

# LANGAGES DE REQUETES SPATIALES: SQL ETENDU vs. LANGAGES VISUELS vs. HYPERMEDIA

Patrice BOURSIER (1), Michel MAINGUENAUD (2)

(1) Université Paris-Sud - Laboratoire de Recherche en Informatique (URA 410 CNRS)  
Bâtiment 490 - 91405 ORSAY - FRANCE  
Tél: 33.1.69.41.66.29 Fax: 33.1.69.41.65.86  
E-mail: patrice@lri.lri.fr

(2) France Télécom - Institut National des Télécommunications  
9, rue Charles-Fourier - 91011 EVRY - FRANCE  
Tél: 33.1.60.76.47.82 Fax: 33.1.60.76.47.80  
E-mail: maing@int-evry.fr

## Résumé

Nous comparons les avantages et inconvénients respectifs de trois approches pour l'interrogation de bases de données géographiques: les langages de type SQL étendu, les langages visuels et les hypermédias géographiques.

Une base de données exemple est proposée, ainsi qu'une typologie des requêtes qui s'appuie sur un ensemble de requêtes que tout véritable SGBD géographique devrait être en mesure de traiter.

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 défini précédemment.

## 1. INTRODUCTION

L'interface-utilisateur est un des principaux aspects à prendre en considération lors de la conception de systèmes d'information à référence spatiale. C'est plus particulièrement le cas pour les Systèmes d'Information Géographique (SIG), et ce le sera de plus en plus avec l'extension du champ d'application de l'information géographique numérique vers des applications grand-public telles que le tourisme, ou vers des utilisateurs non-spécialistes pour la gestion de flottes de véhicules (taxis ou pompiers), ou encore pour le marketing géographique.

Trois types d'approches peuvent être utilisées pour concevoir ces interfaces:

- (i) étendre le pouvoir d'expression de langages de requêtes standard, tels que SQL,
- (ii) utiliser des outils graphiques pour concevoir des interfaces-utilisateurs ou des langages de requêtes visuels ou graphiques,
- (iii) utiliser des techniques de type hypertexte ou hypermédia.

La première approche peut être considérée comme une approche recherche, dans la mesure où aucun produit commercial n'est actuellement capable de traiter des requêtes complexes mettant en oeuvre les relations spatiales et la déduction. La seconde approche est également de type recherche, même si les plus avancés parmi les produits de type SIG ont commencé à l'utiliser

*Revue de Sciences de l'Informatique  
Géographique  
et de l'Analyse Spatiale  
Vol n°1, n°2, 1991  
Hermès*

pour la conception d'interfaces-utilisateurs parfois qualifiés d'orientés-objets. La troisième approche n'a jusqu'à présent reçu que peu d'attention de la part de la communauté recherche.

Une base de données exemple est définie dans la section 2. Elle est destinée à évaluer le pouvoir d'expression des opérateurs disponibles dans un SIG. Une typologie des requêtes est proposée dans la section 3, à partir d'un ensemble minimal de requêtes que tout SIG ou SGBD géographique devrait être capable de traiter. Les approches SQL étendu, langages visuels et hypermédia sont étudiées respectivement dans les sections 4, 5 et 6. Une conclusion est proposée dans la section 7.

## 2. BASE DE DONNEES EXEMPLE

La base de données exemple est constituée de huit relations. Les attributs composant ces relations sont des attributs alpha-numériques standard, excepté l'attribut de géométrie [Stemple et al., 1986]. Le concept de type abstrait de données (TAD) permet de définir un tel attribut sans connaître sa représentation physique.

Dans la suite de l'article, l'attribut géométrie représente l'information graphique, indépendamment du modèle logique auquel il se rapporte, c'est-à-dire notamment avec ou sans topologie. Les requêtes définies sur cette base de données ne sont pas concernées par la spécification du modèle logique [Boursier, 1988; Egenhofer, 1989] (i.e. point, ligne, polygone) dans la mesure où la géométrie est considérée comme un TAD. 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)  
Zone\_électorale (#zone\_électorale, 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, conformément à la définition de [Mainguenaud, 1991]. Dans le but de simplifier la présentation, nous ne considérons pas la possibilité d'avoir plusieurs relations pour une même "couche".

## 3. TYPOLOGIE DES REQUETES

Cette section propose sept classes ou familles de requêtes. A l'intérieur d'une même famille, les requêtes proposées correspondent à des besoins croissants, et donc à un pouvoir d'expression de plus en plus grand du langage de requêtes.

### 3.1. Requêtes non-spatiales

La requête A1 correspond à une opération de sélection de l'algèbre relationnelle, la requête A2 à une jointure, et la requête A3 à une clause "group by + having" dans SQL<sup>1</sup>.

A1 : Quelles sont les villes de plus de "100 000" habitants ?

A2 : Quelles sont les villes ayant la même activité économique ?

A3 : Quelles activités économiques concernent plus de "100 000 habitants" ?

---

<sup>1</sup> Cette requête peut s'avérer plus complexe dans le cas d'un SGBD relationnel, à cause des contraintes liées à la 3ème forme normale.

### 3.2. Topologie

Cette deuxième famille de requêtes concerne la représentation topologique des données. La requête B1 recherche les objets comportant des enclaves, la requête B2 a trait à la connexité des objets, et la requête B3 s'intéresse à l'égalité de la représentation géométrique de deux objets (la géométrie peut être considérée comme un attribut quelconque, et dans ce cas la requête B3 est équivalente à la requête A2).

- B1 : Quelles régions comportent des enclaves ?
- B2 : Quelles régions ont une partie de leur territoire située outre-mer ?
- B3 : Quelles sont les régions qui correspondent à des zones électorales ?

### 3.3. Relations Spatiales

De telles requêtes ont été largement étudiées [Boursier, 1987; Sacks-Davis et al., 1987; Güting, 1988]. Les requêtes C1 à C8 concernent respectivement: la notion de distance, la localisation géographique, l'inclusion, l'intersection<sup>2</sup>, la terminaison ("aboutit à"), la jonction, l'adjacence et la différence.

- C1 : Quelles villes sont situées près d'une forêt ?
- C2 : Quelles villes sont situées au sud de la ville de "Paris" ?
- C3 : Quelles villes sont situées dans la région "Ile-de-France" ?
- C4 : Quelles forêts intersectent la région "Ile-de-France" ?
- C5 : Quelles routes aboutissent à la route "RN13" ?
- C6 : Quelle autoroute prolonge l'autoroute "A6" ?
- C7 : Quelles villes sont bordées par une forêt ?
- C8 : Quelles parties de villes n'ont pas d'intersection avec une forêt ?

### 3.4. Négation

La requête D1 montre un exemple d'utilisation de ce concept dans le cas de l'opérateur d'intersection, mais il peut être généralisé à d'autres opérateurs proposés ci-dessus car la négation peut être considérée comme un opérateur unaire.

- D1 : Quelles sont les routes qui ne traversent pas des villes de plus de "100 000" habitants ?

### 3.5. Disjonction

La requête E1 montre un exemple d'utilisation de la disjonction dans le cas de l'opérateur d'intersection, mais cet opérateur unaire peut également être appliqué à d'autres opérateurs. Deux restrictions concernent l'introduction de la fonction d'agrégation et la définition de l'opérateur d'inclusion. La requête E2 implique la possibilité de définir un critère de sélection associé à un opérateur (de manière à éviter le produit cartésien des possibilités).

- E1 : Quelles sont les routes qui traversent des villes ou des forêts ?
- E2 : Quelles routes traversent une ville sachant que, soit la route est la "RN 13" quelle que soit l'activité économique de la ville, soit l'activité économique de la ville est le "Tourisme" quelle que soit la route? [Ooi et Sacks-Davis, 1989]

### 3.6. Agrégation

L'agrégation peut être définie comme le résultat d'un opérateur (par ex. l'opérateur d'intersection pour la requête F1). Ce concept peut être généralisé à d'autres opérateurs ou connecteurs logiques (par ex. >, < or =). Le traitement des requêtes F2 et F3 présente certaines spécificités qui sont liées à la représentation topologique (avec - pour la requête F2 -, ou sans - pour la requête F3 - un recouvrement des géométries).

---

<sup>2</sup> Soit  $\text{Inc}_g$  l'opérateur d'inclusion géométrique entre deux objets. Les spécifications de l'opérateur d'intersection géométrique,  $\text{Int}_g$ , sont alors définies de la façon suivante:  
 $\text{Inc}_g(A, B) = A \Rightarrow \text{Int}_g(A, B) = A$ .

- F1 : Quelles sont les routes qui croisent une fois et une seule l'autoroute "A6" ?  
 F2 : Quelles sont les routes qui ont une partie urbaine d'une longueur inférieure à "1 km" ?  
 F3 : Quelles sont les routes qui ont une intersection avec une zone polluée, la longueur de cette intersection étant supérieure à "1 km" ?

### 3.7. Dédution

Ce type de requêtes a été largement étudié dans le contexte des bases de données [Aho et Ullman, 1979]. Elles impliquent le calcul de la fermeture transitive d'une relation (ici la relation Réseau). Les requêtes G1 à G15 concernent:

- une fermeture transitive (requête G1),
- avec un critère de sélection sur les noeuds (requête G2),
- avec un critère de sélection sur les noeuds sans notion d'ordre (requête G3),
- avec notion d'ordre (requête G4),
- avec négation sur les noeuds (requête G5),
- avec un critère de sélection sur les arcs (requête G6),
- avec un critère de sélection sur les arcs avec ordre (requête G7),
- avec négation sur les arcs (requête G8),
- avec une contrainte sur le résultat d'un opérateur (requête G9, qui peut être généralisée en utilisant d'autres connecteurs: <, >, ...),
- avec une contrainte sur les arcs (requête G10),
- avec une contrainte sur un sous-ensemble du résultat du point-de-vue arc (requête G11),
- avec une contrainte concernant à la fois les noeuds et les arcs (requête G12),
- avec une contrainte sur un sous-ensemble du résultat (requête G13, qui peut être vue comme une généralisation de la clause SQL "group by + having"),
- avec une contrainte sur un sous-ensemble du résultat du point-de-vue noeud (requête G14),
- avec une contrainte sur les noeuds (requête G15).

Ces critères de sélection et ces contraintes peuvent être considérés comme une généralisation de la notion d'"expression régulière" définie dans [Cruz et al., 1987]. Les requêtes suivantes peuvent être définies comme des exemples de récursion "many-sided" (la requête G16 est notamment bien connue dans le domaine des bases de données déductives). La requête G16 illustre le contrôle de profondeur par les noeuds, et la requête G17 le contrôle par les arcs. La généralisation de ces requêtes ainsi que de celles basées sur le concept d'expression régulière se limite aux requêtes évaluables [Mendelzon et Wood, 89]. La requête G18 illustre un problème NP-complet [Garey et Johnson, 1979] qui peut être résolue par l'application de certaines heuristiques. Nous ne détaillons pas les dernières requêtes (G16 à G18) dont le traitement peut conduire à des temps de réponse prohibitifs dans le contexte d'une véritable base de données géographiques contenant des volumes de données importants.

- G1 : Quels itinéraires mènent de "Paris" à "Nice" ?  
 G2 : Quels itinéraires mènent de "Paris" à "Nice", la population des villes visitées devant être supérieure à "100,000" habitants ?  
 G3 : Quels itinéraires mènent de "Paris" à "Nice", "Grenoble" et "Valence" devant faire partie des villes visitées ?  
 G4 : Quels itinéraires mènent de "Paris" à "Nice", "Grenoble" et "Valence" devant faire partie dans cet ordre des villes visitées ?  
 G5 : Quels itinéraires mènent de "Paris" à "Nice", sans passer par "Marseille" ?  
 G6 : Quels itinéraires mènent de "Paris" à "Nice", en utilisant le "TGV" ?  
 G7 : Quels itinéraires mènent de "Paris" à "Nice", quittant "Paris" avec la compagnie "AF" et arrivant à "Nice" avec la compagnie "AI" ?  
 G8 : Quels itinéraires mènent de "Paris" à "Nice", sans utiliser la compagnie "AF" ?  
 G9 : Quels itinéraires mènent de "Paris" à "Nice", avec une seule ville visitée ?  
 G10 : Quels itinéraires mènent de "Paris" à "Nice" en moins de "3 heures" ?

- G11 : Quels itinéraires mènent de "Paris" à "Nice", avec la restriction qu'en cas de correspondance l'heure de départ soit supérieure à l'heure d'arrivée ?
- G12 : Quels itinéraires mènent de "Paris" à "Nice", avec la restriction que le coût global (hôtel + transport) soit inférieur à "500 FF" ?
- G13 : Quels itinéraires mènent de "Paris" à "Nice", avec la restriction que la partie entre "Lyon" et "Marseille" utilise l'autoroute ?
- G14 : Quels itinéraires mènent de "Paris" à "Nice", avec la restriction que l'ensemble de villes visitées soit le même que celui menant de "Lille" à "Nice" ?
- G15 : Quels itinéraires mènent de "Paris" à "Nice", avec la restriction que la correspondance soit inférieure à "3 heures" ?
- G16 : Quelles villes sont accessibles depuis "Paris" avec le même nombre de villes visitées ?
- G17 : Quelles villes sont accessibles depuis "Paris" avec la même distance<sup>3</sup> ?
- G18 : Quels sont les itinéraires hamiltoniens partant de Paris et visitant les villes de "Lyon", "Marseille", "Valence", "Nice" et "Grenoble" ?

Les requêtes proposées ci-dessus représentent les opérateurs de base. Une requête complexe est une requête qui met en oeuvre une combinaison de ces opérateurs, avec l'hypothèse que le SGBD fournisse des opérateurs et pas seulement des prédicats. Les opérateurs offrent en effet un pouvoir d'expression supérieur à celui offert par les prédicats.

Une requête complexe est définie ci-dessous pour servir d'exemple et de test des différentes approches:

- X1 : Quels itinéraires mènent de "Paris" à "Nice", avec la restriction qu'ils longent un lac, puis traversent une ville de plus de "100 000" habitants dans sa partie non-forestière ?

## 4. SQL ETENDU

### 4.1. Principes

Le langage SQL, normalisé depuis 1986, est le langage de référence pour l'interrogation de bases de données relationnelles. Il opère sur un modèle de données simple qui est représenté par un ensemble de relations. Chaque relation est définie par un ensemble de couples (attribut, domaine). Le domaine représente les valeurs autorisées pour un attribut.

Assurément, SQL en tant que tel n'est pas adapté à l'interrogation de BD géographiques. De nombreuses propositions ont été faites dans le but d'étendre son pouvoir d'expression [Bennis et al., 1990; Frank, 1982; Güting, 1988; Joseph et Cardenas, 1988; Orenstein et Manola, 1988; Sacks-Davis et al., 1987; Scholl et Voisard, 1989; Svensson et Huang, 1991]. Deux tendances sont apparues: (i) les langages autorisant l'utilisation d'opérateurs qui permettent de générer des données géométriques en réponse à une requête, (ii) les langages basés sur les prédicats qui n'offrent pas cette possibilité.

Une requête est définie comme une combinaison d'opérateurs élémentaires. Cette combinaison peut être définie par: (i) une séquence d'ordre SQL (approche séparée), (ii) la combinaison d'opérateurs au sein même d'un ordre SQL, dans la clause Select ou/et dans la clause Where (approche globale).

### 4.2. Avantages

Le principal avantage du langage SQL est d'être reconnu (et normalisé) comme le langage de manipulation de données d'un SGBD relationnel. Il permet de répondre aisément aux requêtes alphanumériques (requêtes A1 à A3). Il est très largement utilisé, et son couplage avec des langages de programmation ou de développement d'applications est opérationnel (même si l'implémentation n'est pas globalement bien réalisée puisque l'évaluation syntaxique d'un

---

<sup>3</sup> La notion de distance exprimée ici est différente de celle exprimée dans la requête C1.

ordre est souvent faite à l'exécution). Son extension pour la gestion d'opérateurs géographiques ou de prédicats est généralement bien acceptée (requêtes B1 à B3 et C1 à C8).

### 4.3. Limites

Trois grands inconvénients se dégagent avec ce type d'approche: (i) le pouvoir d'expression du modèle de données relationnel est faible, (ii) l'optimisation des requêtes est complexe, (iii) la gestion des réseaux est mal prise en compte.

Le modèle relationnel est basé sur le concept de troisième forme normale [Codd, 1970]. Cette contrainte est très forte dans le cas d'applications en grandeur réelle (données alphanumériques). Les extensions pour gérer des données spatiales ne sont pas normalisées. Dans le but d'éviter les limitations de la troisième forme normale, de nombreuses extensions ont été définies en s'appuyant sur un modèle orienté-objet (plutôt que sur une extension avec un attribut de type "bulk"). De plus, la définition d'un attribut par un couple (nom, domaine) est très limitative car aucune règle n'est définie pour gérer la clause Select (par exemple l'attribut population d'une ville peut apparaître dans la clause Select définissant l'intersection dans la requête X1). De même, 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). Cette dernière définition a des conséquences importantes sur les opérateurs géographiques (requête F2 ou F3).

L'optimisation du plan d'exécution est très difficile lorsqu'une requête est définie sur des opérateurs géographiques et des opérateurs relationnels [Becker et Güting, 1990; Park et Sagev, 1988]. Suivant le modèle de résolution d'une requête, l'optimisation globale de la requête doit être effectuée par l'optimiseur du SGBD (approche globale) ou hors du SGBD (approche séparée).

La gestion de réseaux n'a pratiquement pas été étudiée dans le contexte des langages de type SQL étendu (requêtes G1 à G16). Une requête peut nécessiter la définition d'une expression régulière (requête G7) ou des contraintes sur des noeuds et/ou arcs (requête G12).

### 4.4. Expression d'une requête complexe

Aucune règle n'est définie pour gérer les résultats alphanumériques, les formes géométriques ou les opérateurs de chemin (l'opérateur de chemin apparaît-il dans la clause Select ? Si oui sous quelle forme ? Est-ce un opérateur logique ou géométrique ? [Calcinelli et Mainguenaud, 1991]).

L'exemple X1 peut être exprimé par la requête suivante:

```
Select *
From Forêt F, Lac L, Réseau R, Ville V
Where Adjacent (Path ("Paris", "Nice"), L.géométrie)
and Intersection (Path ("Paris", "Nice"), Difference (V.géométrie,
F.géométrie))
and Intersection (F.géométrie, V.géométrie)
```

Dans cet exemple, une combinaison d'opérateurs géographiques est nécessaire. Ces opérateurs peuvent apparaître dans la clause Where<sup>4</sup> ou dans la clause Select (puisque une part de la forme géométrique de la ville peut être pertinente dans le résultat final). Le lecteur remarquera que la sémantique de l'étoile n'est pas définie (présence ou non des opérateurs suite à l'indépendance des lacs et des villes, structure du résultat de "path").

---

<sup>4</sup> Pour simplifier la présentation, nous ne prenons pas en compte les propriétés géométriques dans la clause Where (par exemple le rectangle englobant minimum), qui peuvent être définies en tant qu'attributs, ou en tant que données internes d'un type abstrait utilisé par l'opérateur géométrique.

La définition de l'exemple X1 conduit dans le cas d'une approche séparée à la construction d'une liste de requêtes SQL, avec des risques quant à la cohérence globale des résultats:

```
Insert into New_object
  Select      Difference (F.géométrie, V.géométrie)
  From        Forêt F, Ville V
  Where       Intersection (F.géométrie, V.géométrie)

Insert into Path
  Select      Path ("Paris", "Nice")
  From        Réseau

Select      *
From        Path P, Lac L, New_object N
Where       Adjacent (P.géométrie, L.géométrie) and
           Intersection (P.géométrie, N.géométrie)
```

Suivant cette approche, de nombreux problèmes apparaissent. Parmi eux, la définition ou la validation des résultats obtenus dans les relations New\_object et Path (puisque certains objets apparaissant dans ces relations peuvent ne pas satisfaire l'intégralité des contraintes définies dans la requête). Le modèle de résolution nécessite la définition d'un processus de suppression des résultats partiels. La sémantique de la clause IN n'étant pas définie pour les opérateurs spatiaux, cette requête ne peut pas être exprimée de manière globale.

## 5. LANGAGES VISUELS

### 5.1. Principes

Le concept de langage visuel est apparu avec la baisse des prix des périphériques ayant de bonnes capacités graphiques. Des propositions de langages visuels sont détaillées dans [Angelaccio et al., 1990; Chang, 1987; Kim et al., 1988; Kuhn, 1990; Shu, 1988]. 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). Suivant la philosophie des langages de programmation classiques, deux grandes directions peuvent être définies: (i) les langages impératifs, et (ii) les langages déclaratifs.

Les langages impératifs définissent étape par étape les opérations élémentaires. [Kirby et Pazner, 1990] donne un exemple de ce type de langage.

Les langages déclaratifs définissent les propriétés devant être vérifiées par le résultat final. [Mainguenaud et Portier, 1990] est un exemple de ce type de langage. L'évaluation d'une requête s'effectue alors dans un premier temps en convertissant le dessin en une expression formelle. La deuxième phase concerne la conversion de cette expression en ordres compréhensibles par le SGBD.

### 5.2. Avantages

Les deux principaux avantages de cette catégorie de langages sont: (i) ils sont plus naturels à manipuler par un utilisateur non informaticien, et (ii) par nature ils autorisent la combinaison d'opérations.

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 (requêtes C1 à C8). Le modèle de données peut être orienté-objet au plus haut niveau (même si ce n'est pas le cas au niveau du SGBD). Le deuxième avantage est la notion de combinaison d'opérations implicite au langage. Cette combinaison permet d'être indépendant de l'application.

### 5.3. Limites

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.

Deux niveaux peuvent être définis lors de l'utilisation de ces langages: (i) le niveau graphique, (ii) le formalisme logique d'une requête. Le premier niveau n'est pas du tout normalisé, car cette approche est très récente. Le deuxième niveau peut être basé sur un langage de type SQL étendu si la combinaison d'opérateurs est autorisée, sinon un nouveau formalisme doit être défini. Ce formalisme peut être le niveau d'interface entre les applications développées à l'aide d'un langage de programmation classique et le SIG. Puisque le niveau d'interface avec le SGBD n'est 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, 1991].

La philosophie de l'interrogation par l'exemple (par la représentation sémantique des opérateurs et non par la sémantique du modèle de données) nécessite de fournir un exemple des relations spatiales entre les objets. La négation est donc un concept difficile à exprimer dans un contexte à deux dimensions puisque par définition la relation spatiale n'existe pas. Des ambiguïtés peuvent apparaître entre deux objets indépendants (voir par ex. la requête C1).

### 5.4. Expression d'une requête complexe

Pour illustrer cette classe de langages, nous adopterons le formalisme de CIGALES [Mainguenaud et Portier, 1990]. L'exemple X1 est illustré par la Figure 1.

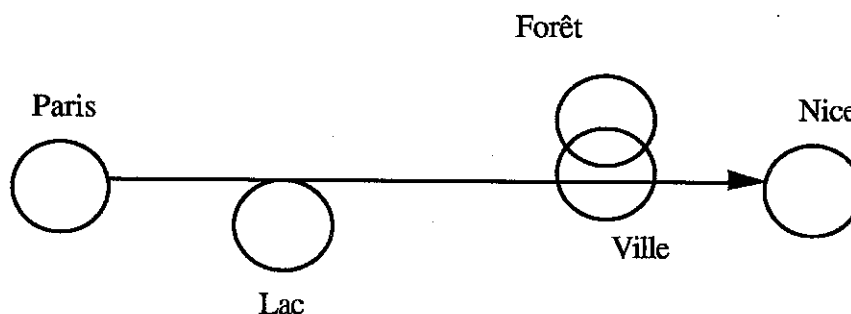


Figure 1: Expression de la requête X1 avec CIGALES.

## 6. HYPERMEDIA

### 6.1. Principes

L'approche hypermédia est relativement récente, et les applications réelles et significatives sont encore peu nombreuses. La notion d'hypertexte date de 1945, avec le "Memex" de Bush [Bush, 45]. Plus récemment, Conklin a proposé un état de l'art assez complet sur le sujet [Conklin, 87].

La caractéristique principale des systèmes hypertextes et hypermédiés 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.

La technologie hypermédia a jusqu'à présent peu touché le domaine de l'information géographique numérique [Laurini et Milleret-Raffort, 1989]. Nous avons utilisé cette technologie pour développer un atlas électronique de la Communauté Européenne sur CD-ROM [Boursier, 1990].

## **6.2. Avantages**

Le principal avantage de cette approche tient à la nature de la technologie hypermédia, qui supprime la contrainte d'apprentissage d'un langage de requêtes. Une seule notion doit être comprise par l'utilisateur: il s'agit de l'utilisation des boutons. Il suffit alors de désigner ce qu'on veut avec la souris (ou avec le doigt dans le cas où l'affichage se fait sur un écran tactile).

Un autre avantage de cette approche est son orientation multi-média, qui permet d'établir des liens et de naviguer entre des documents textuels, des données cartographiques sous forme vectorielle ou "raster", ou encore des photos aériennes ou de l'imagerie satellitaire, en attendant l'image animée et le son [Boursier 1991].

## **6.3. Limites**

La principale limitation des systèmes hypermédiés 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. En d'autres termes, des chemins doivent avoir été précompilés, et l'utilisateur ne peut rien demander d'autre.

L'autre inconvénient tient à l'absence de modèle sous-jacent. Le seul concept prédéfini est celui des liens entre entités.

## **6.4. Expression d'une requête complexe**

Un système hypermédia géographique ne permet évidemment pas d'exprimer la requête  $X_1$ , à moins qu'un bouton " $X_1$ " apparaisse à l'écran et qu'un lien associe directement un objet au bouton, ou qu'une séquence d'instructions associée au bouton permette de construire cette réponse.

## **7. CONCLUSION**

Nous avons comparé dans cet article les avantages et inconvénients respectifs de trois approches pour l'interrogation de bases de données géographiques: l'approche SQL étendu, l'approche langage visuel et l'approche hypermédia.

Une base de données exemple a été définie, ainsi qu'une typologie des requêtes qui s'appuie sur un ensemble de requêtes que tout véritable SGBD géographique devrait être en mesure de traiter.

Les principes de base de chacune des trois approches ont été présentés. Les avantages et inconvénients respectifs des trois approches ont été mis en évidence à partir de l'ensemble de requêtes défini précédemment.

L'approche SQL étendu est celle qui correspond le mieux à l'interrogation d'une base de données géographiques au niveau du SGBD. Elle a toutefois besoin d'être normalisée et étendue au traitement des réseaux, en tenant compte des spécifications du modèle de données sous-jacent. Le développement d'une application géographique peut alors être mené en utilisant un tel langage, à condition qu'il permette la composition d'opérateurs spatiaux au niveau de la clause Select ("approche globale"). Dans le cas d'une "approche séparée", il faut travailler à un niveau supérieur, afin d'assurer la cohérence globale

L'approche langage visuel (ou graphique) est intéressante, car elle est d'une utilisation plus naturelle, et par conséquent plus simple, que l'approche SQL étendu. L'interaction avec le SGBD est toutefois plus complexe à mettre en oeuvre, dans la mesure où le niveau d'abstraction est plus élevé. Cette différence de niveau conduit à définir une expression formelle pour modéliser une requête ayant un pouvoir d'expression plus grand que ne le permettent les requêtes SQL.

L'intérêt de l'approche hypermédia vient de ce qu'elle est plus proche de la façon de penser et de procéder d'un utilisateur "novice" ou occasionnel. Pour cette raison, elle a été utilisée essentiellement pour des applications grand-public sur des terminaux de type borne interactive. Pour cette raison également, le pouvoir d'expression de l'interrogation hypermédia est limité. L'approche hypermédia ne peut être véritablement considérée comme celle d'un langage de requêtes, car elle est limitée aux liens prédéfinis entre entités (cartes, images ou documents photographiques). Il s'agit plutôt de navigation à l'intérieur d'une sorte de réseau sémantique que de langage de requêtes.

Il nous semble que la combinaison de ces différentes approches pourrait conduire à des langages de requêtes plus puissants et moins dépendants d'un modèle de données particulier, et en même temps plus faciles à utiliser. Nous pensons en particulier qu'il serait intéressant de combiner les approches langage visuel et hypermédia au niveau le plus externe, tout en disposant d'un langage standard de type SQL étendu au niveau du SGBD.

## Bibliographie

- Aho A., Ullman J.D., 1979. "Universality of Data Retrieval Languages", *Proceedings of the ACM Principles Of Programming Languages Conf.*, San-Antonio, USA, Janvier 1979.
- Angelaccio M., Catarci T., Santucci G., 1990. "QBD\*: A Graphical Query Language with Recursion", *IEEE Trans. On Software Engineering (TOSE)*, Vol. 16, No. 10, Octobre 1990.
- Becker L., Guting R.H., 1990. "Ruled-Based Optimization and Query Processing in an Extensible Geometric Database System", *Technical Report Universität Dortmund*, 1990.
- Bennis K., David B., Quilio I., Viemont Y., 1990. "Geotropics: Database Support Alternatives for Cartographic Applications", *4th Int. Symposium on Spatial Data Handling*, Zürich, Suisse, Juillet 1990.
- Boursier P., 1987. "Application Planification Urbaine (Urban Planning Application)", *Altair Technical Report*, No. 9-87 (in french), GIP Altair, Rocquencourt, France, 1987.
- Boursier P., 1988. "Analysis of Urban Geographic Queries", *Computer Graphics International Conf.*, Genève, Suisse, Mai 1988.
- Boursier P., et al., 1990. "EUROMAP: a CD-ROM based Digital Cartographic Atlas of Europe", *1st European Conference on Geographical Information Systems (EGIS' 90)*, Amsterdam, Pays-Bas, Avril 1990.
- Boursier P., 1991. "Hypermedia-based geographic information handling", *2nd European Conference on Geographical Information Systems (EGIS' 91)*, Bruxelles, Belgique, Avril 1991.
- Bush, 1945. Bush V., "As we may think", *Atlantic Monthly*, Juillet 1945.
- Calcinelli D., Mainguenaud M., 1991. "The Management of Ambiguities in a Graphical Query Language for Geographical Information Systems", *2nd Symposium on Large Spatial Databases*, Zürich, Suisse, Août 1991 (Lecture Notes in Computer Sciences, No. 525).
- Chang S.K., 1987. "Visual languages: a tutorial and survey", *IEEE Software*, Janvier 1987.

- Codd E.F., 1970. "A Relational Model of Data for Large Shared Data Banks", *Communications of the ACM*, Vol. 13, No. 6, Juin 1970.
- Conklin, 1987. Conklin J., "Hypertext: an introduction an survey", *IEEE Computer*, Septembre 1987.
- Cruz I.F., Mendelzon A.O., Wood P.T., 1987. "A Graphical Query Language Supporting Recursion", *Proceedings SIGMOD Conf.*, San Fransisco, USA, Mai 1987.
- Damier C., Defude B., 1991. "Le prototype OMEGA: un SGBD exensible pour les applications géographiques", *Revue des Sciences de l'Information Géographique et de l'Analyse Spatiale*, Vol. 1, No.1, Ed. Hermès, 1991.
- Egenhofer M., Frank A., 1988. "Towards a Spatial Query Language: User Interface Considerations", *14th Int. Conf. on Very Large Databases (VLDB' 88)*, Los Angeles, USA, Août 1988.
- Egenhofer M., 1989. "A Formal Definition of Binary Topological Relationships", *3rd Int. Conf. on Foundations of Data Organization and Algorithms*, Paris, France, Juin 1989.
- Frank A., 1982. "MAPQUERY: Database Query Language for Retrieval of Geometric Data and their Graphical Representation", *Computer Graphics*, Vol. 16, No. 3, Juillet 1982.
- Gaines, 1988. Gaines B., Vickers J., "Hypermedia design", *RIAO '88 Conf. on User-Oriented Content-Based Text and Image Handling*, Cambridge, MA (Etats-Unis), 1988.
- Garey M.R., Johnson D.S., 1979. "Computer and Intractability - A guide to the theory of NP-completeness", *W.H. Freeman and Co.*, New-York, 1979.
- Güting R.H., 1988. "Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems", *Proceedings of Int. Conf. on Extending Data Base Technology*, Venise, Italie, Mars 1988.
- Joseph T., Cardenas A.F., 1988. "PicQuery: A High Level Query Language for Pictorial DBMS", *IEEE TOSE*, Vol. 14, No. 5, Mai 1988.
- Kim H.Y., et al., 1988. "PICASSO : A Graphical Query Language", *Software -Practice and Experience*, Vol. 18, No. 3, pp. 169-203, Mars 1988.
- Kirby K.C., Pazner M., 1990. "Graphic Map Algebra", *4th Int. Symposium on Spatial Data Handling*, Zürich, Suisse, Juillet 1990.
- Kuhn W., 1990. "Editing Spatial Relations", *4th Int. Symposium on Spatial Data Handling*, Zürich, Suisse, Juillet 1990.
- Laurini R., Milleret-Raffort F., 1989. "A primer of multimedia database concepts", in *Computers and Planning*, Vol. 1, URSA-NET, Athènes, Grèce, 1989.
- Mainguenaud M., Portier M.A., 1990. "CIGALES: A Graphical Query Language for Geographical Information Systems" *4th Int. Symposium on Spatial Data Handling*, Zurich, Suisse, Juillet 1990.
- Mainguenaud M., Raffy J.L., Simatic X.T., 1991. "Graph Manipulations for Network Oriented Data Management - Application to a Telecommunication Network", *14th Urban Data Management Symposium (UDMS)*, Odense, Danemark, Mai 1991.
- Mendelzon A.O., Wood P.T., 1989. "Finding Regular Paths in Graph Databases", *Proceedings of VLDB Conf.*, Amsterdam, Pays-Bas, Juillet 1989.
- Naughton J.F., 1987. "One-sided Recursion", *Proceedings of the 6th PODS Conf.*, San-Diego, USA, Mars 1987.
- Ooi B.C., Sacks-Davis R., 1989. "Query Optimization in an Extended DBMS", *3rd Int. Conf. on Foundations of Data Organization and Algorithms (FODO)*, Paris, France, Juin 1989.

- Orenstein J.A., Manola F.A., 1988. "Probe Spatial Data Modelling and Query Processing in an Image Database Application", *IEEE TOSE*, Vol. 14, No. 5, Mai 1988.
- Park J., Sagev A., 1988. "Using Common Sub-Expression to Optimize Multiple Queries", *Proceedings of the Int. Conf. on Data Engeneering*, Los Angeles, USA, Février 1988.
- Sacks-Davis R., Mc Donnell K.J., Ooi B.C., 1987. "GEOQL: A Query Language for Geographical Information Systems", *Australian and New Zealand Ass. for the Advancement of Science Congress*, Townsville, Australie, Août 1987.
- Scholl M., Voisard A., 1989. "Thematic Map Modelling", *1st Int. Symposium on Large Spatial Databases*, Santa Barbara, USA, Juillet 1989 (LNCS n°274).
- Shu N.C., 1988. "Visual Programming", *Van Nostrand Reinhold Cy*, New York, 1988.
- Stemple D., Sheard T., Bunker R., 1986. "Abstract Data Types in Databases: Specification, Manipulation and Access", *Proceedings of Int. Conf. on Data Engeneering*, Los Angeles, USA, Février 1986.
- Svensson P., Huang Z., 1991. "Geo-SAL: A query Language for Spatial Data Analysis", *2nd Symposium on Large Spatial Databases*, Zürich, Suisse, Août 1991 (LNCS n°525).