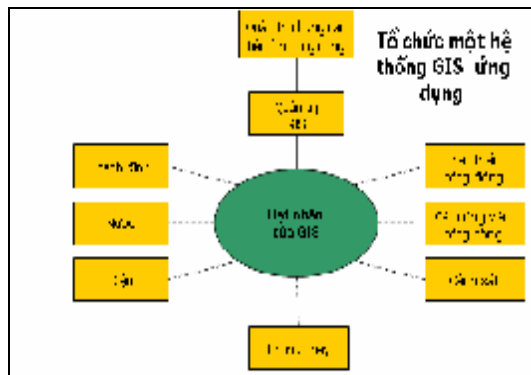


XÂY DỰNG BỘ CÔNG CỤ BIÊN TẬP DỮ LIỆU TOPOLOGY HỖ TRỢ CHO HỆ THỐNG HCMGIS

Vũ Thanh Nguyên, Nguyễn Minh Nam, Lê Phấn Ninh
Sở Bưu chính viễn thông Tp.HCM

1.TỔNG QUAN VỀ HỆ THỐNG HCM GIS

Dự án Hệ thống thông tin địa lý thành phố Hồ Chí Minh (HCM-GIS) do Ủy ban nhân dân thành phố Hồ Chí Minh quản lý, Sở Khoa học và Công nghệ chủ trì. Đây là dự án tổng thể ứng dụng công nghệ GIS cho quản lý, đặc biệt là quản lý đô thị. Các phân hệ được xây dựng trong dự án này mang tính chuyên ngành, ở qui mô toàn thành phố (các lớp thông tin điện, nước, giao thông, công trình ngầm, môi trường, dân cư...). Mỗi phân hệ là một dự án thành phần, là cơ sở cung cấp dữ liệu nền chuẩn cho thành phố.



Hình 1. Tổ chức hệ thống HCM GIS



Hình 2. Sơ đồ các phân hệ của hệ thống HCM GIS

HCM GIS là một hệ thống thông tin địa lý có nhiều thành viên, mỗi thành viên là một phân hệ hoạt động tương tác nhau theo những qui định trong công việc khung của hệ thống, gọi là công việc khung HCM GIS. Số lượng thành viên tham gia không bị ràng buộc bởi thiết kế ban đầu vì HCM GIS có thể kết nạp thêm thành viên bất kỳ tại thời điểm nào. Mỗi phân hệ là một đơn vị quản lý của thành phố, hoạt động chuyên sâu về một lĩnh vực, hoạt động hoàn toàn độc lập về tổ chức mạng, phần mềm, tổ chức dữ liệu chuyên ngành của mình. Hệ thống HCM GIS phát triển theo 2 giai đoạn: giai đoạn một xây dựng cơ sở hạ tầng của hệ thống; giai đoạn hai phát triển các dịch vụ ứng dụng công nghệ thông tin địa lý.

Giống như các mô hình hệ thống thông tin khác, dữ liệu của hệ thống HCM GIS là thành phần quan trọng nhất của hệ thống và nó chiếm khoảng 70% tỷ trọng của hệ thống. Giá trị của hệ thống phụ thuộc vào các yếu tố của dữ liệu như loại dữ liệu, độ chính xác, mức độ chi tiết...Tuy nhiên phần lớn dữ liệu của hệ thống HCM GIS, bao gồm cả các dữ liệu nền, dữ liệu dùng chung và dùng riêng tại các phân hệ thuộc hệ thống HCM GIS đều chưa được topo hoá. Do đó sẽ khó khăn rất nhiều khi triển khai các phân hệ khác nhau của hệ thống (như phân hệ giao thông, phân hệ cấp thoát nước, phân hệ bưu điện, phân hệ điện lực...) và ngay tại Trung tâm tích hợp HCM GIS (cơ sở dữ liệu địa chất, cơ sở dữ liệu bản đồ nền...).

2.MÔ HÌNH DỮ LIỆU KHÔNG GIAN CỦA HỆ THỐNG GIS.

2.1. Dữ liệu địa lý ([2],[7],[11]):

Do sự tích hợp nhiều ngành khác nhau nên hệ thống thông tin địa lý làm việc trên nhiều nguồn thông tin khác nhau: bản đồ, ảnh hàng không, ảnh viễn thám. Dữ liệu rất đa dạng, do đó nếu dữ liệu có mang tính không gian và tình thời gian thì gọi là dữ liệu địa lý. Dữ liệu địa lý bao gồm:

§ Nhóm thông tin về phân bố không gian: tọa độ, hình học...

§ Nhóm thông tin về thuộc tính của đối tượng: tên thành phố, số dân...

§ Để biểu diễn dữ liệu với khả năng chia sẻ thông tin địa lý cao nhất giữa các ứng dụng, người dùng cần có khái niệm về giản đồ (schema):

Mô hình Hình học (Geometry - [3],[12]): mô tả đối tượng thông qua các chức năng toán học bao gồm: kích thước, vị trí, hình dạng, định hướng... Mô hình Hình học dựa trên một hệ thống tham chiếu để đưa ra vị trí không gian.

Mô hình Topology ([4],[10]): được sử dụng để mô tả sự kết nối các đồ thị n-kích thước, có sự trợ giúp của máy tính điện tử.

Mô hình chồng xếp ([5],[6]): Một trong những phương pháp chung nhất cho tổ chức dữ liệu địa lý là tổ chức theo bản đồ và theo các lớp thông tin. Mỗi lớp thông tin là một biểu diễn dữ liệu theo một mục tiêu nhất định. Mô hình dữ liệu địa lý bao gồm bốn thành phần: khoá, định vị, phi không gian và không gian. Thành phần không gian được quản lý theo raster hoặc vector.

Mô hình raster: chia không gian thành các ô lưới đều nhau, thường được gọi là các điểm ảnh (pixel). Độ phân giải của raster phụ thuộc vào kích thước điểm ảnh của nó.

Mô hình vector: thể hiện các đối tượng không gian bằng các cấu trúc dữ liệu mà điểm là đối tượng cơ sở. Các cấu trúc dữ liệu được sử dụng để lưu trữ các đối tượng dữ liệu không gian trong mô hình vector gồm điểm, đường, đa giác.

2.2. Khai thác dữ liệu không gian ([4],[10]):

Khai thác dữ liệu không gian là một loại khai thác dữ liệu đặc biệt. Sự khác biệt chủ yếu giữa khai thác dữ liệu và khai thác dữ liệu không gian là các tác vụ khai thác dữ liệu không gian không chỉ sử dụng các thuộc tính phi không gian mà còn sử dụng các thuộc tính không gian. Trong khai thác dữ liệu không gian, các quan hệ vùng không gian lân cận là rất quan trọng. Có 4 loại quan hệ không gian cơ bản: *Topology*, *khoảng cách*, *hướng*, *lân cận phức*.

Các đối tượng không gian có thể được thể hiện bởi tập điểm chứa trong đối tượng. Một cách tổng quát, các điểm $p = (p_1, p_2, \dots, p_d)$ là các phần tử của không gian véctor Euclid với d chiều. Tuy nhiên, trong bài báo này chỉ trình bày 2 chiều, mặc dù tất cả các ký hiệu có thể dễ dàng áp dụng cho số chiều nhiều hơn. Các đối tượng không gian O được thể hiện bởi một tập các điểm, có nghĩa là $O \in 2^N$ (N : các điểm trong không gian, 2^N : số tập hợp con). Cho một điểm $p = (p_x, p_y)$, p_x và p_y ký hiệu cho các tọa độ của p theo trục x và trục y tương ứng.

§ **Quan hệ Topology:** là các quan hệ bất biến dưới các phép biến đổi topology, ví dụ các quan hệ này được giữ ngay khi cả hai đối tượng bị quay, dịch chuyển, hoặc co giãn.

Định nghĩa 1: Các quan hệ topology giữa 2 đối tượng A và B được dẫn xuất từ 9 trường hợp giao của các phần trong, các biên và phần bù của A và B với nhau. Các quan

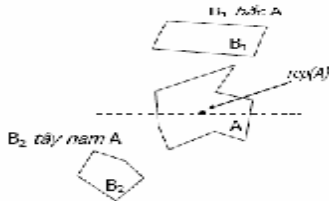
hệ này gồm có: A không giao với B , A tiếp xúc với B , A chồng lấp B , A bằng B , A phủ B , A bị phủ bởi B , A chứa B , và A nằm bên trong B .

§ Quan hệ khoảng cách: là các quan hệ so sánh khoảng cách của hai đối tượng với hằng số cho trước sử dụng một trong các toán tử số học.

Định nghĩa 2: Cho $Dist : O \times O \rightarrow \mathbb{R}^+$ là hàm khoảng cách, cho s là một trong các toán tử số học $\{p, f, =\}$ ($s \in \{p, f, =\}$), cho $d \in \mathbb{R}$ và A và B là hai đối tượng không gian. Thì A và B có quan hệ khoảng cách nếu và chỉ nếu $Dist(A, B) s d$.

§ Quan hệ hướng: Quan hệ hướng của hai đối tượng không gian được định nghĩa bằng một điểm đại diện $rep(A)$ của đối tượng nguồn A và tất cả các điểm của đối tượng đích B . Điểm đại diện của đối tượng nguồn được sử dụng như là gốc của hệ trục tọa độ ảo và các phần tử của nó định nghĩa các hướng. Các quan hệ hướng 2 chiều có thể được thể hiện bằng các tên địa lý.

Định nghĩa 3: Gọi $rep(A)$ là thể hiện của đối tượng nguồn A .



Hình 4. Quan hệ về hướng giữa các đối tượng.

Quan hệ **B đông_bắc A** tồn tại nếu và chỉ nếu $\forall b \in B : b_x \geq rep(A)_x \wedge b_y \geq rep(A)_y$.

Các hướng **đông_nam**, **tây_nam** và **tây_bắc** được định nghĩa tương tự.

Quan hệ **B bắc A** tồn tại nếu và chỉ nếu $\forall b \in B : b_y \geq rep(A)_y$. Các hướng **nam**, **tây**, **bắc** được định nghĩa tương tự.

Quan hệ **B hướng_bất_kì A** đúng cho mọi A, B .

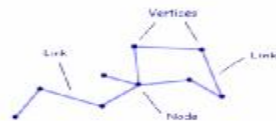
Mỗi cặp đối tượng không gian có tối thiểu một trong các quan hệ hướng, nhưng quan hệ hướng này có thể không phải là duy nhất. Chỉ những quan hệ đặc biệt **tây_bắc**, **đông_bắc**, **tây_nam**, **đông_nam** là loại trừ lẫn nhau.

§ Quan hệ lân cận phức: Các quan hệ Topology, khoảng cách và hướng có thể được kết hợp dùng các toán tử logic $\wedge$ (và) cũng như $\vee$ (hoặc) để biểu diễn các quan hệ lân cận phức.

Hệ Quả: Nếu r_1 và r_2 là các quan hệ lân cận, thì $r_1 \wedge r_2$ và $r_1 \vee r_2$ cũng là các quan hệ lân cận (phức).

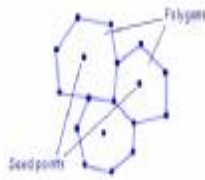
2.3. Mô hình Topology ([4],[10]):

Nếu như bản đồ địa hình mô tả những vị trí vật lý và hình dạng của các đối tượng tự nhiên thì mô hình topology thể hiện các mối quan hệ giữa các đối tượng đó. Mô hình Topology là một mô hình được quản lý dựa trên sự liên kết giữa các đối tượng. Khi một đối tượng thay đổi sẽ ảnh hưởng đến đối tượng khác và thay đổi các liên kết trong mô hình Topology. Sau đây là các khái niệm cơ bản được sử dụng trong mô hình Topology:

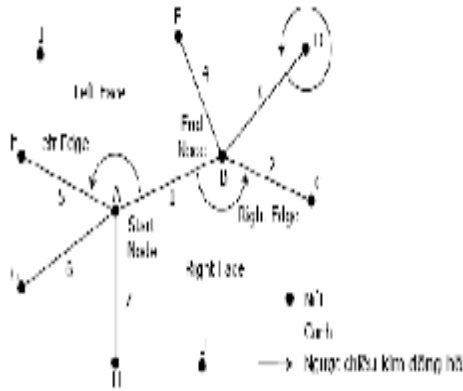


Hình 5. Minh họa về nút và cạnh

Nút và cạnh liên kết: Trong hệ thống thông tin địa lý các đối tượng đường thẳng (Lines) còn được gọi là cạnh liên kết (Links) được tạo từ những điểm (Vertex) và những nút (node) tại hai đầu mút của cạnh.



Hình 6. Minh họa về vùng



Hình 7. Minh họa một số khái niệm cơ bản trong mô hình Topology

Vùng (polygon): là một vùng không gian đóng, được chỉ ra bởi giới hạn của tập hợp các cạnh và điểm mà có mối quan hệ đến tính chất địa lý cấu tạo nên nó.

Start Node: Nút bắt đầu của cạnh.

End Node: Nút kết thúc của cạnh.

Right Edge: Cạnh đầu tiên gặp khi di chuyển ngược chiều kim đồng hồ tại EndNode của cạnh hiện tại.

Left Edge: Cạnh đầu tiên gặp khi di chuyển ngược chiều kim đồng hồ tại StartNode của cạnh hiện tại.

First Edge: Cạnh được chọn ngẫu nhiên, được xem như là cạnh đầu tiên cho việc tìm kiếm các cạnh kề của nút.

Left Face: Mặt ở bên trái của cạnh khi di chuyển từ nút bắt đầu đến nút kết thúc.

Right Face: Mặt ở bên phải của cạnh khi di chuyển từ nút bắt đầu đến nút kết thúc.

Minimum Bounding Rectangle (MBR): Khung chữ nhật nhỏ nhất chứa toàn bộ đối tượng.

Inner Ring: Biên trong của mặt. Mỗi đối tượng vùng có thể không có, hay có một hoặc nhiều biên trong.

Outer Ring: Biên bao ngoài của mặt. Mỗi đối tượng vùng có duy nhất một biên ngoài.

Đặc trưng: mô hình của đối tượng địa lý thế giới thực. Các đối tượng này có thể là đối tượng vô hướng, một chiều, hai chiều và ba chiều.

Nút: Các nút là các đối tượng cơ sở vô hướng được sử dụng để lưu trữ những vị trí có ý nghĩa. Hai nút bất kì không thể có cùng tọa độ. Nút có hai loại: nút thực thể và nút kết nối.



Hình 8. Các đối tượng hình học cơ sở trong mô hình Topology minh họa cạnh, vùng, nút kết nối, nút thực thể, văn bản

Nút thực thể: Nút thực thể dùng để thể hiện các đặc trưng riêng biệt hoặc vô hướng. Nút thực thể được liên kết về mặt topo với mặt chứa nó khi topo mặt hiện diện. Các nút thực thể không được nằm trên cạnh.

Nút kết nối: Các nút kết nối xuất hiện ở các đầu mút của cạnh và được liên kết về topo với các cạnh khác. Các nút kết nối được sử dụng theo hai cách: (1) định nghĩa các cạnh về topo và (2) thể hiện các đặc trưng của điểm được tìm thấy tại đầu và cuối của cạnh của các đặc trưng tuyến tính. Với cách dùng (1), các nút kết nối được xem như là các điểm đầu và cuối. Với cách dùng (2), các thuộc tính sẽ được kết hợp với các đặc trưng của điểm được liên hệ với các nút kết nối. Nếu nhiều cạnh giao nhau tại một nút, thì chỉ một cạnh sẽ được duy trì cho mỗi nút trong bảng node kết nối.

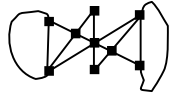
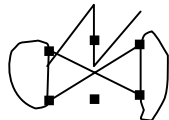
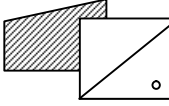
Cạnh: là đối tượng cơ sở một chiều được sử dụng để thể hiện các vị trí của các đặc trưng tuyến tính và các biên của các mặt. Các cạnh được cấu thành từ một tập có thứ tự của hai hay nhiều cặp tọa độ (x,y) hoặc (x,y,z) . Tối thiểu hai trong số các cặp tọa độ phải được phân biệt. Các cạnh về topo được định nghĩa bởi các nút tại các điểm đầu và điểm cuối; các cạnh (topology cấp 1-3) đến lượt nó định nghĩa các mặt (topology cấp 3). Các thông tin topo cho phép truy vấn và lấy ra các đặc trưng.

Mặt: là đối tượng hình học hai chiều được định nghĩa bởi các cạnh; đối tượng mặt thường được dùng để thể hiện các đặc trưng vùng. Các đối tượng mặt được định nghĩa bởi các tham chiếu về topo tới tập các cạnh hình thành nên các biên của mặt. Đối tượng mặt có thể có các biên trong cũng như các biên ngoài, và có thể chứa các mặt nhỏ hơn trong nó. Quan hệ này gồm một tham chiếu đến điểm khởi đầu của một biên khép kín của các cạnh, rồi theo chiều kim đồng hồ để khép kín biên. Mặt có thể có nhiều biên (rings); có thể có một biên ngoài và không có hay có một hoặc nhiều biên trong. Các mặt không được chồng lấp nhau, và các mặt trong một lớp sử dụng toàn bộ vùng mặt phẳng.

Topology: Có 4 cấp độ topology: 0,1,2,3. Ở cấp 3, các kết nối về topo hiện diện một cách tường minh. Ở cấp 0, không có thông tin topo được thể hiện một cách tường minh.

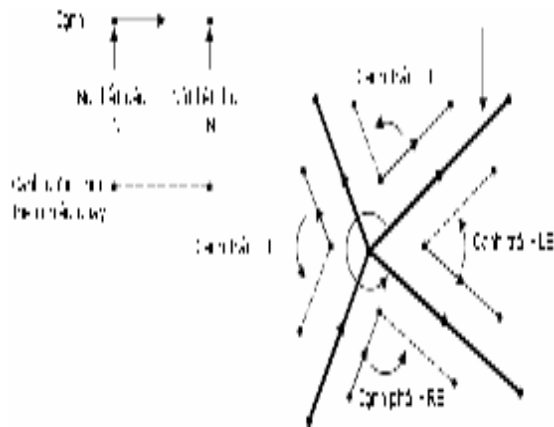
Bảng 1. Tổng kết đặc tính của 4 cấp và ví dụ về mỗi cấp

Cấp	Tên	Các đối tượng cơ sở	Mô tả	Ví dụ
3	Quan hệ Topology đầy đủ	Nút kết nối, nút thực thể, cạnh và mặt	Bề mặt được phân chia bởi tập các mặt được chọn kỹ và phần chung duy nhất. Các cạnh chỉ gặp nhau tại các nút.	

2	Đồ thị phẳng	Nút thực thể, nút kết nối và cạnh	Một tập các cạnh và các nút ở đó khi chiếu vào bề mặt phẳng, các cạnh chỉ gặp nhau tại các nút.	
1	Đồ thị không phẳng	Nút thực thể, nút kết nối và cạnh	Tập các nút thực thể và các cạnh có thể gặp nhau tại các nút.	
0	Thể hiện hình bao	Nút thực thể và cạnh	Tập các nút thực thể và các cạnh. Các cạnh chỉ chứa các tọa độ, không phải là nút bắt đầu và nút kết thúc.	

Một số thuật toán cơ bản trên mô hình Topology ([1],[3]).

§ Thuật toán: Tìm cạnh kề của nút ([1]).



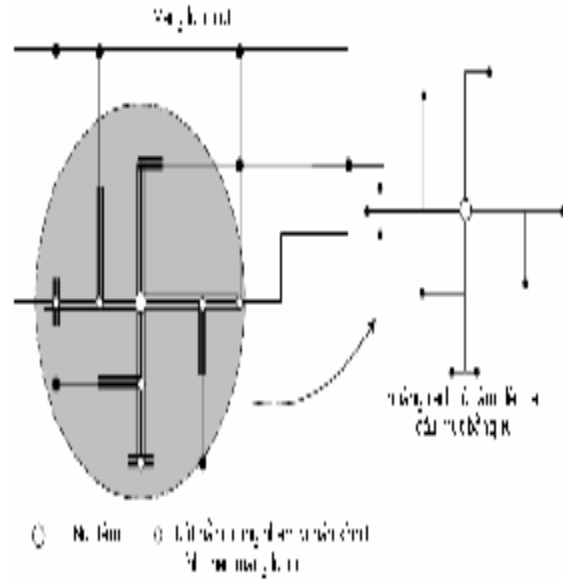
Thuật toán này cho phép tìm kiếm các cạnh nối với một nút được cho trước. Mỗi nút có thuộc tính First Edge chứa thông tin về cạnh đầu nối với nút đó. Hàm `getNeighbourEdges` trả về tập các cạnh nối với nút có Id là `NodeId`: `void getNeighbourEdges(long GID, list<long> lstEdges)`, ở đó `GID` là nút, `lstEdges` là danh sách các cạnh kề.

Hình 9. Các cạnh lân cận của nút

§ Thuật toán: Tìm các nút lân cận của nút ([1]).

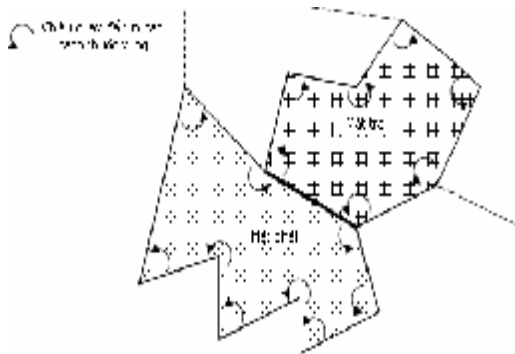
Thuật toán tìm các nút lân cận một nút cũng thường gặp trong các bài toán trên mô hình topo. Hàm ***getNeighbourNodes*** sau trả về tập nút lân cận với nút được cho: `void getNeighbourNodes(long GID, list<long> lstNodes)`, ở đó `GID` là nút, `lstNodes` là Danh sách các nút kề.

Hai thuật toán tìm nút lân cận trên chỉ cho phép tìm kiếm các nút kề trực tiếp với nút được cho (hai nút nối với nhau thông qua một cạnh). Thuật toán sau đây cho phép tìm kiếm các nút nằm trong bán kính R với tâm là nút được cho. Như vậy nút được chọn trả về nếu khoảng cách từ tâm đến nó $\leq R$ (được tính theo đường đi trên mạng, không phải là đường chim bay). Hàm **double Distance(long GID_1 , long GID_2)** trả về khoảng cách giữa hai nút kề nhau GID_1 , GID_2 .



Hình 10. Các nút trong bán kính R

§ Thuật toán: Tìm vùng



Hình 11. Các vùng của cạnh

Theo mô hình Topology này, đối tượng vùng tồn tại khi mô hình ở dạng Topology cấp 3. Khi đó mỗi cạnh có hai vùng được gọi là vùng bên trái (LF - Left Face) và vùng bên phải (RF - Right Face). Các thuật toán này được áp dụng để tìm các vùng không chứa hố (hole) bên trong và để dùng khi chưa có dữ liệu về vùng.

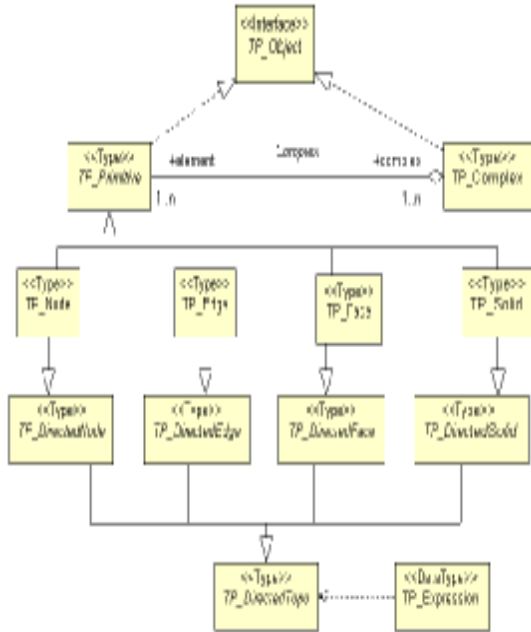
§ Thuật toán: Xác định vùng bên trái của cạnh ([1]): Hàm **getLeftFace** trả về danh sách ID của các cạnh tạo thành vùng bên trái của cạnh: **void getLeftFace(long GID , list<long> $lstEdges$)**, ở đó GID là cạnh và $lstEdges$ là danh sách các cạnh tạo thành vùng.

§ Thuật toán: Xác định vùng bên phải của cạnh ([1]): Hàm **getRightFace** trả về danh sách các ID của các cạnh tạo thành vùng bên phải của cạnh: **void getRightFace(long GID , list<long> $lstEdges$)**, ở đó GID là cạnh, $lstEdges$ là danh sách các cạnh tạo thành vùng

3. PHẦN MỀM XÂY DỰNG BỘ BIÊN TẬP DỮ LIỆU TOPOLOGY

Chương trình minh họa được sử dụng ngôn ngữ Visual C++ 6.0 cho cài đặt và được kết nối đến MySQL để quản lý cơ sở dữ liệu. Cơ sở dữ liệu gồm các bảng Node_Table, Edge_Table, Face_Table và Ring_Table. Mỗi bảng chứa thông tin các đối tượng cần thao tác như node, edge, và face...Chương trình minh họa thực hiện các thao tác biên tập cơ bản nhất trên các đối tượng topology. Các bảng cơ sở dữ liệu được tổ chức lưu trữ trong MySQL dạng chuẩn nhất định. Trong phạm vi của nhóm nghiên cứu, mô hình Vector

Format Product (VPF) được sử dụng như một công cụ có hiệu quả trong thao tác dữ liệu. Topology Editor chủ yếu thực hiện việc thao tác trên nguồn dữ liệu theo dạng chuẩn VPF và chuyển các khuôn dạng từ non_topo sang dạng chuẩn này. Dưới đây trình bày tổng thể mô hình quản lý các đối tượng dữ liệu và ý nghĩa của chúng:

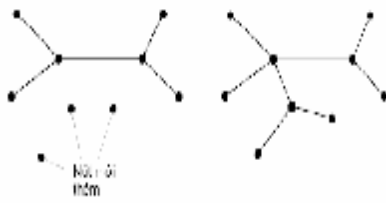


Việc thao tác với những đối tượng dữ liệu không gian sẽ gồm ba quá trình sau:

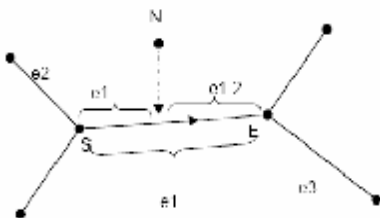
- § Quá trình tạo ra các đối tượng mới hay loại bỏ đối tượng cũ.
- § Quá trình tạo quan hệ giữa những đối tượng khác nhau. Có thể thấy rằng việc quản lý các đối tượng bản đồ theo mô hình VPF hiệu quả cho quá trình phân tích mạng tuyến, nhằm đưa các phép toán hình học phức tạp trên các đối tượng hình học về các phép toán đại số trên đối tượng Topology.

Hình 12. Mô tả tổng thể quản lý đối tượng dữ liệu § Quá trình tạo thuộc tính cho đối tượng.

3.1.Một số chức năng cơ bản của phần mềm



Hình 13. Minh họa nút mới thêm vào

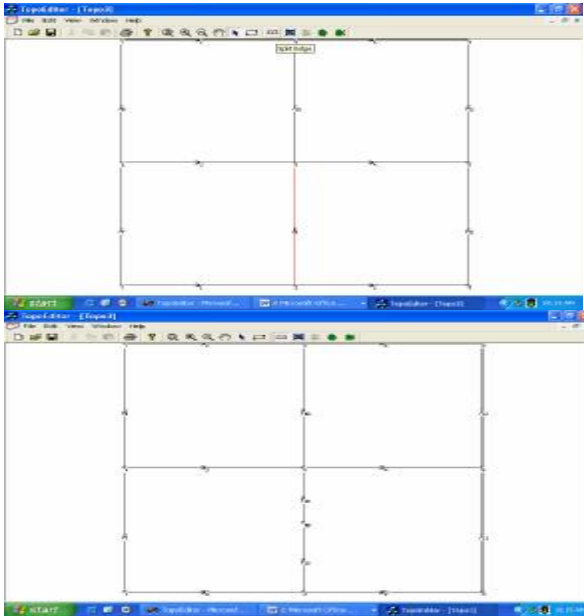


Hình 14. Minh họa bài toán thêm một nút

Chức năng thêm một nút mới: Trong mô hình Topology, một nút thực được gọi là nút có hướng. Nút thực có nhiệm vụ liên kết các cạnh khác nhau và hỗ trợ cho việc phân tích mạng tuyến. Việc thêm một nút mới rồi lần lượt thêm các cạnh do nút này quản lý sẽ dễ dàng và đơn giản hơn thêm một cạnh trực tiếp .

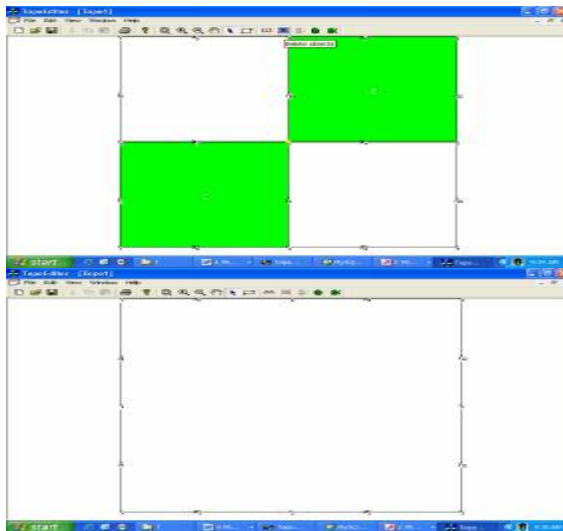
Theo chuẩn OpenGis, một nút Topo thường quan hệ đến các đối tượng cạnh Topo. Khi một nút mới thêm vào có thể không có mối quan hệ với bất kỳ đối tượng nào trong bản đồ nếu nó là nút cô lập, có bậc bằng 0. Ngược lại, mọi nút có hướng phải quan hệ với ít nhất một cạnh mà nó quản lý.

§ Chức năng Split Edge: Khi một nút được thêm vào nằm trên cạnh cho trước thì quá trình thêm nút thực chất là quá trình chia cạnh với điểm giữa là nút mới thêm vào.



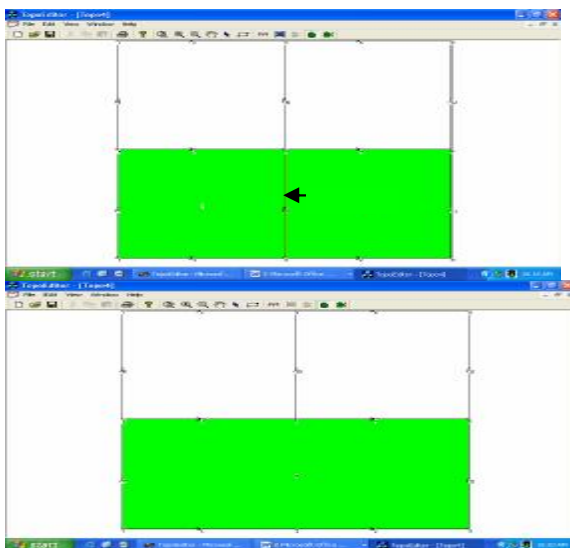
Hình 15. Minh họa bài toán Split Face

§ Chức năng xóa một nút: các bước trong quá trình xóa một nút: Tìm tất cả các cạnh nối đến nút đó; Xóa các cạnh này, cập nhật lại leftface và rightface của mỗi cạnh; Xóa nút.



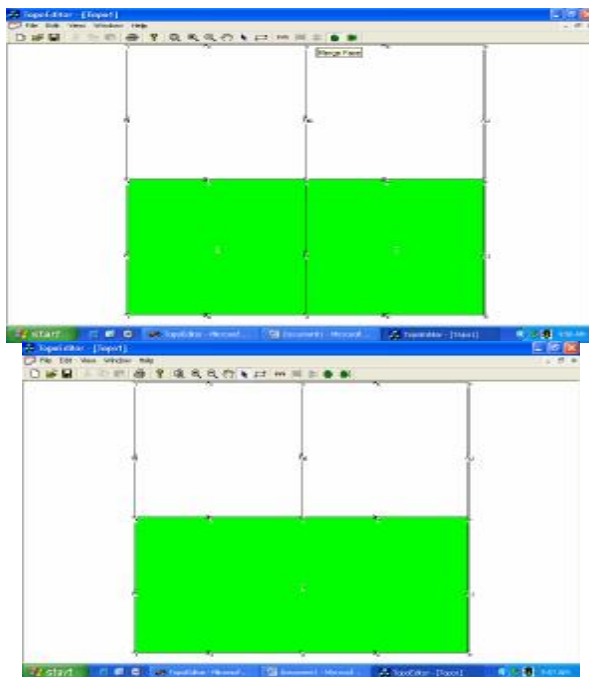
Hình 16. Minh họa bài toán xóa nút

§ Chức năng xóa một cạnh: Các bước tiến hành: Tìm StartNode, EndNode, LeftEdge, RightEdge, LeftFace, RightFace; Cập nhật các thông tin LeftEdge, RightEdge, LeftFace, RightFace của cạnh bị xóa; Cập nhật lại StartNode và EndNode; xóa cạnh.



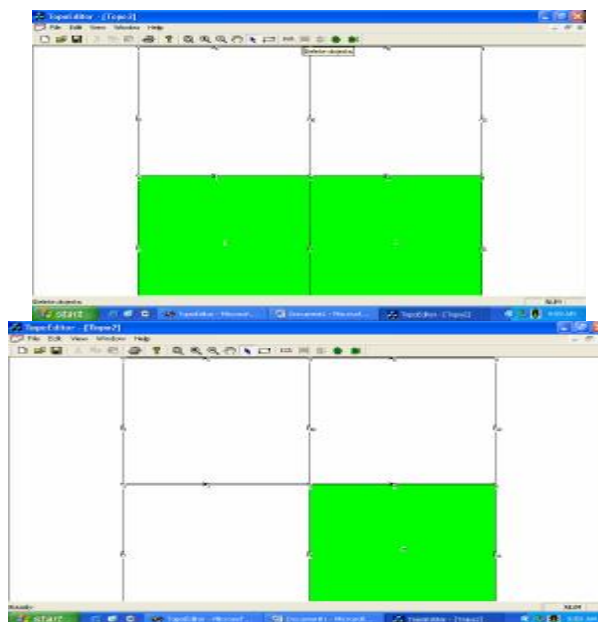
Hình 17. Minh họa bài toán xóa một cạnh.

§ Chức năng Merge mặt: Quá trình Merge mặt thực chất là xóa cạnh chung giữa các mặt. Việc xóa đối tượng cạnh làm cho quan hệ Topo trong bản đồ thay đổi nên cần cập nhật lại thông tin cho các đối tượng cạnh, mặt, và nút để đảm bảo tính liên kết chặt chẽ trong mô hình topo.



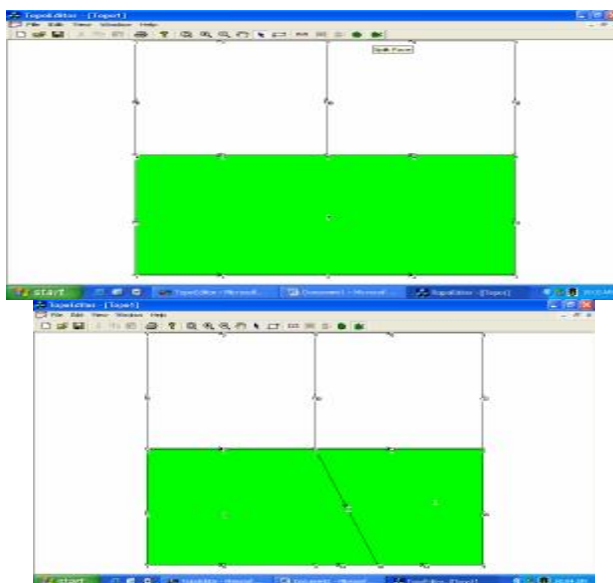
Hình 18. Minh họa bài toán Merge mặt

§ Chức năng Xóa mặt: Qui trình xóa mặt thực chất là một quá trình cập nhật lại các thông số thích hợp trong các bảng cơ sở dữ liệu. Khi một mặt bị xóa, các đối tượng bị ảnh hưởng gồm: LeftFace, RightFace của một cạnh và FirstEdge, FaceID, RingID của mặt.



Hình 19. Minh họa bài toán xóa mặt

§ Chức năng Split mặt: Quá trình Split mặt liên quan đến việc chọn điểm đầu và cuối cho cạnh thêm vào. Điểm đầu và điểm cuối phải nằm trên hai cạnh khác nhau của một mặt. Ngoài ra, quá trình Split mặt còn được hỗ trợ bởi cấu trúc nút ảo để có thể chọn bất kỳ nút nào trong vùng của một mặt. Điều này hỗ trợ việc phân chia vùng càng chính xác hơn nếu áp dụng vào thực tế, vì trong thực tế các cạnh thường gấp khúc.



Hình 20. Minh họa bài toán Split mặt

Nhóm tác giả đã thử nghiệm và ứng dụng phần mềm bộ công cụ biên tập dữ liệu Topology để xây dựng lưới tam giác TIN hiệu quả ([1]) là nền tảng cơ sở để giải quyết các bài toán như xây dựng bản đồ địa hình; chồng lớp thông tin; xây dựng vùng đệm cho các đối tượng,...từ đó góp phần giải quyết các vấn đề quan trọng trong hệ thống HCM GIS như phân hệ quy hoạch, phân hệ giao thông, phân hệ điện lực, phân hệ bưu điện...

4.KẾT LUẬN

Trên những kết quả đã nghiên cứu và thử nghiệm ở trên, nhóm tác giả nhận thấy dữ liệu sau khi đã được topo hoá (mô hình topology) có thể cho phép xử lý thông tin hiệu quả hơn khi chưa được topo (mô hình hình học thông thường). Nhóm tác giả đã nghiên cứu về công nghệ GIS, hệ quản trị MySQL và xây dựng thành công với những kết quả cụ thể như sau:

- Nghiên cứu rõ ràng, cụ thể và chi tiết về công nghệ GIS, hệ quản trị MySQL và các hệ quản trị CSDL không gian khác.
- Xây dựng thành công bộ công cụ topoeditor với những chức năng cho phép thực hiện các thao tác như thêm, xóa, sửa các thành phần cơ bản cấu thành nên bản đồ: nút, cạnh và vùng trên mô hình topology đã nghiên cứu.
- Bộ công cụ Topoeditor được xây dựng trực quan, thân thiện, dễ sử dụng, đã được áp dụng trực tiếp cho hệ thống HCM GIS của thành phố và tại Trung Tâm HCMGIS, được đánh giá kết quả ứng dụng tương đối tốt so với yêu cầu của hệ thống GIS ứng dụng trên mô hình Topology ([1]).

TÀI LIỆU THAM KHẢO

- [1]. TS. Vũ Thanh Nguyên, *Báo cáo nghiệm thu giai đoạn 1 đề tài xây dựng bộ công cụ biên tập dữ liệu topology hỗ trợ cho hệ thống HCMsGIS*- 21/04/2005.
- [2]. Ordnance Survey, *The GIS files*, (2003).
- [3]. Martin Held, *Algorithmic Geometrie Triangulation*, (2004).
- [4]. Martin Ester, Hans-Peter Kriegel, Jörg Sander, *Knowledge Discovery in Spatial Databases*, (1999).
- [5]. *Vector Product Format Overview*, <http://164.214.2.59/vpfproto>.
- [6]. Cook, Steve, and John Daniels, *Designing Objects Systems: Object-Oriented Modeling with Syntropy*, Prentice Hall, New York, 1994, 389 pp.
- [7]. *Open GIS Simple Features Specification for SQL, Revision 1.1*, <http://www.opengis.org/docs/99-049.pdf>
- [8]. My SQL Technical Reference for Version 5.0.0-alpha.
- [9]. James Arvo, *Graphics Germ*, Vol 2.
- [10]. Petr Kuba, *Data Structures for Spatial Dataming*, (2001).
- [11]. Open GIS™ Abstract Specification, OpenGIS™ Project Documents 99-100 through 99-116, có sẵn tại : <http://www.opengis.org/techno/specs.htm>
- [12]. Dan Sunday, *Basic Linear Algebra*, <http://www.geometryalgorithms.com>.