="TextSymbolizer">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Geometry" minOccurs="0"/>
      <xs:element ref="sld:Label" minOccurs="0"/>
      <xs:element ref="sld:Font" minOccurs="0"/>
      <xs:element ref="sld:LabelPlacement" minOccurs="0"/>
      <xs:element ref="sld:Halo" minOccurs="0"/>
      <xs:element ref="sld:Fill" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Label: Phần tử *Label* được sử dụng để cung cấp nội dung của text-label. Nó được định nghĩa như sau:

```
<xs:element name="Label" type="sld:ParameterValueType"/>
```

Font: Phần tử *Font* cho phép xác định họ phông nhất định, kiểu phông, kích thước. Nó được định nghĩa như sau:

```
<xs:element name="Font">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element ref="sld:CssParameter" minOccurs="0"  
                 maxOccurs="unbounded"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

LabelPlacement: Các phần tử *LabelPlacement* được sử dụng để vị trí một nhãn liên quan đến một điểm hoặc một chuỗi các đoạn thẳng và được định nghĩa như sau:

```
<xs:element name="LabelPlacement">  
  <xs:complexType>  
    <xs:choice>  
      <xs:element ref="sld:PointPlacement"/>  
      <xs:element ref="sld:LinePlacement"/>  
    </xs:choice>  
  </xs:complexType>  
</xs:element>
```

Halo: Phần tử là một kiểu của *Fill* được áp dụng cho màu nền của phông. Công dụng của halo là để làm tăng tính dễ đọc của các nhãn. Một *halo* được định nghĩa như sau:

```
<xs:element name="Halo">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Radius" minOccurs="0"/>
      <xs:element ref="sld:Fill" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

▪ Raster Symbolizer

Format: *RasterSymbolizer* cho biết hiển thị dữ liệu *raster* như thế nào. Nó được định nghĩa như sau:

```
<xs:element name="RasterSymbolizer">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:Geometry" minOccurs="0"/>
      <xs:element ref="sld:Opacity" minOccurs="0"/>
      <xs:element ref="sld:ChannelSelection" minOccurs="0"/>
      <xs:element ref="sld:OverlapBehavior" minOccurs="0"/>
      <xs:element ref="sld:ColorMap" minOccurs="0"/>
      <xs:element ref="sld:ContrastEnhancement" minOccurs="0"/>
      <xs:element ref="sld:ShadedRelief" minOccurs="0"/>
      <xs:element ref="sld:ImageOutline" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

PHỤ LỤC B - OPENGIS SIMPLE FEATURES SPECIFICATION FOR SQL

1. Khái niệm

Đặc tả OpenGIS cho SQL định nghĩa một chuẩn lược đồ SQL hỗ trợ việc lưu trữ, sử dụng, truy vấn và cập nhật tập hợp các đối tượng không gian thông qua ODBC API. Một đối tượng đơn giản được xác định trong bảng đặc tả OpenGIS sẽ bao gồm hai thuộc tính: không gian và phi không gian. Các thuộc tính không gian là các đối tượng hình học, những đối tượng đơn giản sẽ dựa trên hình học 2D với phép suy diễn tuyến tính giữa hai đỉnh.

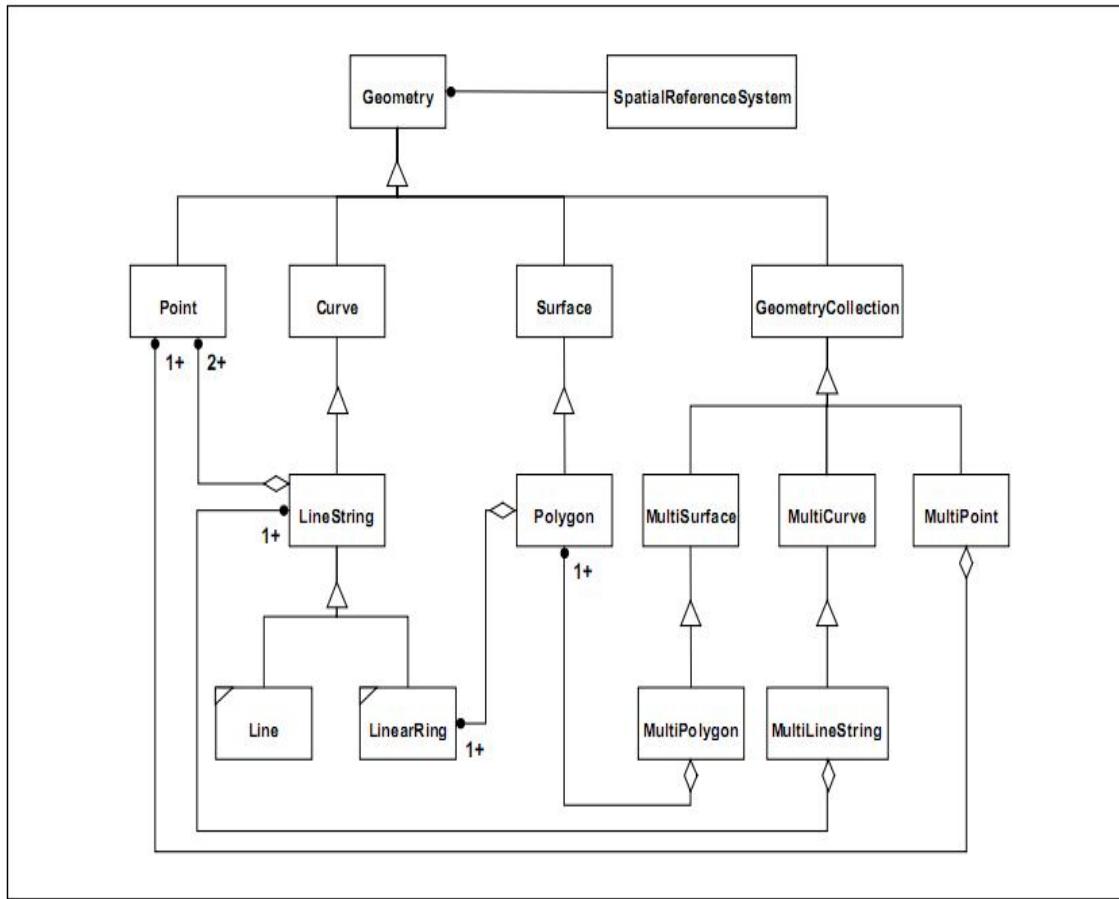
Tập hợp các đối tượng không gian đơn giản được lưu trữ như các bảng trong hệ quản trị cơ sở dữ liệu quan hệ với một số cột có giá trị dữ liệu hình học, mỗi đối tượng được lưu trữ như một dòng trong bảng. Các thuộc tính phi không gian của đối tượng sẽ được ánh xạ vào các cột mà kiểu dữ liệu của nó theo chuẩn ODBC/SQL92. Các thuộc tính không gian của đối tượng sẽ được ánh xạ vào các cột mà các kiểu dữ liệu SQL của nó dựa trên các kiểu dữ liệu hình học được bổ sung vào cho SQL. Một bảng mà các dòng của nó đại diện cho các đối tượng theo chuẩn OpenGIS sẽ được tham khảo như là một bảng đối tượng. Một bảng như thế sẽ chứa đựng một hoặc nhiều cột có giá trị hình học.

2. Mô hình đối tượng hình học (Geometry Object Mode)

Mô hình các đối tượng hình học theo chuẩn OpenGIS được trình bày như hình bên dưới. Lớp cơ sở *Geometry* có các lớp con là *Point*, *Curve*, *Surface* và *Geometry Collection*. Mỗi đối tượng hình học sẽ được liên kết với một hệ qui chiếu (*Spatial Reference System*), hệ này mô tả tọa độ không gian mà các đối tượng được định nghĩa.

Hình PL.B.1 được dựa trên việc mở rộng các mô hình hình học được qui định trong OpenGIS Abstracts Specification với các lớp chuyên biệt 0, 1 hoặc 2 chiều như: *MultiPoint*, *MultiLineString* và *MultiPolygon* cho mô hình hình học tương ứng với các tập hợp *Points*, *LineStrings* và *Polygons*. *MultiCurve* và *MultiSurface* được

giới thiệu như các lớp trùu tượng tổng quát hóa của các giao tiếp như: *Curves* và *Surfaces*.



Hình PL.B.1 - Geometry Class Hierarchy

a. Geometry

Geometry là lớp cơ sở của hệ phân cấp và là một lớp trùu tượng. Các lớp con của *Geometry* có thể được định nghĩa theo đối tượng hình học 0, 1, 2 chiều trong hệ tọa độ không gian 2 chiều.

Các phương thức cơ bản của *Geometry*:

- *Dimension()*: Integer - Chiều hiện có của *Geometry* này nhỏ hơn chiều của hệ tọa độ.
- *GeometryType()*: String - Kết quả trả về là chuỗi tên kiểu *Geometry* của đối tượng đang xét.

- *SRID()*: Integer - Kết quả trả về *id* hệ qui chiếu không gian của *Geometry* này.
- *Envelope()*: Geometry - Vùng bao nhỏ nhất của *Geometry* này, kết quả trả về là một *Geometry*. Các *Polygon* được định nghĩa bởi các điểm gốc của vùng bao((*MINX, MINY*), (*MAXX, MINY*), (*MAXX, MAXY*), (*MINX, MAXY*), (*MINX, MINY*)).
- *AsText()*: String - Chuyển *Geometry* sang định dạng well-known text đại diện cho nó.
- *AsBinary()*: Binary - Chuyển *Geometry* sang định dạng well-known binary đại diện cho nó.
- *IsEmpty()*: Integer - Kết quả trả về 1 (TRUE) nếu *Geometry* này là rỗng . Nếu đúng, thì sau đó *Geometry* này sẽ được thiết đặt là một điểm rỗng, \emptyset , theo hệ tọa độ không gian.
- *IsSimple()*: Integer - Kết quả trả về 1 (TRUE) nếu *Geometry* này không là một kiểu hình học đặc biệt, như là tự giao hay tự cắt. Mô tả của mỗi lớp hình học sẽ bao gồm các điều kiện tạo ra thể hiện của lớp này được xếp vào loại không đơn giản.
- *Boundary()*: Geometry - Kết quả trả về tổ hợp vùng biên của *Geometry* này.

Phương thức kiểm tra mối quan hệ giữa các đối tượng hình học

- *Equals(anotherGeometry:Geometry)*: Integer - Kết quả trả về 1 (TRUE) nếu *geometry* này có không gian bằng với *anotherGeometry*.
- *Disjoint(anotherGeometry:Geometry)*: Integer - Kết quả trả về 1 (TRUE) nếu *geometry* này có không gian không giao với *anotherGeometry*.
- *Intersects(anotherGeometry:Geometry)*: Integer - Kết quả trả về 1 (TRUE) nếu *geometry* này có không gian giao với *anotherGeometry*.
- *Touches(anotherGeometry:Geometry)*: Integer - Kết quả trả về 1 (TRUE) nếu *geometry* này có không gian tiếp xúc với *anotherGeometry*.

- *Crosses(anotherGeometry:Geometry): Integer* - Kết quả trả về 1 (TRUE) nếu *geometry* này có không gian cắt với *anotherGeometry*.
- *Within(anotherGeometry:Geometry): Integer* - Kết quả trả về 1 (TRUE) nếu *geometry* này có không gian nằm trong *anotherGeometry*.
- *Contains(anotherGeometry:Geometry): Integer* - Kết quả trả về 1 (TRUE) nếu *geometry* này có không gian chứa *anotherGeometry*.
- *Overlaps(anotherGeometry:Geometry): Integer* - Kết quả trả về 1 (TRUE) nếu *geometry* này có không gian nằm chồng lên không gian của *anotherGeometry*.
- *Relate(anotherGeometry:Geometry,intersectionPatternMatrix:String): Integer* - Kết quả trả về 1 (TRUE) nếu *Geometry* này có không gian quan hệ với *anotherGeometry*.

Phương thức hỗ trợ phân tích không gian

- *Distance(anotherGeometry:Geometry): Double* - Kết quả trả về khoảng cách ngắn nhất giữa 2 điểm trong hệ qui chiếu không gian của *Geometry* đang xét.
- *Buffer(distance:Double): Geometry* - Kết quả trả về một *Geometry* đại diện cho tất cả các điểm mà khoảng cách từ chúng đến *Geometry* đang xét nhỏ hơn hoặc bằng *distance*. Việc tính toán dựa trên hệ qui chiếu không gian của đối tượng đang xét.
- *ConvexHull(): Geometry* - Kết quả trả về một *Geometry* đại diện cho phần lồi của *Geometry* đang xét.
- *Intersection(anotherGeometry:Geometry): Geometry* - Kết quả trả về một *Geometry* đại diện cho tập hợp các điểm giao của *Geometry* đang xét với *anotherGeometry*.
- *Union(anotherGeometry:Geometry): Geometry* - Kết quả trả về một *Geometry* đại diện cho tập hợp các điểm là hợp của *Geometry* đang xét với *anotherGeometry*.

- *Difference(anotherGeometry:Geometry)*: *Geometry* - Kết quả trả về một *Geometry* đại diện cho tập hợp các điểm thuộc *Geometry* đang xét và không thuộc *anotherGeometry*.
- *SymDifference(anotherGeometry:Geometry)*: *Geometry* - Kết quả trả về một *Geometry* đại diện cho tập hợp các điểm đối xứng khác biệt của *Geometry* đang xét với *anotherGeometry*.

b. Geometry Collection

GeometryCollection là một đối tượng hình học bao gồm một hoặc một tập các đối tượng hình học con. Tất cả các thành phần con trong *GeometryCollection* phải được trong cùng một không gian tham chiếu. Đây cũng là tham chiếu không gian cho *GeometryCollection* này.

Các phương thức

- *NumGeometries()*: *Integer* - Kết quả trả về số lượng *Geometry* nằm trong *GeometryCollection*.
- *GeometryN(N:integer)*: *Geometry* - Kết quả trả về *Geometry* thứ N nằm trong *GeometryCollection*.

c. Point

Một *Point* là một *geometry* 0 chiều đại diện cho một vị trí đơn lẻ trên hệ tọa độ không gian. Một *Point* có một tọa độ x và một tọa độ y.

Vùng bao của một *Point* là một tập rỗng.

Các phương thức

- *X()*: *Double* - Kết quả trả về giá trị tọa độ x của *Point* đang xét.
- *Y()*: *Double* - Kết quả trả về giá trị tọa độ y của *Point* đang xét.

d. MultiPoint

Một *MultiPoint* là một tập các đối tượng hình học 0 chiều. Các thành phần con của *MultiPoint* là các *Point*.

Một *MultiPoint* là đơn giản nếu không chứa 2 *Point* bằng nhau.

Vùng bao của *MultiPoint* là một tập rỗng.

e. *Curve*

Một *Curve* là một đối tượng hình học một chiều chứa một chuỗi các *Points* với *subtype* của *Curve* quy định cụ thể hình thức nội suy giữa các điểm. Một *Curve* được gọi là đóng nếu nó có điểm đầu và điểm cuối trùng nhau. Vùng bao của một *Curve* có điểm đầu và điểm cuối trùng nhau là rỗng.

Một *Curve* đơn giản có điểm đầu và điểm cuối trùng nhau tạo thành một *Ring*.

Vùng bao của một *Curve* có điểm đầu và điểm cuối không trùng nhau sẽ chứa hai điểm cuối của nó.

Một *Curve* được định nghĩa như một hình trạng đóng.

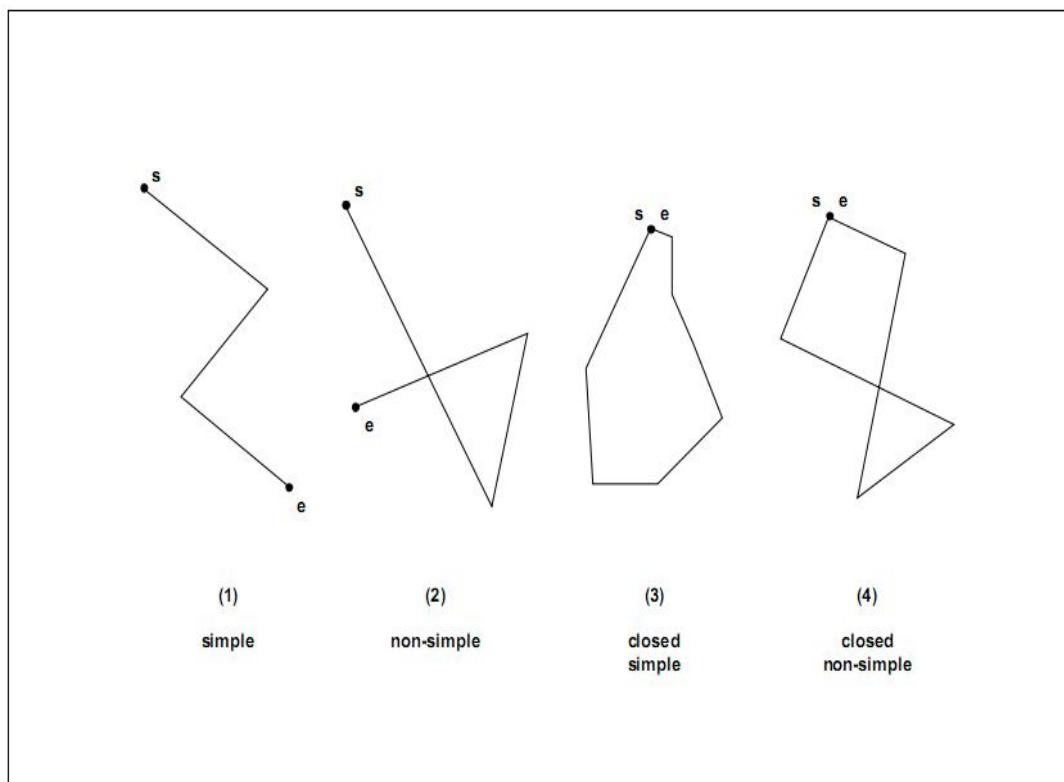
Các phương thức

- *Length()*: *Double* - Chiều dài của *Curve* đang xét trong hệ qui chiếu không gian của nó.
- *StartPoint()*: *Point* - Điểm bắt đầu của *Curve* đang xét.
- *EndPoint()*: *Point* - Điểm kết thúc của *Curve* đang xét.
- *IsClosed()*: *Integer* - Kết quả trả về 1 (*TRUE*) nếu *Curve* đang xét có điểm đầu và điểm cuối trùng nhau (*StartPoint()* = *EndPoint()*).
- *IsRing()*: *Integer* - Kết quả trả về 1 (*TRUE*) nếu *Curve* đang xét có điểm đầu và điểm cuối trùng nhau (*StartPoint()* = *EndPoint()*) và *Curve* này là đối tượng đơn giản (không đi qua một điểm 2 lần)

f. *LineString, Line, LinearRing*

Một *LineString* là một *Curve* với phép suy diễn tuyến tính giữa các điểm. Mỗi cặp liên tiếp của các điểm định nghĩa một đoạn thẳng. Một *Line* là một *LineString* với chính xác hai điểm. Một *LinearRing* là một *LineString* đóng và đơn giản. *Curve*

trong hình PL.B.2-(3) là một *LineString* đóng và cũng là một *LinearRing*. *Curve* trong hình PL.B.2-(4) là một *LineString* đóng nhưng không là một *LinearRing*



Hình PL.B.2 - (1) simple LineString, (2) non-simple LineString, (3) simple, closed LineString (LinearRing), (4) non-simple closed LineString

Các phương thức

- *NumPoints(): Integer* - Số lượng các *Point* thuộc *LineString* đang xét.
- *PointN(N:Integer): Point* - Kết quả trả về là *Point* thứ N của *LineString* đang xét.

g. MultiCurve

Một *MultiCurve* là một đối tượng *GeometryCollection* một chiều mà các phần tử của nó là các *Curve* (Hình PL.B.3).

Một *MultiCurve* là đơn giản nếu và chỉ nếu tất cả các phần tử của nó là đơn giản, giao giữa hai phần tử bất kỳ phải nằm trên điểm thuộc vùng bao của cả hai phần tử đó.

Vùng bao của *MultiCurve* có được bằng cách áp dụng qui tắc hợp “Mod 2”: điểm A nằm trong ranh giới của một *MultiCurve* nếu nó đang ở trong ranh giới của một số lẻ của các phần tử của *MultiCurve*

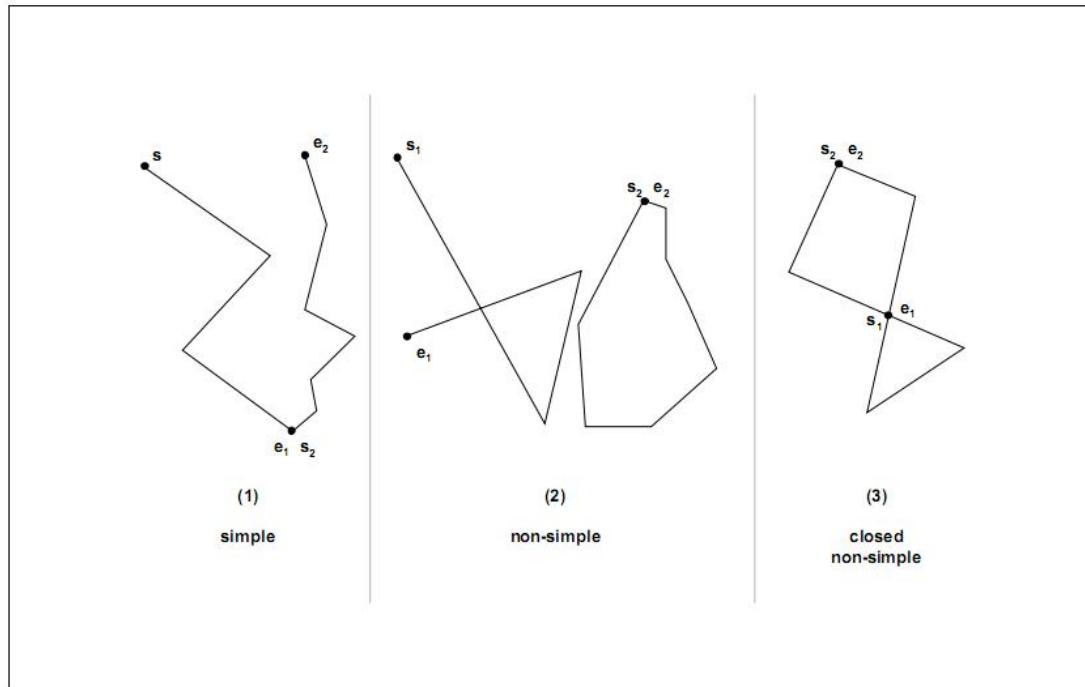
Một *MultiCurve* là đóng nếu tất cả các phần tử của nó đều đóng. Vùng bao của một *MultiCurve* đóng luôn luôn rỗng.

Các phương thức

- *IsClosed()*: *Integer* - Kết quả trả về 1 (*TRUE*) nếu *MultiCurve* đang xét là đóng (*StartPoint() = EndPoint()*)
- *Length()*: *Double* - Độ dài của *MultiCurve* đang xét, độ dài này bằng với tổng độ dài của các phần tử *Curve*.

h. MultiLineString

Một *MultiLineString* là một *MultiCurve* mà các phần tử của nó là *LineStrings*.



Hình PL.B.3 - (1) simple MultiLineString, (2) non-simple MultiLineString với 2 phần tử, (3) non-simple, MultiLineString đóng với 2 phần tử.

i. Surface

Một *Surface* là một đối tượng hình học hai chiều

Trong OpenGIS Abstract Specification định nghĩa một *Surface* đơn giản bao gồm các ‘patch’ đơn kết hợp với một ‘exterior boundary’ và 0 hoặc nhiều hơn vùng bao ‘interior’. Các *Surfaces* đơn giản trong không gian ba chiều thì đẳng cấu với các *surfaces* phẳng.

Vùng bao của một *Surface* đơn giản là tập hợp các đường cong khép kín tương ứng với các vùng bao ‘exterior’ và ‘interio’ của nó.

Các phương thức

- *Area()*: *Double* - Diện tích của *Surface* đang xét và được tính trong hệ qui chiếu không gian của chính nó.
- *Centroid()*: *Point* – Trọng tâm toán học của *Surface* này là một *Point*. Kết quả có thể không nằm trên *Surface* này.
- *PointOnSurface()*: *Point* – Kết quả trả về một điểm nằm trên *Surface* này.

j. Polygon

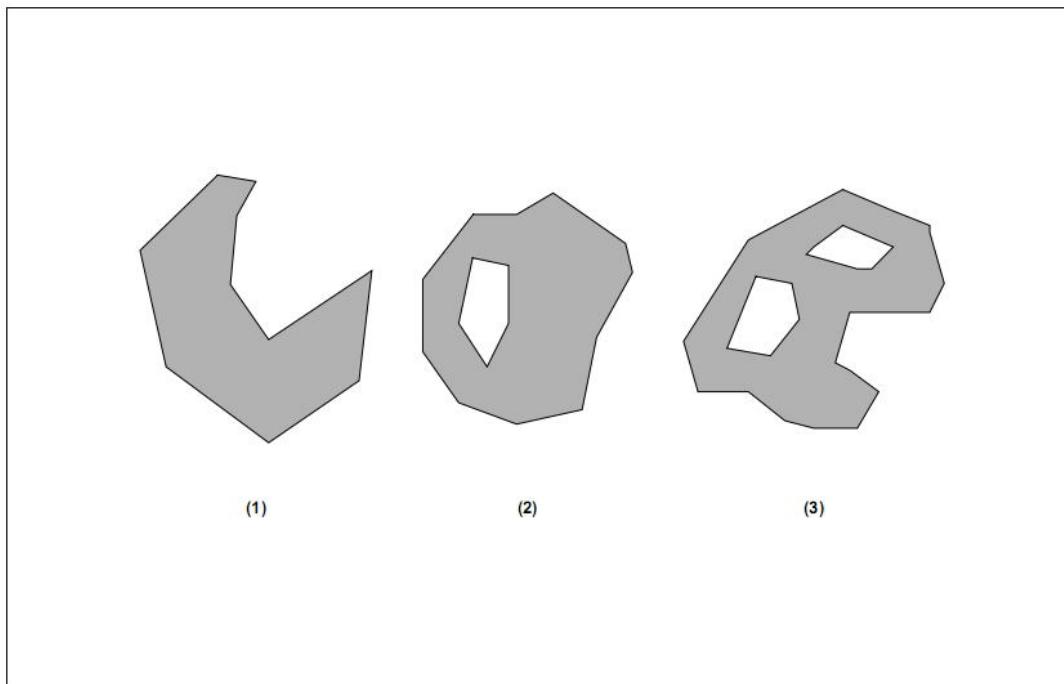
Một *Polygon* là một *Surface* phẳng, được định nghĩa bởi một vùng bao “exterior” và 0 hoặc nhiều vùng bao “interior”. Mỗi vùng bao “interior” định nghĩa một lỗ trong *Polygon*.

Khẳng định cho đa giác (các quy tắc xác định đa giác hợp lệ) là:

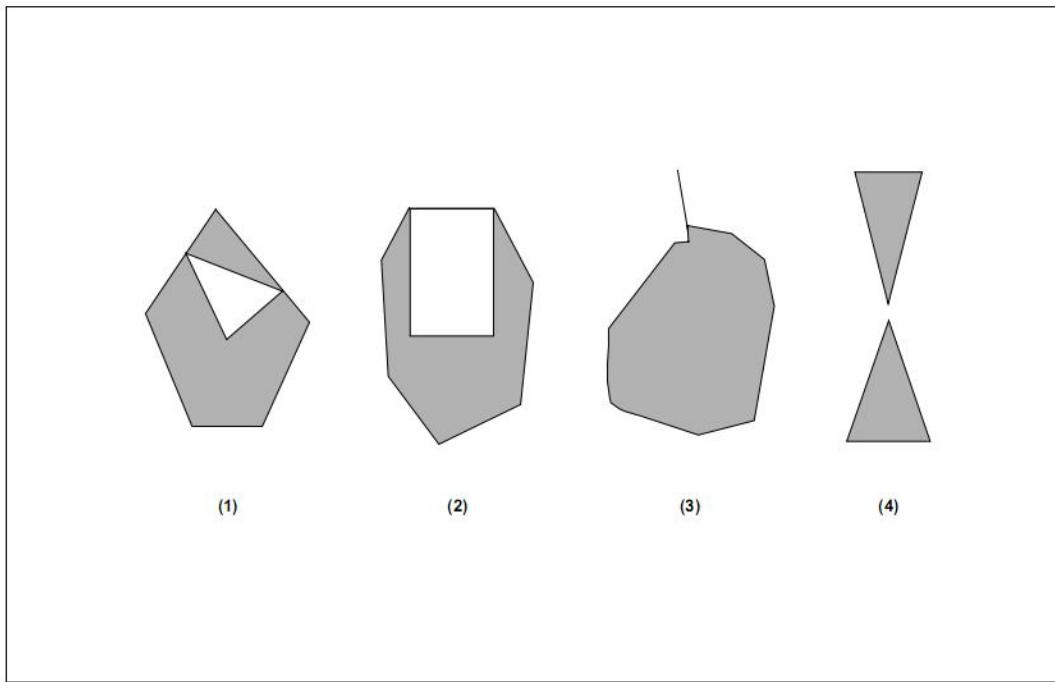
- *Polygon* là một đối tượng hình học dạng đóng.
- Vùng bao của *Polygon* bao gồm một tập hợp các *LinearRings* tạo nên vùng bao “exterior” và “interior” của nó.
- Không thể xảy ra hai đường thuộc vùng bao cắt nhau, chúng chỉ có thể giao nhau tại một *Point* như là một tiếp tuyến: $\forall P \in \text{Polygon}, \forall c1, c2 \in P.\text{Boundary}(), c1 \neq c2, \forall p, q \in \text{Point}, p, q \in c1, p \neq q, [p \in c2 \Rightarrow q \notin c2]$.

- Một Polygon có thể không cắt lines, spikes hoặc punctures: $\forall P \in \text{Polygon}$, $P = \text{Closure}(\text{Interior}(P))$.
- “*Interior*” của một *Polygon* là một tập hợp các Point liên kết nhau.
- “*Exterior*” của một *Polygon* với 1 hoặc nhiều lỗ trống thì không liên kết nhau. Mỗi lỗ trống định nghĩa cho một thành phần kết nối của các “*Exterior*”.

Hình PL.B.4 minh họa một vài ví dụ của *Polygon*. Hình PL.B.5 minh họa một vài ví của các đối tượng hình học vi phạm các khảng định trên và không được xem là *Polygon*. Các đối tượng trong hình PL.B.5-(1) và PL.B.5-(4) như hai *Polygon* riêng biệt.



Hình PL.B.4 - Ví dụ Polygon với 1, 2 và 3 vòng tương ứng.



Hình PL.B.5 - Ví dụ các đối tượng không được xem là *Polygon*. (1) và (4) có thể biểu diễn cho 2 *Polygon* riêng biệt.

Các phương thức

- *ExteriorRing():LineString* - Kết quả trả về vòng “*interior*” của *Polygon* này.
- *NumInteriorRing():Integer* - Kết quả trả về số lượng vòng “*interior*” của *Polygon* này.
- *InteriorRingN(N:Integer):LineString* - Kết quả trả về số lượng vòng “*interior*” thứ N của *Polygon* này như một *LineString*.

k. MultiSurface

MultiSurface là một tập hợp các đối tượng hình học hai chiều, các phần tử của nó là *Surface*. Các “*interior*” của bất kỳ 2 *Surface* trong một *MultiSurface* có thể không giao nhau. Vùng bao của bất kỳ 2 phần tử trong một *MultiSurface* có thể cắt nhau tại một số hữu hạn điểm.

Các phương thức

- *Area():Double* - Diện tích của *MultiSurface* đang xét và được tính trong hệ qui chiều không gian của nó

- *Centroid()*: *Point* - Trả về điểm trọng tâm của MultiSurface đang xét. Kết quả này không chắc chắn là nằm trên MultiSurface.
- *PointOnSurface()*: *Point* - Một Point được đảm bảo là nằm trên MultiSurface đang xét.

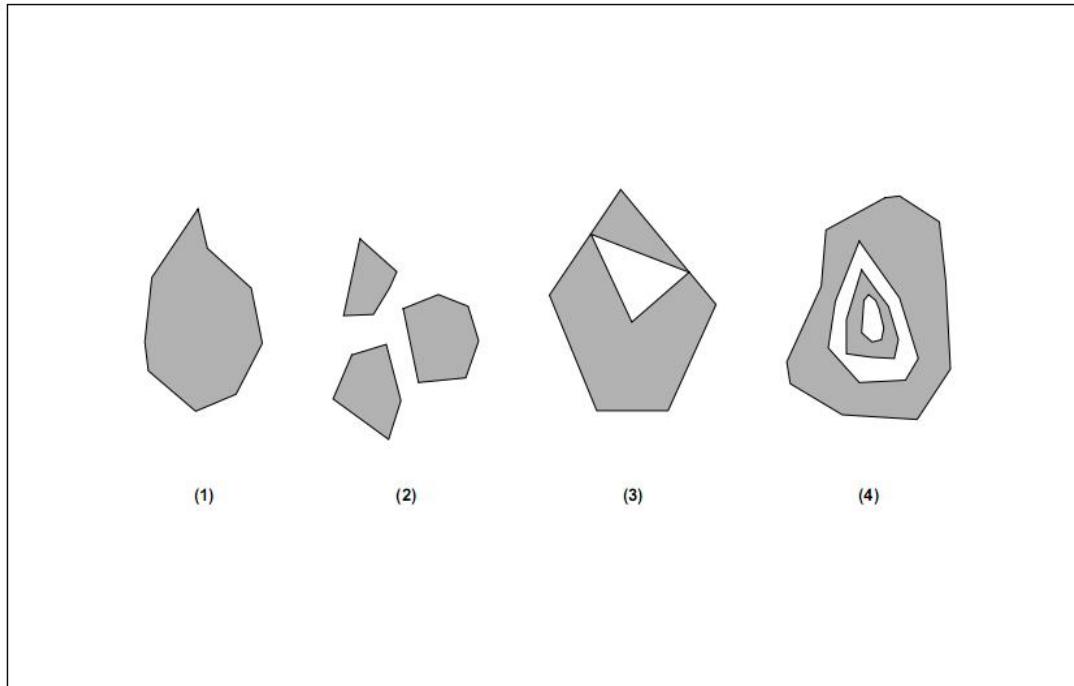
I. MultiPolygon

Một *MultiPolygon* là một *MultiSurface* mà các phần tử của nó là *Polygons*. Các khái niệm cho *MultiPolygons*:

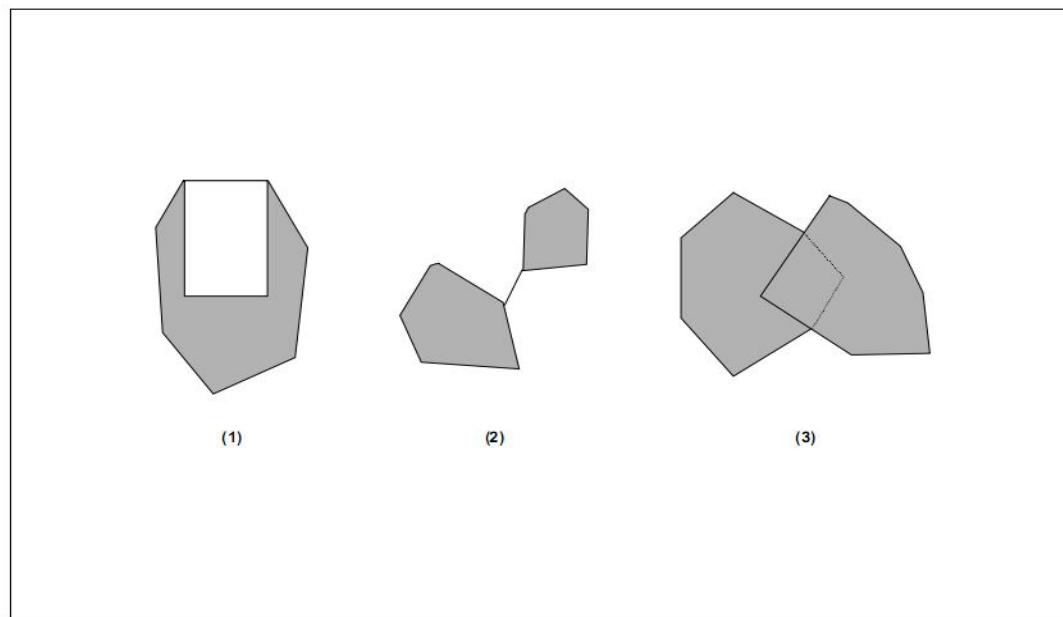
- Các “*Interior*” của 2 Polygons cùng là phần tử của 1 MultiPolygon có thể không giao nhau: $\forall M \in MultiPolygon, \forall Pi, Pj \in M.Geometries(), i \neq j, Interior(Pi) \cap Interior(Pj) = \emptyset$
- Vùng bao của 2 Polygons bất kỳ cùng là phần tử của 1 MultiPolygon có thể không cắt nhau hay tiếp xúc nhau tại một số điểm hữu hạn: $\forall M \in MultiPolygon, \forall Pi, Pj \in M.Geometries(), \forall ci \in Pi.Boundaries(), cj \in Pj.Boundaries() ci \cap cj = \{p1, ..., pk | pi \in Point, 1 \leq i \leq k\}$
- Một *MultiPolygon* được định nghĩa như một đối tượng hình học dạng đóng.
- Một *MultiPolygon* có thể không cắt các đường, spikes hoặc punctures, a *MultiPolygon* is a Regular, Closed pointset: $\forall M \in MultiPolygon, M = Closure(Interior(M))$
- “*Interior*” của một *MultiPolygon* với nhiều hơn 1 Polygon thì không liên kết nhau, số lượng các thành phần liên kết của “*Interior*” trong một MultiPolygon thì bằng với số lượng *Polygon* thuộc *MultiPolygon* đó.

Vùng bao của một MultiPolygon là một tập hợp các curves đóng (*LineStrings*) tương ứng với vùng bao của các phần tử *Polygon* của nó. Mỗi *Curve* trên vùng bao của *MultiPolygon* là vùng bao chính xác của một phần tử *Polygon*, và mỗi *Curve* trong vùng bao của một phần tử *Polygon* là vùng bao của *MultiPolygon*.

Hình PL.B.6 minh họa 4 ví dụ về các MultiPolygon hợp lệ với 1, 3, 2 và 2 polygon tương ứng. Hình PL.B.7 minh họa cho các đối tượng hình học không được xem là *MultiPolygon*



Hình PL.B.6- Ví dụ MultiPolygon



Hình PL.B.7- Ví dụ các đối tượng không là MultiPolygon

PHỤ LỤC C - GIAO DIỆN QUẢN LÝ THÔNG TIN VỚI GEOSERVER

The screenshot shows the 'Layers' management page in the GeoServer web interface. The left sidebar contains links for Server (Status, Contact Information, Global Settings, JAI Settings, About GeoServer), Services (GWC, WCS, WFS, WMS), Data (Workspaces, Stores, Layers, Layer Groups, Styles), Security (Users, Data security, Service security, Catalog security), Map CSS Demo, and Demos. The main content area is titled 'Layers' and displays a table of published layers. The table has columns for Type, Workspace, Store, Layer Name, Enabled?, and Native SRS. The data shown is:

Type	Workspace	Store	Layer Name	Enabled?	Native SRS
ws_cantho	cantho	quanhuyen	<input checked="" type="checkbox"/>	EPSG:4326	
ws_cantho	cantho	truong	<input checked="" type="checkbox"/>	EPSG:4326	
ws_cantho	cantho	uyban	<input checked="" type="checkbox"/>	EPSG:4326	
ws_cantho	cantho	vertices_tmp	<input checked="" type="checkbox"/>	EPSG:4326	
ws_cantho	cantho	xaphuong	<input checked="" type="checkbox"/>	EPSG:4326	

Hình PL.C.1 - Giao diện quản lý các lớp (Layers) trong hệ thống

The screenshot shows the 'Styles' management page in the GeoServer web interface. The left sidebar contains links for Server (Status, Contact Information, Global Settings, JAI Settings, About GeoServer), Services (GWC, WCS, WFS, WMS), Data (Workspaces, Stores, Layers, Layer Groups, Styles), Security (Users, Data security, Service security, Catalog security), Map CSS Demo, and Demos. The main content area is titled 'Styles' and displays a table of published styles. The table has columns for Style Name. The data shown is:

Style Name
population
rain
raster
restricted
simple_roads
simple_streams
sld_pointnow
tiger_roads

Hình PL.C.2 - Giao diện quản lý các styles trong hệ thống.

PHỤ LỤC D - MỘT SỐ THỦ TỤC TRONG POSTGIS HỖ TRỢ XÂY DỰNG DỊCH VỤ

1. Hàm tìm cạnh gần nhất từ một điểm trên bản đồ

```
CREATE OR REPLACE FUNCTION find_nearest_edge(x text , y text)
RETURNS geometry AS
$BODY$
DECLARE
result geometry;
three_d_temp text;
point_temp text;
BEGIN
three_d_temp:='BOX3D(' || (to_number(x,'9999999D9999999999')-10000) ||
' ' || (to_number(y,'9999999D9999999999')-10000) || ', ' ||
(to_number(x,'9999999D9999999999')+10000) || ' ' ||
(to_number(y,'9999999D9999999999')+10000) || ')';
point_temp:='POINT(' || x || ' ' || y || ')';
result:= the_geom FROM giaothong
WHERE the_geom && setsrid(three_d_temp::box3d,4326)
ORDER BY distance(the_geom, GeometryFromText(point_temp, 4326))
LIMIT 1;
return result;
END;
$BODY$
LANGUAGE 'plpgsql' IMMUTABLE STRICT;
```

2. Hàm tách một con đường thành hai con đường với đầu vào là điểm mà người dùng chọn trên bản đồ

```
CREATE OR REPLACE FUNCTION split_multilinestring(gid_a integer,
point_click geometry)
RETURNS text AS
$BODY$
DECLARE
result1 text;
result2 text;
result text;
line text;
```

```

num_of_point integer;
j integer;
k integer;
start_point text;
end_point text;
temp geometry[] := '{}';
point_geom geometry;
result_id integer;
p text;
len1 float;
len2 float;
point_click_text text;
BEGIN
    p:='f';
    result1:="";
    result2:="";
    len1:=0;
    len2:=0;
    num_of_point:= ST_NumPoints(the_geom) FROM giaothong where
    gid=gid_a;
    start_point:= astext(PointN(the_geom,1)) from giaothong where
    gid=gid_a;
    end_point:= astext(PointN(the_geom,num_of_point)) from giaothong
where gid=gid_a;
    IF start_point=astext(point_click) THEN
        result_id:= id FROM dinh WHERE
        astext(the_geom)=start_point;
        return '_'||result_id;

    ELSIF end_point=astext(point_click) THEN
        result_id:= id FROM dinh where astext(the_geom)=end_point;
        return '_'||result_id;
    ELSE
        FOR j IN 1 .. num_of_point LOOP
            point_geom:= PointN(the_geom,j) from giaothong where
            gid=gid_a;
            temp[j]:=point_geom;
            line:="";
            line:= line || X(temp[j])|| '||Y(temp[j])||', '||X(temp[j-1])||'
            '||Y(temp[j-1]);
            line:='LINESTRING'||'(' || line || ')';

```

```

IF j=1 THEN
    result1:= result1 ||X(temp[j])||' ||Y(temp[j])||', ';
END IF;
SELECT INTO p
ST_intersects(ST_GeomFromText(line,4326),ST_GeomFromText(AsText(ST_Buffer(
$2,0.001)),4326));
IF p!= 't' THEN
    result1:= result1 ||X(temp[j])||' ||Y(temp[j])||', ';

END IF;
IF p = 't' THEN
    result1:= result1 ||X(point_click)||' ||Y(point_click);
    result1:='MULTILINESTRING'||(' || result1 || ')';
    result2:= result2 ||X(point_click)||' ||Y(point_click)||', ';
    FOR k IN j .. num_of_point LOOP
        point_geom:=PointN(the_geom,k) from giaothong where
        gid=gid_a;
        if astext(point_geom)=end_point then
            result2:= result2 ||X(point_geom)||'
            '||Y(point_geom);
            result2:='MULTILINESTRING'||(' || result2 || ')';
        else
            result2:= result2 ||X(point_geom)||'
            '||Y(point_geom)||', ';
        end if;
    END LOOP;
    len1:=ST_Length(ST_GeomFromText(result1,4326));
    len2:=ST_Length(ST_GeomFromText(result2,4326));
    result:= result1||"$||len1||"$||result2||"$||len2;
    return result1||"$||len1||"$||result2||"$||len2;
END IF;
END LOOP;
END IF ;
END;
$BODY$
LANGUAGE 'plpgsql' IMMUTABLE STRICT;

```

3. Hàm tìm kiếm các địa điểm xung quanh một vị trí theo tên và bán kính

```

CREATE OR REPLACE FUNCTION find_place_around_point(x text, y text, t
text, radius float )
RETURNS SETOF coquan AS

$BODY$
DECLARE
    r coquan%rowtype;
    point_text text;
    point_geometry geometry;
BEGIN
    point_text:='POINT(' || x || ' ' || y || ')';
    point_geometry := ST_GeomFromText(point_text,4326);
    FOR r IN SELECT gid,'coquan' as ma,ten,diachi,sdt,the_geom
    FROM coquan where lower(ten )like '%'||lower(t)||'%'
    and
    ST_Distance(point_geometry,the_geom)<=radius LOOP RETURN NEXT r; END
LOOP;--1
    FOR r IN SELECT gid,'truong' as ma,ten,diachi,sdt,the_geom
    FROM truong where lower(ten )like '%'||lower(t)||'%'
    and
    ST_Distance(point_geometry,the_geom)<=radius LOOP RETURN NEXT r; END
LOOP;--2
    FOR r IN SELECT gid,'benhvien' as ma,ten,diachi,sdt,the_geom
    FROM benhvien where lower(ten )like '%'||lower(t)||'%'
    and
    ST_Distance(point_geometry,the_geom)<=radius LOOP RETURN NEXT r; END
LOOP;--3
    FOR r IN SELECT gid,'cho' as ma,ten,diachi,sdt,the_geom FROM
    cho where lower(ten )like '%'||lower(t)||'%'
    and
    ST_Distance(point_geometry,the_geom)<=radius LOOP RETURN NEXT r; END
LOOP;--4
    FOR r IN SELECT gid,'ben' as ma,ten,diachi,null as sdt,the_geom
    FROM ben where lower(ten )like '%'||lower(t)||'%'
    and
    ST_Distance(point_geometry,the_geom)<=radius LOOP RETURN NEXT r; END
LOOP;--5
    FOR r IN SELECT gid,'khachsan' as ma,ten,diachi,sdt,the_geom
    FROM khachsan where lower(ten )like '%'||lower(t)||'%'
    and
    ST_Distance(point_geometry,the_geom)<=radius LOOP RETURN NEXT r; END
LOOP;--6
    FOR r IN SELECT gid,'congty' as ma,ten,diachi,sdt,the_geom
    FROM congty where lower(ten )like '%'||lower(t)||'%'
    and
    ST_Distance(point_geometry,the_geom)<=radius LOOP RETURN NEXT r; END
LOOP;--7

```

```

FOR r IN SELECT gid,'giaitri' as ma,ten,diachi,sdt,the_geom      FROM
giaitri      where lower(ten )like '%'||lower(t)||'%' and
ST_Distance(point_geometry,the_geom) <=radius LOOP RETURN NEXT r; END
LOOP;--8

FOR r IN SELECT gid,'denchua' as ma,ten,diachi,sdt,the_geom
FROM denchua      where lower(ten )like '%'||lower(t)||'%' and
ST_Distance(point_geometry,the_geom) <=radius LOOP RETURN NEXT r; END
LOOP;--9

FOR r IN SELECT gid,'buudien' as ma,ten,diachi,sdt,the_geom
FROM buudien      where lower(ten )like '%'||lower(t)||'%' and
ST_Distance(point_geometry,the_geom) <=radius LOOP RETURN NEXT r; END
LOOP;--10

FOR r IN SELECT gid,'nganhang' as ma,ten,diachi,sdt,the_geom
FROM nganhang      where lower(ten )like '%'||lower(t)||'%' and
ST_Distance(point_geometry,the_geom) <=radius LOOP RETURN NEXT r; END
LOOP;--11

FOR r IN SELECT gid,'congvien' as ma,ten,diachi,null as sdt,the_geom
FROM congvien      where lower(ten )like '%'||lower(t)||'%' and
ST_Distance(point_geometry,the_geom) <=radius LOOP RETURN NEXT r; END
LOOP;--12

FOR r IN SELECT gid,'cau'    as ma,ten,diachi,null as sdt,the_geom
FROM cau      where lower(ten )like '%'||lower(t)||'%' and
ST_Distance(point_geometry,the_geom) <=radius LOOP RETURN NEXT r; END
LOOP;--15

FOR r IN SELECT gid,'thuvien' as ma,ten,diachi,sdt,the_geom
FROM thuvien      where lower(ten )like '%'||lower(t)||'%' and
ST_Distance(point_geometry,the_geom) <=radius LOOP RETURN NEXT r; END
LOOP;--14

RETURN;
END
$BODY$
LANGUAGE 'plpgsql' IMMUTABLE STRICT;

```

Ngoài ra còn một số hàm quan trọng khác như:

- Tách một con đường thành ba con đường
- Tìm kiếm địa điểm theo chuỗi có dấu và không dấu
- Tìm kiếm địa điểm theo lộ trình
- Tìm kiếm con đường
- Hàm tạo hình trạng mạng cho hệ thống đường đi