

CIGALES¹ : L'INTERFACE UTILISATEUR

D. CALCINELLI, M. MAINGUENAUD

FRANCE TELECOM
Institut National des Télécommunications
9 rue Charles FOURIER
91011 EVRY - FRANCE
+ 33 1 60 76 47 82 + 33 1 60 76 47 80 (fax)
Email : calci@int-evry.fr maing@int-evry.fr

Résumé

Dans ce papier, nous présentons l'interface utilisateur de Cigales, langage de requête pour Systèmes d'Information Géographique (SIG). L'interface utilisateur de Cigales fonctionne selon le principe de l'interrogation par l'exemple : l'utilisateur dispose d'outils graphiques avec lesquels il construit une expression graphique symbolisant sa requête. Cette requête graphique est modélisée à l'aide d'un langage fonctionnel pour être ensuite traduite en ordres compréhensibles pour le système de gestion des données.

Nous comparons d'abord trois approches pour l'interrogation de bases de données géographiques. Nous décrivons ensuite l'interface utilisateur de Cigales. Son utilisation est alors illustrée par un exemple de construction de requête graphique. Nous présentons enfin les problèmes posés par cette approche graphique.

Mots clés : Systèmes d'Information Géographiques, Langage Visuel

I. INTRODUCTION

La gestion d'informations géographiques est nécessaire dans des applications aussi diverses que la gestion du réseau téléphonique, le tourisme ou le cadastre. Pour gérer ces applications, un Système d'Information Géographiques doit manipuler à la fois des données spatiales (point, ligne, surface) qui présentent non seulement un aspect géométrique, mais sont également organisées en réseau, et des données alphanumériques (nom de ville, population ...). Un SIG doit fournir à l'utilisateur un langage de manipulation de données capable de gérer ces informations hétérogènes et volumineuses. Le langage Cigales a un double objectif : offrir une interface simple et naturelle pour l'expression des requêtes, et proposer un modèle de données qui cache la représentation physique des données à l'utilisateur.

Nous comparons trois types de langages de requête dans la section II. Nous décrivons dans la section III l'interface utilisateur de Cigales. La section IV présente un exemple d'utilisation de l'interface, en suivant la construction d'une requête. Les principaux problèmes ouverts par cette approche sont présentés dans la section V. Une conclusion est proposée dans la section VI.

¹CIGALES signifie Cartographical Interface Generating an Adapted Language for Extensible Systems

II. LANGAGES DE REQUETE

Nous comparons ici trois types de langages pour l'interrogation de bases de données géographiques [Boursier et Mainguenaud 92] : les langages de type SQL étendu [Bennis 90; Frank 82; Güting 88; Joseph et Cardenas 88; Orenstein et Manola 88; Sacks-Davis 87; Scholl et Voisard 89; Svensson et Huang 91], les langages visuels [Angelaccio 90; Chang 87; Kim 88; Khun 90; Shu 88] et les langages hypermédia géographiques [Boursier 91]. Une base de données exemple est proposée, ainsi qu'une typologie des requêtes s'appuyant sur un ensemble de requêtes. Les principes de base de chacune des trois approches sont présentés. Les avantages et inconvénients respectifs des trois approches sont mis en évidence à partir de l'ensemble de requêtes définies précédemment.

II.1. Base de données exemple

La base de données exemple est constituée de sept relations. Les attributs composant ces relations sont des attributs alphanumériques standard, excepté l'attribut de géométrie [Stemple 86]. Le concept de type abstrait de données (TAD) permet de définir un tel attribut sans connaître sa représentation physique. Le formalisme propre au modèle relationnel est utilisé pour définir les relations de la base de données exemple :

Ville (#ville, nom, population, activité_économique, coût_hôtel, géométrie)
Région (#région, nom, activité_principale, géométrie)
Transport (#transport, nom, type_de_transport, prix, géométrie)
Réseau (#ligne, origine, destination, heure_de_départ, heure_d'arrivée, ≠transport)
Forêt (#forêt, nom, géométrie)
Lac (#lac, nom, géométrie)
Zone_polluée (#zone_polluée, type_de_pollution, géométrie)

La relation Réseau correspond au premier niveau d'abstraction de la relation Transport. Elle permet de voir la relation Transport comme un graphe logique [Mainguenaud 91].

II.2. Typologie des requêtes

Cette section propose sept classes de requêtes.

La classe des requêtes non-spatiales correspond à des opérations de sélection ou de jointure de l'algèbre relationnelle ("Quelles sont les villes de plus de 100 000 habitants ?", "Quelles sont les villes ayant la même activité économique ?").

La deuxième classe concerne la représentation topologique des données ("Quelles régions comportent des enclaves ?").

La classe des requêtes qui s'appuient sur des relations spatiales a été largement étudiée [Sacks-Davis 87] [Güting 88]. Elle concerne les notions de distance, localisation géographique, inclusion, intersection, terminaison, jonction, adjacence et différence ("Quelles villes sont situées au sud de la ville de Paris ?", "Quelles villes sont bordées par une forêt ?").

L'opérateur de négation engendre une classe ("Quelles sont les routes qui ne traversent pas des villes de plus de 100 000 habitants ?").

Les requêtes qui expriment une disjonction constituent également une classe ("Quelles sont les routes qui traversent des villes ou des forêts ?").

L'agrégation peut être définie comme le résultat d'un opérateur ("Quelles sont les routes qui ont une partie urbaine d'une longueur inférieure à 1 km ?").

Enfin, la classe des requêtes de type déductif a été largement étudiée dans le contexte des bases de données [Aho et Ullman 79; Cruz 87]. Elles impliquent le calcul de la fermeture transitive d'une relation - ici la relation Réseau ("Quels itinéraires mènent de Paris à Nice, avec la restriction que la partie entre Lyon et Marseille utilise l'autoroute ?").

II.3. SQL étendu

Le principal avantage du langage SQL est d'être reconnu comme le langage de référence pour l'interrogation de bases de données relationnelles. Normalisé depuis 1986, il permet de répondre aisément aux requêtes non-spatiales. Son extension pour la gestion d'opérateurs géographiques ou de prédicats est généralement bien acceptée (classes de requêtes topologiques ou avec relations spatiales).

Trois grands inconvénients se dégagent de ce type d'approche : (i) le pouvoir d'expression du modèle de données est faible, (ii) l'optimisation des requêtes est complexe, (iii) la gestion des réseaux est mal prise en compte. Les extensions pour gérer des données spatiales ne sont pas normalisées. Aucune règle n'est définie pour gérer le chevauchement logique ou non des formes géométriques entre deux objets de même type (par exemple deux villes n'ont par définition pas d'intersection, alors que rien n'interdit deux zones polluées d'avoir un chevauchement). L'optimisation de requêtes impliquant des opérateurs alphanumériques et géométriques est difficile à réaliser. La gestion de réseaux n'a pratiquement pas été étudiée dans le contexte des langages de type SQL étendu.

II.4. Langages visuels

Un langage est dit visuel lorsque la sémantique de la requête est exprimée par un dessin (et non par une série de clics dans des boîtes pour construire une requête pseudo-SQL). Deux directions peuvent être définies : (i) les langages impératifs, et (ii) les langages déclaratifs. Les langages impératifs définissent étape par étape les opérations élémentaires [Kirby 90]. Les langages déclaratifs définissent les propriétés devant être vérifiées par le résultat final [Mainguenaud et Portier 90]. L'évaluation d'une requête s'effectue en convertissant le dessin en une expression formelle, puis en traduisant cette expression en ordres compréhensibles par le SGBD. Le premier avantage des langages visuels est d'être plus naturel d'utilisation. Les composants élémentaires d'une requête sont graphiques et reflètent les deux dimensions de l'objet qu'ils modélisent. Le deuxième avantage est la notion de combinaison implicite au langage. Cette combinaison permet d'être indépendant de l'application.

Trois inconvénients principaux apparaissent avec ce type de langage : (i) le manque de normalisation, (ii) la complexité d'une requête, et (iii) l'expression de la négation. Le niveau graphique d'interface n'étant pas normalisé, ces langages ont les mêmes limitations que les langages de type SQL étendu. Dans le cas d'une requête avec de nombreuses contraintes spatiales définies entre deux objets, la représentation graphique peut être confuse. De plus, la représentation en deux dimensions peut entraîner des ambiguïtés dans la représentation graphique de la requête [Calcinelli et Mainguenaud 91]. La négation est un concept difficile à exprimer dans un contexte à deux dimensions puisque par définition la relation spatiale n'existe pas.

II.5. Hypermedia

La caractéristique principale des systèmes hypertextes et hypermédia est de laisser l'utilisateur "naviguer" au sein d'un ensemble d'informations reliées entre elles par des "liens". Ces liens sont symbolisés à l'écran par des "boutons" qui apparaissent sous la forme de pictogrammes (ou icônes) ou d'enrichissements typographiques (caractères gras ou italiques par exemple) dans le cas des hypertextes. Le principal avantage de cette approche tient à la nature des technologies hypermédia, qui supprime la contrainte d'apprentissage d'un langage de requêtes. Un autre avantage est son orientation multimédia, qui permet d'établir des liens et de naviguer entre des documents textuels, des données cartographiques, ou encore de l'imagerie aérienne ou satellitaire.

La principale limitation des systèmes hypermédia vient de ce que la notion de requête n'existe pas, et qu'il n'est possible de demander que ce qui a été prévu et préparé en établissant des liens. L'autre inconvénient tient à l'absence de modèle sous-jacent. Le seul concept prédéfini est celui des liens entre entités.

III. DESCRIPTION DE L'INTERFACE

L'interface offre deux objets de base (ligne et zone), ainsi que différents opérateurs spatiaux (inclusion, intersection, adjacence, liaison et distance à vol d'oiseau). La représentation par ligne et zone des objets de base correspond à une métaphore complètement indépendante du modèle géométrique de stockage des objets dans la base de données. Lorsqu'un objet de base est sélectionné, une fonction est invoquée qui permet d'associer une sémantique à cet objet. Cette fonctionnalité confère à Cigales une puissance équivalente à celle des opérateurs de sélection et de jointure du modèle relationnel.

L'expression graphique de la composition de deux objets par un opérateur apparaît dans une fenêtre nommée "fenêtre travail". L'expression graphique de la requête elle-même se construit dans une fenêtre dite "fenêtre requête". La construction s'effectue par étapes successives, chaque étape consistant en la validation de l'expression présente dans la fenêtre travail, par le bouton Validation. Une fois la requête totalement exprimée, le même bouton Validation provoque la traduction de l'expression fonctionnelle associée en ordres compréhensibles par le SGBD. Le bouton Annulation permet soit d'effacer le contenu de la fenêtre travail si elle n'est pas vide, soit d'effacer le contenu de la fenêtre requête dans le cas contraire.

La sémantique des opérateurs est modifiable à travers deux boutons. Le bouton Négation provoque un changement d'aspect des icônes qui représentent les opérateurs en barrant chacun d'eux d'une croix. Ce bouton opère en flip/flop, et en l'invoquant on revient donc dans l'état précédent (absence de négation). Le second bouton permet de choisir entre une vision logique (organisation en réseau) ou purement géométrique des opérateurs. Ce dernier bouton opère également en flip/flop en ayant pour label un mot adapté à la sémantique choisie (Logique ou Géométrique).

L'utilisateur peut également sélectionner des objets dans la fenêtre requête. Cette fonctionnalité est indispensable à la composition d'opérateurs différents agissant sur un même objet. Sachant qu'il doit être possible de sélectionner un ou plusieurs objets à la fois, un premier bouton sert à marquer le début de cette sélection (bouton Début Sélection). Lors de son activation, il se change en bouton Fin Sélection, qui servira bien entendu à marquer la fin de cette sélection. Un second bouton permet la sélection du ou des objets soit dans leur globalité (chaque objet dans sa totalité), soit de façon partielle (i.e., le résultat de l'application préalable d'un opérateur). Ce dernier

bouton opère également en flip/flop en ayant pour label un mot adapté au fonctionnement choisi (Global ou Partiel).

A un niveau plus général, différents boutons permettent de sauvegarder une requête en cours de construction (bouton Sauvegarde), d'accéder à une aide en ligne (bouton Aide) ou de quitter l'interface (bouton Fin).

La figure 1 montre une représentation de l'interface réalisée sous X/Motif.

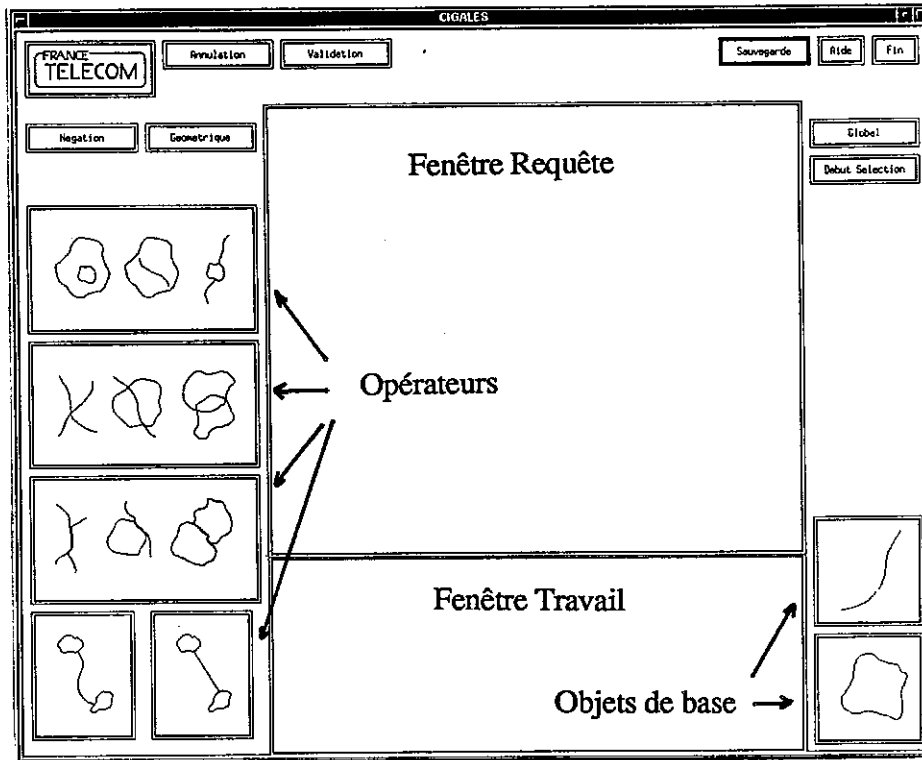


Figure 1

IV. UN EXEMPLE D'UTILISATION

Afin de montrer l'utilisation de l'interface de Cigales, nous allons suivre pas à pas la construction d'une requête qui, quoique complexe, offre l'avantage de présenter les différents composants de l'interface. Cette requête est la suivante : " Quels sont les routes de Paris à Nice, qui ont une partie commune avec l'axe Lille - Bordeaux, et qui ne passent pas près d'un lac dans la première partie du chemin ?", exprimée graphiquement par la figure 2.

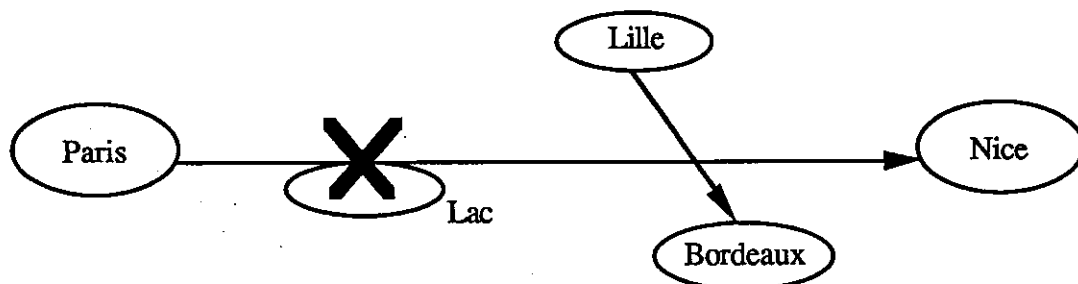


Figure 2

Pour construire cette requête graphique, l'utilisateur sélectionne d'abord une zone parmi les deux objets de base. Cette sélection provoque l'apparition du modèle de données à travers lequel l'utilisateur qualifie l'objet (il s'agit d'une Ville, dont le nom est Paris), puis l'affichage de l'objet qualifié dans la fenêtre travail. Il procède de la même façon pour la ville de Nice, et sélectionne ensuite l'opérateur de liaison. Le modèle de données apparaît de nouveau pour qualifier la liaison, et l'expression graphique de la liaison Paris-Nice s'affiche dans la fenêtre travail. L'utilisateur valide alors cette étape, ce qui provoque l'affichage de la liaison Paris-Nice dans la fenêtre requête, et l'effacement de la fenêtre travail.

En procédant selon le même principe, la liaison Lille-Bordeaux s'affiche également dans la fenêtre requête, à côté de la liaison Paris-Nice. Il utilise ensuite le bouton 'Début Sélection', sélectionne la liaison Paris-Nice dans la fenêtre requête, en concluant par le bouton 'Fin Sélection', faisant ainsi apparaître cette ligne dans la fenêtre travail. Il procède de la même façon pour la liaison Lille-Bordeaux, et positionne alors l'interrupteur de sémantique sur le mode 'Logique'. De ce fait, en sélectionnant l'opérateur d'intersection, il demande que les deux liaisons se croisent au sens du réseau routier, et non pas simplement au sens géométrique. A ce stade, le symbole d'intersection des deux lignes est affiché dans la fenêtre travail, et la validation provoque l'effacement des deux liaisons dans la fenêtre requête, l'affichage des deux liaisons "croisées" dans cette même fenêtre, et l'effacement de la fenêtre travail.

Pour achever la construction de la requête, l'utilisateur sélectionne une nouvelle zone dans les objets de base, qu'il qualifie à travers le modèle de données (il s'agit d'un lac). Il change alors le mode de sélection de 'Global' en 'Partiel', ce qui lui permet de ne sélectionner ensuite que la première partie de la liaison Paris-Nice. Il agit alors sur l'interrupteur de négation. Les icônes des opérateurs sont remplacées par ceux qui expriment la négation, et il reste à l'utilisateur à choisir l'opérateur d'adjacence (nié), puis à valider la fenêtre travail, pour faire apparaître l'expression graphique de la figure 2 dans la fenêtre requête.

V. LES PROBLEMES OUVERTS

La nature graphique de la représentation des requêtes sous Cigales engendre des problèmes sur les plans de la sémantique, de l'ergonomie, ou de l'implémentation.

Les objets et opérateurs graphiques ne permettent pas d'exprimer la sémantique de certaines requêtes particulières comme, par exemple, connaître les enclaves espagnoles en France. Cette requête ne peut s'exprimer ni par l'adjacence, trop permissive car la péninsule ibérique est adjacente mais n'est pas enclavée, ni par l'inclusion, trop restrictive car par définition une enclave espagnole n'apparaît pas dans la représentation géométrique de la France. Dans Cigales, la construction des requêtes ne tient pas compte de l'orientation géographique. Ainsi sur la figure 2, Lille nous apparaît intuitivement au nord de Bordeaux, mais les positions relatives des quatre villes impliquées sont néanmoins erronées. Cette absence de repères cardinaux ne nuit nullement à l'expression formelle des requêtes, mais elle interdit l'accès à la classe des requêtes qui font explicitement référence à la notion d'orientation ("Quels sont les départements au nord de Paris ?").

La négation est un concept difficile à exprimer à l'aide d'une interface graphique, puisqu'il s'agit de montrer une relation spatiale inexistante entre deux objets. Notre choix actuel est de barrer d'une croix le résultat de l'opération. Sur la figure 2, la partie adjacente est barrée. Cependant, cette représentation est moins naturelle dans d'autres cas (inclusion ou intersection de zones).

Les problèmes d'implémentation concernent la fenêtre requête. Les trois problèmes exposés ci-après ne sont - hélas - pas limitatifs. Plusieurs niveaux d'inclusion vont poser des problèmes de précision du tracé, et de représentation d'opérations sur les objets les plus "internes". Si plusieurs opérateurs doivent agir sur un même objet, il faut pouvoir assurer que les autres opérandes n'interragissent pas graphiquement entre eux, à moins que ceci ne soit expressément demandé (et dans ce cas, comment procéder ?). Plusieurs requêtes disjointes peuvent être exprimées en même temps; se pose alors le problème du découpage de la fenêtre en fonction du nombre et des dimensions de ces requêtes.

VI. CONCLUSION

L'interface de Cigales permet de construire des requêtes selon la philosophie de l'interrogation par l'exemple, à l'aide d'outils graphiques simples à manipuler par un utilisateur, et qui autorisent la combinaison des opérations. Les principaux inconvénients sont liés à la représentation graphique de certaines classes de requêtes, à l'absence d'orientation, à l'expression graphique de la négation, et à différents problèmes d'implémentation.

Références

- [Aho et Ullman 79]
"Universality of Data Retrieval Languages", Proceedings of the ACM Principles of Programming Languages Conf., San-Antonio, USA, January 1979.
- [Angelaccio 90]
"QBD* : a Graphical Query Language with Recursion", IEEE Trans. on Software Engineering, Vol. 16, N° 10, October 1990.
- [Bennis 90]
"Geotropics: Database Support Alternatives for Cartographic Applications", 4th Int. Symposium on Spatial Data Handling, Zürich, Switzerland, July 1990.
- [Boursier 91]
"Hypermedia-based geographic information handling", 2nd European Conference on Geographical Information Systems, Brussels, Belgium, April 1991.
- [Boursier et Mainguenaud 92]
"Spatial Query Languages : Extended SQL vs. Visual Languages vs. Hypermaps", 5th Int. Symp. on Spatial Data Handling, Charleston, USA, August 1992
- [Calcinelli et Mainguenaud 91]
"The Management of Ambiguities in a Graphical Query Language for Geographical Information Systems", 2nd Symposium on Large Spatial Databases, Zürich, Switzerland, August 1991 (Lecture Notes in Computer Sciences, No. 525).
- [Chang 87]
"Visual languages: a tutorial and survey", IEEE Software, January 1987.
- [Cruz 87]
"A Graphical Query Language Supporting Recursion", Proceedings SIGMOD Conf., San-Fransisco, USA, May 1987.
- [Frank 82]
"MAPQUERY: Database Query Language for Retrieval of Geometric Data and their Graphical Representation", Computer Graphics, Vol. 16, No. 3, July 1982.
- [Güting 88]
"Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems", Proceedings of Int. Conf. on Extending Data Base Technology, Venice, Italy, March 1988.
- [Joseph et Cardenas 88]
"PicQuery: A High Level Query Language for Pictorial DBMS", IEEE TOSE, Vol. 14, No. 5, May 1988.
- [Khun 90]
"Editing Spatial Relations", 4th Int. Symposium on Spatial Data Handling, Zürich, July 1990.
- [Kim 88]
"PICASSO : A Graphical Query Language", Software -Practice and Experience, Vol. 18, No. 3, pp. 169-203, March 1988.
- [Kirby 90]
"Graphic Map Algebra", 4th Int. Symposium on Spatial Data Handling, Zürich, Switzerland, July 1990.
- [Mainguenaud 91]
"Graph Manipulations for the Network Oriented Management : Application to a Telecommunication Network", 14th Urban Data Management Symposium, Odense, Denmark, May 1991
- [Mainguenaud et Portier 90]
"CIGALES: A Graphical Query Language for Geographical Information Systems", 4th Int. Symposium on Spatial Data Handling, Zürich, July 1990.

[Orenstein et Manola 88]

"Probe Spatial Data Modelling and Query Processing in an Image Database Application", IEEE TOSE, Vol. 14, No. 5, May 1988.

[Sacks-Davis 87]

"GEOQL: A Query Language for Geographical Information Systems", Australian and New Zealand Ass. for the Advancement of Science Congress, Townsville, Australia, August 1987.

[Scholl et Voisard 89]

"Thematic Map Modelling", 1st Int. Symposium on Large Spatial Databases, Santa Barbara, USA, July 1989 (LNCS n°274).

[Shu 88]

"Visual Programming", Van Nostrand Reinhold Cy, New York, 1988.

[Stemple 86]

"Abstract Data Types in Databases: Specification, Manipulation and Access", Proceedings of Int. Conf. on Data Engineering, Los Angeles, USA, February 1986.

[Svensson et Huang 91]

"Geo-SAL: A query Language for Spatial Data Analysis", 2nd Symposium on Large Spatial Databases, Zürich, Switzerland, August 1991 (LNCS n°525).