

GESTION DE L'HISTORIQUE DES DONNEES DANS LES BASES DE DONNEES LOCALISEES

Par BENMOHAMED Mohamed

Chargé de Recherche, Laboratoire de Géomatique
Centre National des Techniques Spatiales BP13, 31200 ARZEW
benmohamedm@cnts.dz

RESUME :

Les applications spatio-temporelles représentent une part importante des activités de la communauté des sciences géographiques. Si en cartographie classique, cela était possible en utilisant les cartes à différentes dates, l'usage des bases de données localisées comme source d'information ne permet pas très souvent de répondre à ce besoin.

En effet la plupart des bases de données localisées sont bâties sur un modèle statique qui ne comporte que les données les plus récentes. L'usage des bases de données dans les applications spatio-temporelles exige la conservation de l'historique des données lors des opérations de mise à jour.

Le modèle conceptuel de la base de données doit intégrer en plus des composantes de l'espace (x,y,z), **la composante temps (t)**. La conservation de l'historique des données pose certains problèmes en particulier :

- **la complexité de la modélisation.**
- **la complexité des algorithmes de reconstitution des objets.**
- **la gestion d'un volume de données plus important qui croît avec la fréquence et l'importance des mises à jour.**

Lors de la définition des besoins, il est très important de bien sélectionner les informations à conserver dans l'historique, le volume de données et la complexité de la modélisation dépendront de cela.

La conservation de l'historique présente un autre avantage tout aussi important, c'est la possibilité de régénérer en différé l'état antérieur de la BD en cas de pannes ou d'erreurs.

D'une manière générale, il existe deux modèles de conservation de l'historique dans une base de données localisées :

- **modèle version**
- **modèle journal**

ABSTRACT:

Spatiotemporal applications represent an important part of the activities of the community of geographic sciences.

If in classic cartography, it was possible by using maps in various dates, the usage of spatial data bases as source of information does not very often allow answering this need.

Most part of the spatial data bases are built on a static model which contains only the most recent data. The usage of data bases in spatiotemporal applications requires the conservation of the historic of data during the operations of update. The conceptual model of the data base has to integrate both space components (x, y, z), and components **time(t)**

The conservation of the historic of data puts certain problems in particular:

- **The complexity of the modelling.**
- **The complexity of the algorithms of reconstruction of objects.**
- **The management of a more important volume of data which believes with the frequency and the importance of updates.**

During the definition of data bases, it is very important to select information to be preserved in the historic, the volume of data and the complexity of the modelling will depend on it.

The conservation of the historic presents another advantage so important; it is the possibility of regenerating in recording the previous state of the BD in case of breakdowns or errors.

Generally, there are two models of conservation of the historic in a spatial data base:

- Model version**
- Model journal**

1. Introduction :

En cartographie classique, le simple fait de mettre à jour une carte nécessite parfois la reprise de tous les documents et planches qui ont servi à l'élaboration de cette carte, cette opération est donc lourde et coûteuse. En revanche, ce processus présente l'avantage de préserver toutes les anciennes planches de rédaction. Il est donc possible de réaliser des applications spatio-temporelles en combinant les cartes à différentes dates de mise à jour.

Le problème s'inverse pour les bases de données localisées, en effet, la mise à jour des données est une opération courante et très simple à réaliser, détruire, déplacer ou modifier la géométrie ou la sémantique d'un objet ne pose a priori aucun problème en utilisant les logiciels SIG ou CAO-DAO.

Une bonne partie de la rentabilité des bases de données est assurée justement grâce à la souplesse et la rapidité des opérations de mise à jour.

Mais cette relative simplicité de l'opération de mise à jour par rapport à la cartographie classique cache un problème bien plus complexe et toujours d'actualité d'ailleurs.

Une base de données quelle que soit sa nature et son degré de fidélité au terrain, reste aveugle et muette par rapport à des questions du type : « Qu'est ce qu'il y avait à cette endroit à telle date ?, Quel changement a eu entre les dates D1 et D2 dans la zone Z ? »

Pourtant, ce sont des questions que l'on se pose très fréquemment lorsqu'on effectue des applications spatio-temporelles.

Ces requêtes ne peuvent en effet trouver de réponses dans la base de données que si le modèle intègre la conservation de l'historique des données.

La conservation de l'historique des données dans la BD pose le problème du volume de données qui croît avec la fréquence et l'importance des mises à jour. Il est donc nécessaire de bien définir dès le départ ce que l'on doit conserver dans l'historique.

La conservation de l'historique permet également de régénérer en différé l'état antérieur de la BD en cas de problèmes d'incohérence des données dans la BD.

2. Notion de temps :

Sous la pression énorme des besoins des utilisateurs, beaucoup de recherches ont été effectuées dans le domaine de la gestion du temps dans les bases de données géographiques.

Contrairement à la composante « espace » dont les valeurs sont finies⁽¹⁾, la composante « temps » est infinie puisque la limite supérieure est indéterminée⁽²⁾

	composante « espace »	composante « temps »
situation	carte	état
séparé par	polylignes	événements
unité	point, ligne, surface, pixel	année, jour, heure
élément de bases	objet	version
élément mesuré	longueur, surface	durée
position	coordonnées	date
voisins contigus	objets adjacents « objets voisins »	état avant, état après « voisins temporels »
nombre de voisins	infini	deux

Tableau : analogie espace/temps

Par analogie aux requêtes spatiales où la restriction se fait sur les coordonnées (pointé, fenêtrage), l'accès aux données temporelles se fait par restriction sur les attributs représentant le temps. Les restrictions temporelles sont le **pointé temporel (date fixe)**, le **fenêtrage temporel (intervalle de date, durée)**.

Il existe deux modèles de conservation de l'historique dans une base de données localisées :

- La conservation des différentes versions de données à chaque mise à jour (**modèle type version**).
- La conservation de l'historique des opérations effectuées lors de la mise à jour (**type journal**).

3. Modèle « version » :

Il existe également deux manières de conserver l'historique des données en utilisant l'approche « type version ».

3.1 Versions de BD (partitions) :

L'espace géographique étant décomposé en plusieurs partitions, cette méthode consiste à créer après chaque mise à jour une nouvelle version de la partition. Cette méthode est donc un archivage complet d'une partition (géométrie et sémantique).

Si la modélisation des données et des opérateurs est relativement simple, elle présente l'inconvénient de multiplier considérablement la redondance des données.

⁽¹⁾ Les valeurs prises par la composante spatiale sont comprises dans le rectangle englobant l'emprise de la région couverte par la BD.

⁽²⁾ La borne supérieure de la composante « temps » n'est pas fermée, elle est représentée par une date variable

Date = « maintenant ».

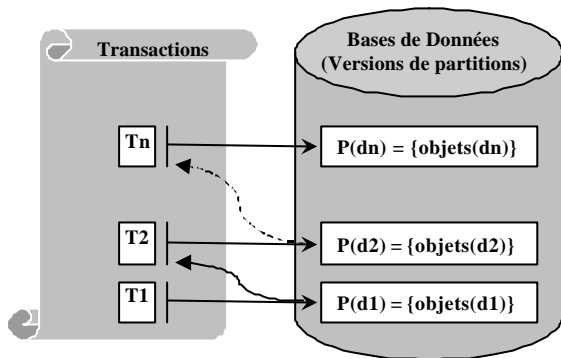


Figure 1 : Principe des versions de partitions

La structure minimale pour cette approche est :

Partition (version)	
-	Identifiant de la partition courante
-	Liste des objets
-	Date D_i de la version ⁽³⁾

L'algorithme général de recherche de l'état d'un objet à une date d est :

Tant que $d_i.partition > d$
Lire partition(v_i)
Déincrémenter i à $i-1$ ($1 < i < n$)
Fin (tant que)
Objet(d) = objet.partition(v_i)

3.2 Versions d'objets :

Contrairement à l'approche précédente, celle-ci opère directement sur l'objet et non sur toute la partition. Chaque transaction $T(i)$ (création, modification) effectuée à la date $D(i)$ sur un objet crée une nouvelle version $V(i)$ de cet objet, la destruction met fin à l'existence de l'objet à la date courante (figure 2).

Cette approche offre moins de redondances que la première puisqu'il y a moins de duplication des données, les modèles sont cependant plus complexes à réaliser.

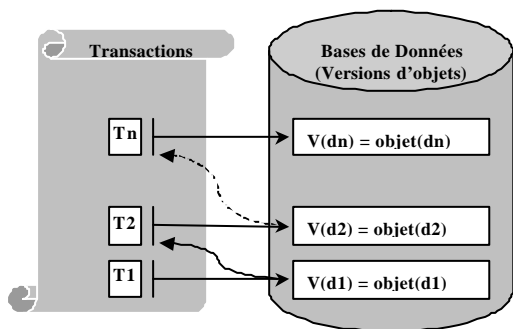


Figure 2 : Principe des versions d'objets

Pour cette approche, la structure minimale est

Objet (version)	
-	Identifiant de l'objet courant
-	Description de l'objet
-	Localisation de l'objet
-	Date D_i de la version

L'algorithme général de recherche de l'état d'un objet à une date d est :

Tant que $d_i.objet(v_i) > d$
Lire objet(v_i)
Déincrémenter i à $i-1$ ($1 < i < n$)
Fin (tant que)
Objet(d) = $d_i.objet(v_i)$

Pour les deux approches « versions », les algorithmes de gestion de l'historique sont relativement moins complexes, mais la duplication des informations inchangées pose le problème de la redondance des données et à fortiori le volume des données qui augmente avec la cadence des mises à jour et la quantité d'information traitée.

4. Modèle « Journal » :

Plutôt que de dupliquer les données sémantiques ou géométriques, le principe de la conservation de l'historique « type journal » dans la base de données consiste à sauvegarder dans un journal toutes les opérations (transactions) effectuées sur l'objet avant de remplacer l'ancienne version de l'objet par la nouvelle (figure 3).

Dans une structure « type journal » seules les dernières versions des objets sont définies dans leur propre structure. Les structures des anciennes versions sont reconstituées en exécutant les opérations dans le sens inverse.

Le mécanisme de gestion des transactions longues dans une base de données historique s'effectue sur le même principe que les transactions courtes à la différence fondamentale entre les deux est que les transactions courtes ne sont pas persistantes dans le temps contrairement aux transactions longues qui le doivent impérativement.

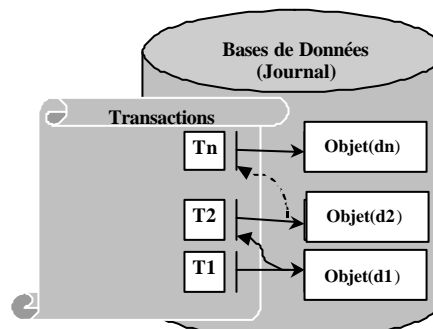


Figure 3 : Principe du journal

⁽³⁾ la date peut avoir deux natures : a) date d'existence sur le terrain, b) date de mise à jour dans la BD

La structure minimale du journal est :

Journal
- Identifiant de l'objet courant
- Type de transaction ⁽⁴⁾
- Date de transaction
- Information sur les données modifiées ⁽⁵⁾

L'algorithme général de recherche de l'état d'un objet à une date **d** est :

```
Lire état courant (dn)  
Si d ≙ dn alors objet(d) = état courant(dn)  
Sinon  
  Tant que d < di  
    Lire journal (di)  
    Calculer état objet (di)  
    Déincrémenter i à i-1 (1 < i < n)  
Fin (tant que)  
  Objet(d) = état objet(di)  
Fin (si non)
```

Dans la structure journal, seules les dernières versions des objets sont définies dans leur propre structure, la structure des versions antérieures est reconstituée par l'exécution de la transaction du journal.

Le fait de ne sauvegarder que les informations qui sont concernées par la transaction minimise les redondances, l'encombrement est donc moins important que dans le type version.

Cependant les états antérieurs des objets n'étant pas sauvegardés dans leur propre structure, les versions antérieures sont donc reconstituées par calcul, les algorithmes de reconstitution sont assez complexes à mettre en oeuvre.

Conclusion :

L'inclusion de l'historique dans une base de données fait augmenter son volume d'une manière assez importante, qui croit avec l'importance de la mise à jour. A terme, le volume des données peut dépasser les 50 % du volume initial de la BD. Il est donc nécessaire d'optimiser à la fois la structure de données et les opérations, particulièrement pour les bases de données de grande extension.

La modélisation de l'historique par la méthode des versions présente l'avantage d'être intuitive, donc relativement simple à réaliser mais présente l'inconvénient de dupliquer fortement les données.

La méthode journal permet en revanche de remédier à ce problème, mais la modélisation est assez complexe à mettre en œuvre.

On pourra éventuellement combiner les deux approches pour à la fois réduire les redondances et simplifier les procédures. La recherche dans le domaine des bases de données localisées spatio-temporelles est toujours très active.

Références Bibliographiques :

- M. BENMOHAMED, 93 : *Etude des bases de données localisées de grande extension*, thèse de doctorat, IGN, Université Paris 7.
- G. LANGRAN, 89 : *time in Geographic information System*, Taylor & Francis.

⁽⁴⁾ type de transaction : création, destruction, modification, transfert, scission, fusion

⁽⁵⁾ arcs créés, changés ou détruits, attributs changés, etc...