

# L'intégration du temps dans les SIG

■ Patricia BORDIN

*Les SIG, les systèmes dédiés à la gestion et aux traitements des informations géographiques, peinent à devenir de véritables outils d'aide à la décision. En effet, pour aider au suivi, à l'analyse et à la compréhension des phénomènes qui se jouent sur les territoires, il est indispensable de pouvoir étudier conjointement la dimension géographique et la dimension temporelle. Or, en partie à cause de leur origine d'outils dédiés à la cartographie, mais surtout à cause de la complexité technique de la modélisation temporelle, les SIG proposent souvent une représentation sommaire des évolutions. Pourtant des solutions existent pour améliorer leurs performances. Dans cet article, nous verrons comment certaines agissent a posteriori et d'autres interviennent dès la constitution de la base de données au niveau de la modélisation.*

## MOTS-CLÉS

Observations spatio-temporelles, SIG, modélisation, historique

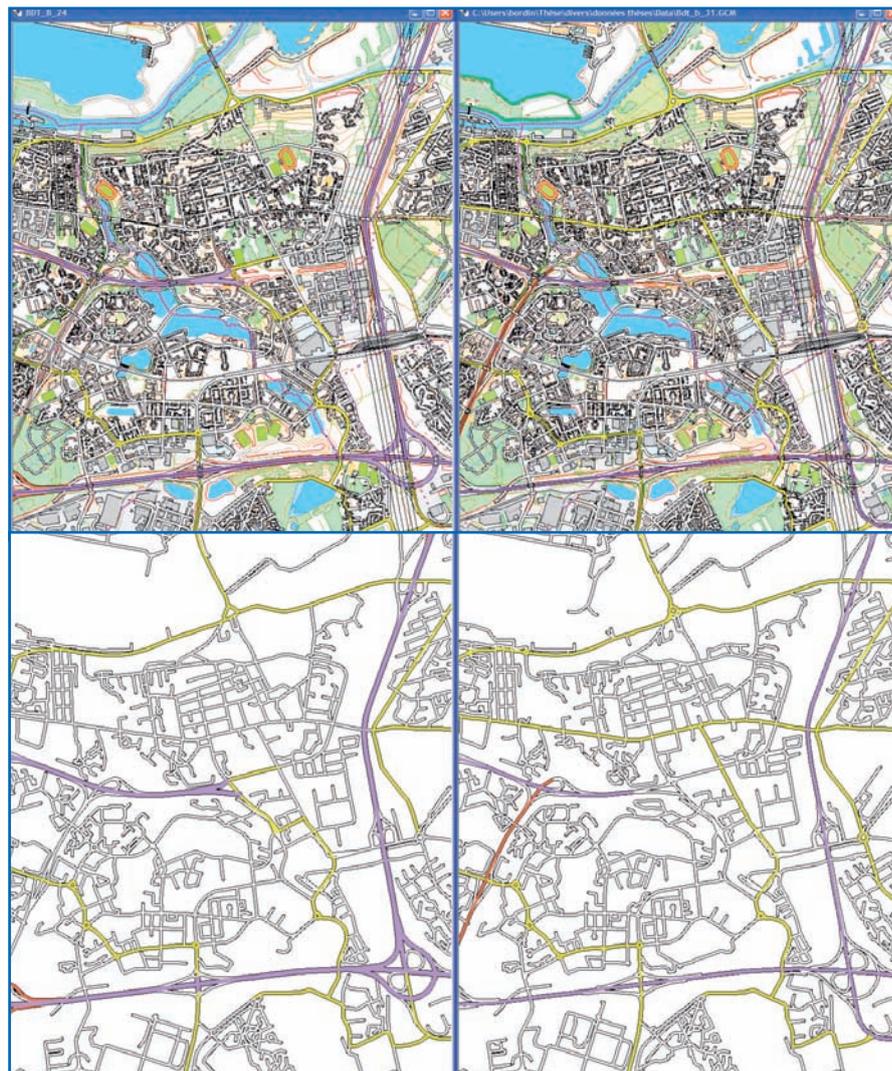
## Observations temporelles et SIG

Quand un utilisateur souhaite observer la dimension temporelle dans un SIG, il opère souvent comme avec des cartes papier : il affiche la base de données dans deux états différents et compare deux à deux les "objets" des "images" (comme dans le jeu classique des "sept erreurs", à la différence près qu'il ne sait pas d'avance le nombre de changements à trouver). Lorsque ces images sont au format raster, l'utilisateur peut exploiter un logiciel dit de "traitements d'images", pour tenter une comparaison automatique. Cependant, les conditions de prises de vues, la saison, l'heure, etc. variant généralement d'une fois sur l'autre, les changements de couleurs ou de niveaux de gris sont parfois mal interprétés. Ainsi, malgré des résultats prometteurs, l'extraction automatique des évolutions n'est pas encore totalement opérationnelle (en particulier, il reste difficile d'extraire automatiquement les évolutions du bâti).

Lorsque les "images" sont au format vecteur, l'utilisateur bénéficie du format "objet" de ce mode de représentation : des requêtes simples lui permettent de sélectionner et d'afficher uniquement les "objets" qui l'intéressent, ce qui allège d'autant son observation. Dans la figure 1 par exemple, l'utilisateur concerné uniquement par le "réseau routier", peut choisir de n'observer que ce type "d'objet" au lieu de comparer des

états comportant l'ensemble des données de la BD TOPO®. Cependant, même dans ces conditions, le travail

d'identification des évolutions reste long et fastidieux, sans garantie de qualité (d'exhaustivité).



**Figure 1.** Identification par observation visuelle des changements intervenus entre deux états de la BD TOPO® sur la totalité de la base (en haut) sur un extrait portant uniquement sur le réseau routier (en bas).



L'identification des changements sémantiques intervenus sur un type "d'objet" particulier est encore plus laborieuse. Elle implique d'observer autant de cartes thématiques qu'il y a d'attributs pour ces "objets". Ainsi

dans l'exemple précédent, pour identifier les changements sémantiques du "réseau routier", l'utilisateur devra réaliser  $n$  couples de cartes pour pouvoir observer les changements sur les  $n$  différents

attributs décrivant le "réseau routier" dans la BD TOPO® (nombre de voies, classement de la route, sens de circulation...) (figure 2).

Or, le suivi d'un phénomène géographique nécessite de pouvoir prendre en compte la dimension temporelle dans l'observation. La qualité des observations dans un SIG dépend donc de sa capacité à intégrer cette dimension. Toutefois, les SIG dédiés aux informations spatiales, sont mal adaptés à la gestion et à l'analyse des informations spatio-temporelles. Des travaux de recherches fournissent des éléments de solutions, mais l'observation d'un phénomène géographique dans un SIG reste une application délicate pour la plupart des utilisateurs. Cela explique probablement pourquoi beaucoup se contentent d'une observation visuelle.

## Intégration des évolutions dans les SIG : La mise à jour

### ■ Le temps dans les SIG : point de vue conceptuel

Le problème du temps dans les SIG résulte de la nécessité de lier simultanément : dimension spatiale et dimension temporelle. Cette difficulté n'est pas propre aux SIG, mais la prépondérance qu'ils accordent à la dimension spatiale a entraîné une modélisation plus sommaire de la dimension temporelle. Dans la pratique, peu de solutions SIG prévoient autre chose qu'un attribut de type "date" pour intégrer le temps. Ainsi, la plupart des utilisateurs s'appuient implicitement sur la représentation du temps classique, linéaire, orientée, discrète pour situer une suite d'états. Ce modèle possède une qualité (en dehors de sa simplicité) : il étend aux bases de données le modèle utilisé en cartographie classique, et permet ainsi des analogies (les versions des bases correspondent aux éditions des cartes). En contrepartie, cette simplification conceptuelle tend à ancrer les SIG dans leur rôle originel d'outils de cartographie, avec pour effet de les cantonner à rester des cartothèques un peu sophistiquées plus que de véritables outils d'analyse.



Figure 2. Identification visuelle des changements sémantiques intervenus sur les "tronçons de routes" de la BD TOPO®.



Pour intégrer le temps dans un SIG, il faut passer par une modélisation. Elle simplifie le problème des relations spatio-temporelles pour les rendre accessibles aux requêtes spatiales, à l'analyse et à la visualisation.

Cette modélisation introduit toutefois des difficultés supplémentaires pour celui qui veut observer un phénomène géographique avec un SIG. De façon un peu caricaturale, on peut en effet affirmer que : le temps des informaticiens n'est pas celui des thématiciens. L'implémentation du temps dans les SIG ne correspond pas exactement à l'observation de phénomènes géographiques, et plus généralement, l'observation de données sur un phénomène n'équivaut pas à l'observation du phénomène lui-même. En effet, l'opération de mise à jour des données consiste à introduire des changements dans la base de données pour refléter au mieux la réalité du terrain. Elle ne reflète pas exactement les évolutions de ce terrain. Elle peut par exemple introduire des changements de type correction d'erreurs qui ne correspondent à aucune modification dans la "réalité". Par ailleurs, suivant le niveau de sophistication du modèle utilisé pour gérer les changements de la base de données (dit modèle de gestion des mises à jour) l'utilisateur observera des changements plus ou moins radicaux (certains modèles ne saisissent que des créations / destructions, d'autres envisagent des modifications plus subtiles). Poser l'observation des phénomènes géographiques avec un SIG revient à considérer que l'aide à la gestion et à la création d'informations nouvelles sur le phénomène, compensera le décalage entre :

- l'observation géographique souhaitée
- l'observation réalisée avec les données.

### ■ La mise à jour : point de vue technique

Les thématiciens et les techniciens ont des points de vue différents sur la dimension temporelle dans les SIG :

- pour les thématiciens, il s'agit de pouvoir suivre des objets géographiques, d'analyser des évolutions et d'étudier le temps comme composante du phénomène observé.

- les techniciens, quant à eux, considèrent le temps plutôt comme un paramètre externe qui affecte la qualité des données qu'ils administrent. Il leur impose d'introduire régulièrement des changements dans ces données, au risque pour les bases de données de devenir sinon obsolètes. L'intégration de la dimension temporelle est alors principalement une opération de maintenance (temporelle) des données. Cette opération est appelée la mise à jour. Elle permet d'établir un lien entre l'observation des données et celle du phénomène géographique étudié. Cependant, elle n'équivaut pas à l'introduction de la dimension temporelle ; elle demeure fondamentalement technique. Les changements introduits dans les données ne résultent pas uniquement d'évolutions sur le terrain. Ils correspondent à des aspects techniques de maintenance et de gestion des données. Il existe plusieurs types de mises à jour de maintenance technique. Certaines n'agissent que sur les données, d'autres interviennent également au niveau de la structure de la base de données elle-même.

Les mises à jour de maintenance sont soit :

- des mises à jour intervenant sous la forme de correction d'erreurs, qui introduisent des changements au niveau des données. Elles sont difficiles à identifier, car elles apparaissent comme des évolutions "vraies". On trouve soit :
  - des corrections d'erreurs vraies, par exemple elles compensent l'oubli d'un "objet géographique", elles rectifient la valeur d'un attribut ou la forme d'une emprise, etc. Après ces mises à jour la qualité des données est meilleure;
  - ou des corrections dues à des différences d'interprétations des règles de saisies. Ces mises à jour n'améliorent en général pas la qualité des données. En effet, dans certains cas ambigus ou du fait d'un défaut d'interprétation, l'opérateur de saisie des données peut décider d'introduire des corrections. Pensant intro-

duire une "vraie" mise à jour ou une correction d'erreurs, il introduit alors un changement d'une autre nature.

- des mises à jour introduisant des modifications dans la structure des données (en plus de celles sur les données elles-mêmes). Ces mises à jour sont généralement plus faciles à repérer (la disparition d'un attribut, l'apparition d'un nouvel "objet" ou d'une nouvelle couverture de zone se voient clairement). Ces mises à jour regroupent :
  - des changements de spécifications. La décision du producteur des données de modifier les spécifications de la base de données implique des changements dans la base de données "pour la mettre à niveau", c'est-à-dire pour la rendre conforme aux nouvelles règles qu'elle doit respecter;
  - des enrichissements de la base. Ces enrichissements peuvent prendre la forme d'une introduction de données supplémentaires, par exemple l'ajout d'un simple attribut sur un "objet géographique" (ex. : le trafic sur le "réseau routier") ou d'un nouvel "objet" (ex. : les "sentiers piétonniers"), voire d'un nouveau thème (ex. : l'hydrographie, la végétation). Ils peuvent également prendre la forme d'une extension de la zone couverte par la base.

### ■ La gestion des mises à jour : point de vue pratique

La mise en œuvre des mises à jour soulève de nombreux problèmes. Beaucoup portent sur des aspects organisationnels concernant la collecte, l'intégration ou encore la livraison des mises à jour. Nous n'évoquerons ici que les aspects de modélisation et d'implémentation, pour montrer les enjeux de ces modèles pour l'observation de phénomènes géographiques. Nous adopterons une définition qui ne considère que des modifications en rapport avec des évolutions du terrain physique : La mise à jour est alors l'opération qui tend à ajuster les données à l'actualité de la situation sur le terrain.





### Modèles d'implémentation et de gestion des mises à jour

Il existe plusieurs modèles pour intégrer les changements effectués sur les données d'une base de données. Ces modèles possèdent des qualités différentes en termes de :

- commodité d'accès aux différents états temporels d'un même "objet" au sein de la base,
- explicitation des liens existant entre ces différents états temporels, pour aider à établir la série temporelle de ces états,
- caractérisations de ces liens décrivant les changements opérés, il s'agit ici de posséder des informations sur le changement lui-même, comme par exemple son type (création / destruction / modifications) ou sa date d'occurrence, voire son origine, sa cause.

Dans ce qui suit nous présentons quatre modèles qui permettent d'illustrer en quoi l'observation d'informations temporelles dans un SIG dépend des modèles de gestion mis en œuvre. Les quatre modèles sont présentés en fonction de la richesse d'informations qu'ils fournissent sur les changements de données et de la facilité à reconstituer la suite des états successifs d'un "même" "objet". Cette présentation suit un ordre croissant relativement bien corrélé avec la complexité de mise en œuvre. Les modèles que nous avons retenus sont :

*Le modèle par snapshots* qui gère la mise à jour des bases de données comme une mise à jour de carte papier : à chaque mise à jour correspond une version de la base. Chaque version est indépendante des autres, il n'y a aucun lien entre les données des différentes versions.

*Le modèle par time stamp*, qui archive différents états "d'objets". Dans ce modèle la base de données peut être vue comme une "donnéethèque" sorte d'équivalent d'une cartothèque pour les données ; c'est une collection de représentations "d'objet géographique". Deux représentations différentes sont indépendantes ; même si elles correspondent dans la "réalité" à deux états d'un même objet ; il n'existe pas de lien formalisé entre elles.

*Le modèle par versionnement d'objet* qui archive différentes versions "d'objets". Ce modèle considère qu'un même "objet" peut avoir plusieurs états et formalise un lien (par le biais d'un numéro de version) entre les différentes versions de cet "objet". En revanche pour éviter les informations redondantes, chaque version ne comporte qu'une description partielle de l'état de la donnée correspondant aux caractéristiques modifiées.

*Le modèle par historique* qui introduit le concept de successeur. Chaque "objet géographique" est décrit dans sa totalité. S'il change d'état, il existe alors un lien formalisé par le biais d'un identifiant pour spécifier l'état qui lui "succède".

#### Le modèle par snapshots ou par estampillage de la base de données

Le modèle par snapshots, qui reprend la métaphore de la prise de vue instantanée, est le modèle le plus élémentaire. Dans ce modèle, la mise à jour d'une base de données est conçue comme un processus ponctuel, pendant lequel l'ensemble des données est "mis à jour" avec pour résultat une base de données à jour. Cette base, sorte de représentation statique (ou d'instantané) à une date d fixée (dite date de mise à jour), est archivée en tant que version de la base à la date d. Les différentes versions de la base se succèdent alors, sans qu'il existe de lien automatique entre les différentes versions, celles-ci étant conçues comme des jeux de données autonomes.

La simplicité du modèle explique son succès. Conceptuellement, il correspond au modèle classique d'observa-

tion discrète, statique (que nous expérimentons régulièrement, ne serait-ce qu'en regardant un film). Pour les utilisateurs habitués aux cartes papiers, il transpose le mode traditionnel de mise à jour en cartographie. Pratiquement, il est simple à implémenter. La gestion des évolutions se limite à leur collecte et à leur intégration sous forme d'intervention pour modifier les données ; il n'y a pas de gestion des données d'évolutions (c.a.d. on ne traite pas de questions telles que : "Quelles sont les données qui ont changé ? En quoi ont-elles changé ? Pourquoi ont-elles changé ?"). Le modèle par snapshots n'implique pas d'étudier préalablement comment modéliser, collecter, exploiter, analyser la dimension temporelle : il est le plus simple à livrer et simple à recevoir.

En revanche, sa simplicité limite ses performances pour le suivi de phénomènes dans un SIG. Il ne comporte en effet aucune donnée d'évolution, c'est-à-dire aucune information sur les modifications effectuées sur les données, qui pourraient servir de "trace" d'évolutions réelles. En fait, ce modèle offre le minimum d'information temporelle : il ne propose pas d'autre information que la date d'estampillage associée au jeu de données. Il arrive que cette date ne soit même pas implémentée dans les données. Elle n'apparaît alors que sous la forme d'une inscription sur le support ou le bon de livraison ! Dans ce cas, si un champ (ou attribut) de type "date d'estampillage" n'est pas préalablement introduit et automatiquement rempli avec la bonne valeur et ce, pour les différentes versions de la base de

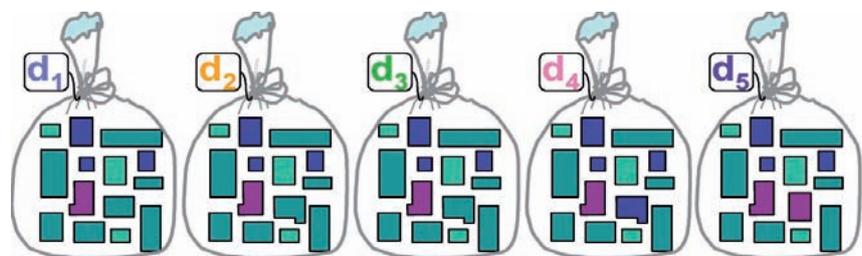


Figure 3. Principe du modèle par snapshots

Le snapshots archive les différentes versions de la base de données de façon autonome. Sans identifiant, les jeux de données temporels sont indépendants. Rien ne permet d'identifier automatiquement un "objet" dans la base de données. Il faut le repérer et le suivre visuellement dans les différentes versions.



données étudiées, alors lorsque l'utilisateur affiche simultanément deux versions pour pouvoir comparer visuellement deux états différents de la base, il n'a aucun moyen de distinguer de quelle version sont issues les données. Le modèle par snapshots peut toutefois être amélioré simplement et à moindre coût en introduisant un identifiant ou une signature. Cette opération qui consiste à créer un attribut supplémentaire à chaque "objet", permet alors par simple requête de sélectionner les "objets" dont l'identifiant se trouve dans une version antérieure mais est absent des versions suivantes (destruction) (et réciproquement les "objets" dont l'identifiant est absent d'une version antérieure mais présent dans celles qui suivent (création).

#### **Le modèle par time stamp ou par estampillage des données**

On peut enrichir le modèle basique par snapshot en ajoutant pour chaque "objet" un identifiant et deux champs supplémentaires : "date de création" et "date de suppression". On obtient alors le modèle par time stamp. Il permet d'intégrer dans une même base de données l'ensemble des données géographiques issues des différentes mises à jour. Cette base de données unique regroupe alors toutes les données (passées et actuelles) décrivant un même territoire. La date de création et de suppression étant attachées à chaque "objet", on peut alors envisager des dates qui ne soit plus liées à des versions de cartes mais qui soient des dates ayant une signification plus proches des objets eux-mêmes. Ce modèle n'implique en effet pas une mise à jour ponctuelle et sait au contraire profiter d'une mise à jour en continu. Ce modèle permet facilement de connaître les objets présents ou absents à une date fixée, ainsi que la durée de vie d'un objet particulier. Techniquement, il implique une gestion d'un volume important de données, mais reste simple à implémenter.

#### **Le versionnement par objet**

Dans le modèle précédent, chaque état d'un même "objet" est enregistré comme une donnée avec sa période de validité donnée par le couple {"date de création" ; "date de suppression"}. Si

l'on choisit de travailler avec un identifiant strict (notion d'identifiant strict, index ou clé en informatique), c'est-à-dire de changer d'identifiant dès que cet état subit un changement alors les différents états d'un "même" "objet géographique" sont indépendants les uns des autres. Il n'existe pas de relation entre eux. Chaque identifiant renvoie à un "objet" unique dans un certain état. Mais on peut aussi choisir de travailler avec un identifiant souple (identificateur). Un identifiant souple est affecté à un objet. Il sera associé à toutes les représentations de l'objet, à ces différents états dès lors que les modifications ont préservé l'"identité" de l'objet. On peut alors avoir dans la base de données différents états d'un même objet. Il devient possible de reconstituer les liens de succession par requêtes sur la date de création (ou la date de suppression) (ex. : "Sélectionner les objets dont l'identifiant est XXX et trier-les par ordre croissant en fonction de la "date de création"). Dans ce cas, le lien de succession n'est pas direct : pour l'obtenir il faut effectuer des requêtes spécifiques.

Le modèle par versionnement propose de formaliser ce lien, en introduisant un numéro de version. Pour obtenir la série temporelle des états, il suffit alors de faire les chaînages des différents numéros de versions, à valeur d'identifiant constante. Ce modèle a été élaboré pour réduire le nombre de données archivées et pour éviter d'avoir à gérer un volume important d'informations (comme c'était le cas dans le modèle précédent) ; il optimise le stockage des données en ne dupliquant pas les données redondantes. Ainsi, la description de chaque version "d'objet" ne comporte que les informations nouvelles (modifiées) mais pas celles qui n'ont pas changé. La reconstitution de l'état complet d'un "objet géographique" met en œuvre un mécanisme de gestion interne à ce système qui permet de remonter les informations au travers des différentes versions. Dans la pratique, pour obtenir la série temporelle des différents états d'un "objet", il est nécessaire de reconstituer la description de chaque état, ce qui dans la pratique est fastidieux. L'exploitation de ce modèle peut être simplifiée en acceptant d'in-

roduire des informations redondantes et d'associer à chaque version la description entière de l'état de "l'objet". On obtient alors un modèle très proche du modèle par time stamp : seul le numéro de version les distingue.

#### **L'historique des données**

La mise à jour par gestion de l'historique des données est l'une de celles qui s'attachent le plus à l'aspect temporel. Elle ne vise pas seulement la gestion technique des changements dans les données. Elle propose d'établir un lien directement accessible entre les différents états d'un "objet". Elle ébauche ainsi une première solution exploitable par les utilisateurs pour faire du suivi d'évolutions. Ce modèle est proche du modèle par time stamp avec identifiant, mais il introduit en plus la notion de successeur. Il associe ainsi systématiquement à tous les objets : un identifiant et les deux attributs "date de création" et "date de suppression" mais également une "date de modification" et un "identifiant du successeur". Lorsqu'une donnée doit être modifiée, il faut alors définir si :

- l'objet géographique représenté par la donnée a disparu et s'il a été remplacé par un autre (ex. : destruction d'un bâtiment et construction d'un nouveau). Dans ce cas, on remplit le champ "date de suppression" avec la date à laquelle "l'objet géographique" a "disparu" et on crée une nouvelle donnée pour représenter le nouvel "objet géographique", avec un nouvel identifiant et une "date de création" correspondante.
- ou si l'objet géographique décrit par la donnée ayant évolué, la donnée n'est plus une bonne représentation de cet objet (ex. : réhabilitation et agrandissement d'une construction). Dans ce second cas, on remplit le champ "date de modification" avec la date à laquelle "l'objet" géographique a "évolué". On inscrit dans le champ "identifiant du successeur" la valeur de l'identifiant attribué à la nouvelle donnée créée pour le successeur. Cette donnée reçoit pour valeur de champ "date de création", la date à laquelle l'objet géographique considéré a évolué.

Ce modèle qui simplifie le suivi d'évolutions reste techniquement lourd à



### Modèles et variations

#### sc1. « Snapshot » sans attribution de la date d'estampillage générale aux données

→ Plusieurs versions de la même base

→ Une date est affectée à chaque instance de données. Rien ne permet d'identifier automatiquement un objet dans la base de données. Il faut le repérer et le suivre visuellement.

#### sc2. « Snapshot » avec attribution de la date d'estampillage générale aux données

→ La date globale de la version est affectée à chacune des instances. Rien ne permet d'identifier automatiquement un objet dans la base de données. Il faut le repérer et le suivre visuellement.

#### sc3. « Snapshot » avec signatures

→ La signature permet de distinguer en fait particulier et dans d'identifier ceux qui sont identiques d'une version à l'autre.

#### bt1. « Time stamp »

→ Une seule base qui réunit toutes les données

→ Les données font partie de la même base. Les données sont « actives », « supprimées » permettant de connaître la période de validité d'un état d'objet.

#### bt2. « Time stamp » avec identificateur

→ L'identificateur permet d'identifier automatiquement les différents états d'un même objet dans le temps. Les actives dates de « création », « suppression » permettant de connaître leur période de validité. Et de reconnaître la suite des états.

### Identification des changements Reconstitution de la série des états

#### sc1. « Snapshot » sans attribution de la date d'estampillage générale aux données

→ Plusieurs versions de la même base

\* La « date des états d'un 'objet' » ne peut pas être reconnue de façon automatique. Les données sont indépendantes les unes des autres.  
 \* On ne peut pas saisir les changements.  
 \* Si on lit les lignes toutes les dates des données dans une même base, on ne distingue pas de quelle version ces données sont issues.  
 \* Ce mode n'a pas intérêt que pour une observation visuelle éphémère ou pour des applications où la dimension temporelle n'importe pas.

#### sc2. « Snapshot » avec attribution de la date d'estampillage générale aux données

\* La « date des états d'un 'objet' » ne peut pas être reconnue de façon automatique. Les données sont indépendantes les unes des autres.  
 \* Pour une date correspond à une version on a facilement l'état de la base à cette date en « stockant les 'objets' par l'attribut : 'date d'estampillage' ».

#### sc3. « Snapshot » avec signatures

\* La « date des états d'un 'objet' » ne peut pas être reconnue de façon automatique (car d'une façon que l'objet est modifié, il change de signature). On ne distingue pas les dates de « création/suppression des données modifiées ». On identifie simplement les 'objets' qui n'ont pas changé entre deux versions: ce sont ceux dont la signature se retrouve parmi les « signatures des deux versions ».

#### bt1. « Time stamp »

\* La « date des états d'un 'objet' » ne peut pas être reconnue de façon automatique, car chaque état d'objet correspond à une date de indépendance.  
 \* Pour une date de lire, on identifie les nouveaux objets: ce sont ceux dont la « date de création » est d'époque les objets d'époque. Il faut que « date de suppression » soit égale à 0. On reconnaît, on ne distingue pas les objets créés/supprimés des objets modifiés.  
 \* Pour un état d'objet final, on accède directement à sa période de validité par rapport au couple: (date de création; date de suppression).

#### bt2. « Time stamp » avec identificateur

\* La série des états d'un 'objet' peut être reconnue de façon automatique, en additionnant à l'identificateur existant, les 'objets' par ordre croissant de 'date de création'.  
 \* Les 'objets' créés sont ceux pour lesquels il n'existe pas d'objet ayant à la fois le même identificateur et une « date de création » antérieure à la leur (les 'objets' supprimés sont ceux pour lesquels il n'existe pas d'objet ayant à la fois le même identificateur et une « date de suppression » postérieure à la leur).  
 \* Pour un état d'objet final, on accède directement à sa période de validité par rapport au couple: (date de création; date de suppression).

### Modèles et variations

#### cr1. Versionnement d'objet sans description totale

→ Une seule base avec différentes versions d'objet

Identifiant	Type	Version	Création	Suppression
Id <sub>A</sub> 1	1	1	d <sub>1</sub>	d <sub>1</sub>
Id <sub>A</sub> 2	1	2	d <sub>2</sub>	d <sub>2</sub>
Id <sub>A</sub> 3	1	3	d <sub>3</sub>	d <sub>3</sub>
Id <sub>A</sub> 1	1	1	d <sub>1</sub>	∅

→ A chaque état d'objet sont associés: l'identificateur de l'objet représenté et un numéro de version. Pour établir la suite des états, il faut retrouver les versions successives ainsi que la valeur des attributs qui s'est pas été modifiés. Les attributs dans « création », « suppression » permettant de connaître la période de validité de cet état.

#### cr2. Versionnement d'objet avec description totale

→ A chaque état d'objet sont associés: l'identificateur de l'objet représenté et un numéro de version. Pour établir la suite des états, il faut retrouver les versions successives. Les attributs dans « création », « suppression » permettant de connaître la période de validité de cet état.

#### dt1. Historique

→ Une seule base qui réunit toutes les données

Identifiant	Version	Création	Date de création	Date de suppression	Id. successives
Id <sub>A</sub>	1	d <sub>1</sub>	∅	d <sub>1</sub>	Id <sub>A</sub>
Id <sub>A</sub>	2	d <sub>2</sub>	∅	d <sub>2</sub>	Id <sub>A</sub>
Id <sub>A</sub>	3	d <sub>3</sub>	∅	d <sub>3</sub>	Id <sub>A</sub>
Id <sub>A</sub>	1	d <sub>1</sub>	∅	∅	-

→ Chaque état d'objet a sa propre signature. La suite entre les différents états doit être reconstruite par rapport au l'attribut « Id. successives ». Les attributs dans « création », « suppression » permettant de connaître la période de validité de cet état.

#### dt2. Historique et informations différencielles

→ Chaque état d'objet a sa propre signature. La suite entre les différents états doit être reconstruite par rapport au l'attribut « Id. successives ». L'attribut « type de modification » explique le changement réel par l'état de l'objet. Les attributs dans « création », « suppression » permettant de connaître la période de validité de cet état.

### Identification des changements Reconstitution de la série des états

#### cr1. Versionnement d'objet sans description totale

\* Une « date des états partiels d'un 'objet' » peut être reconnue de façon automatique, on s'abandonne à l'identificateur existant, les 'objets' par ordre croissant de « numéro de version ». La description de chaque état ne comporte que les valeurs de l'attribut qui est été modifié.  
 \* Les 'objets' créés sont ceux dont le numéro de version est 1. Les 'objets' détruits sont ceux dont la « date de suppression » est non vide et le « numéro de version » est maximal.  
 \* Pour un état d'objet final, on accède directement à sa période de validité par rapport au couple: (date de création; date de suppression).  
 \* A chaque changement de version l'objet subi un changement (ou une destruction), pour connaître le nom de l'attribut de l'objet composant et attributs concernés.

#### cr2. Versionnement d'objet avec description totale

\* La « date des états d'un 'objet' » peut être reconnue de façon automatique, on s'abandonne à l'identificateur existant, les 'objets' par ordre croissant de « numéro de version ».  
 \* Les 'objets' créés sont ceux dont le numéro de version est 1. Les 'objets' détruits sont ceux dont la « date de suppression » est non vide et le « numéro de version » est maximal.  
 \* Pour un état d'objet final, on accède directement à sa période de validité par rapport au couple: (date de création; date de suppression).  
 \* A chaque changement de version l'objet subi un changement (ou une destruction), mais il n'y a pas de moyen automatique pour préciser la nature.

#### dt1. Historique

\* La « date des états d'un 'objet' » peut être reconnue de façon automatique, en fonctionnant de proche en proche la « date des états successifs à partir de l'attribut « Id. successives ».  
 \* Les 'objets' créés sont ceux dont l'identifiant n'est pas une valeur prise par l'attribut « Id. successives ». Les 'objets' détruits sont ceux dont la « date de suppression » est non vide.  
 \* Pour un état d'objet final, on accède directement à sa période de validité par rapport au couple: (date de création; date de suppression).  
 \* Chaque fois que la « date de modification » est non nulle il y a un changement (révision et/ou destruction), mais il n'y a pas de moyen automatique pour en préciser la nature.

#### dt2. Historique et informations différencielles

\* La « date des états d'un 'objet' » peut être reconnue de façon automatique, en fonctionnant de proche en proche la « date des états successifs à partir de l'attribut « Id. successives ».  
 \* Les 'objets' créés sont ceux dont l'identifiant n'est pas une valeur prise par l'attribut « Id. successives ». Les 'objets' détruits sont ceux dont la « date de suppression » est non vide.  
 \* Pour un état d'objet final, on accède directement à sa période de validité par rapport au couple: (date de création; date de suppression).  
 \* Chaque fois que la « date de modification » est non nulle il y a un changement (révision et/ou destruction), pour en préciser la nature, il suffit de regarder la valeur de l'attribut « type de modification ».



gérer. Par ailleurs, il est délicat à mettre en œuvre par des producteurs de données qui ne sont pas les utilisateurs finaux de ces données. En effet, l'interprétation des évolutions d'un objet géographique est souvent thématique ; suivant le contexte, les évolutions seront perçues différemment. Ainsi pour un architecte, un bâtiment de bureaux transformé en habitations a évolué en subissant des modifications, tandis que pour le service qui recense les entreprises, il a disparu. Parfois, même dans un contexte thématique identique, il est difficile de déterminer si l'objet est modifié ou s'il a disparu. (Par exemple dans le phénomène de mitage où dans un mouvement "sournois" des habitations s'installent petit à petit dans un espace vert : quand dira-t-on qu'il évolue et à partir de quand dira-t-on qu'il a disparu au profit d'une zone d'habitat ?). Ces questions sur le maintien de "l'identité" des objets au cours de leur évolution (et sous-jacent de leur identifiant dans la base) rend difficile toute interprétation des évolutions par un autre acteur que l'utilisateur-thématicien.

## Extraction des évolutions dans les SIG : méthodes et outils

Nous avons vu plusieurs modèles pour intégrer les informations de mises à jour. Certains proposent des moyens pour reconstituer automatiquement la série temporelle des états d'un "objet géographique" ; d'autres se limitent à proposer un ensemble de données indépendantes. Certains considèrent les mises à jour comme les étapes d'un processus d'évolution ; d'autres les intègrent comme des opérations autonomes dont ils ne retiennent que le résultat final, parfois de façon globale (au niveau d'un jeu de données et pas de chaque donnée). Il est important de remarquer qu'aucun de ces modèles ne propose d'informations explicites sur les changements intervenus sur les données. Ces modèles qui intègrent la dimension temporelle fournissent uniquement un lien pour relier les états avant/après. Ils ne permettent pas de décrire le chan-

gement intervenu entre les deux. Des outils et des méthodes issus de développements informatiques spécifiques existent toutefois, qui peuvent venir compléter les modèles d'intégration des mises à jour en fournissant des éléments directs sur les changements opérés sur les données. Nous en présentons trois, du plus standard au plus compliqué à mettre en œuvre :

- la clef MD5
  - la journalisation
  - les méthodes et outils pour l'appariement et les calculs de différentiels.
- Il faut noter ici que l'intégration de ces outils et méthodes ne permet toutefois pas de modéliser directement les faits temporels que sont les événements : ces méthodes donnent des détails sur les modifications opérées entre deux états de données, mais elles ne gèrent pas d'information directe sur la nature de l'événement à l'origine du changement. En particulier, même en complétant les modèles de gestion de la dimension temporelle par des outils d'extraction des changements, l'utilisateur ne sera pas en mesure de savoir : pourquoi les objets géographiques qu'il observe ont changé, qui (ou quoi) est à l'origine de ce changement, ou toute autre information détaillée sur cette modification.

### La clef MD5

Il s'agit d'un outil informatique standard, une sorte "d'empreinte digitale" pour les messages informatiques. Il permet d'identifier rapidement et simplement les données qui ont changé, même si elles n'ont subi qu'une modification minime. En revanche, cet outil ne donne aucun détail sur la nature des changements. Il utilise l'algorithme MD5 qui permet de produire une clef de 128 bits à partir d'un message de taille quelconque. Cet algorithme permet de vérifier l'intégrité des données d'un message de façon beaucoup plus sûre que le classique contrôle de parité.

Dans le cadre du suivi d'évolutions au sein de bases de données géographiques, la clef MD5 peut être utilisée lorsque le modèle de référence est un modèle par snapshots basique sans signature. Elle permet d'identifier simplement et automatiquement les

données restées identiques entre deux versions.

### La journalisation

La journalisation consiste à enregistrer au fur et à mesure l'ensemble des commandes informatiques qui sont effectuées sur une base pour en modifier des données. Partant de l'état initial de cette base, il suffit de lancer la suite des commandes enregistrées dans le journal pour reproduire la suite des opérations qu'elle a subie et obtenir ainsi au final l'état de la base le plus récent. Le journal donne un accès direct à tous les changements opérés sur la base, mais il reste factuel : il les décrit mais ne donne pas leur cause. Par exemple il peut révéler qu'un "bâtiment" a été supprimé, mais on ne sait pas si le bâtiment participe à un complexe plus large suite à une opération de rénovation ou s'il a été effectivement détruit sur le terrain.

Par ailleurs, le journal ne peut apporter de solution effective que si la base initiale considérée correspond exactement à la base à l'origine du journal. Ainsi, si un utilisateur intervient sur les données géographiques d'un producteur autre, en les modifiant pour intégrer ses propres données, il interrompt la chaîne logique du journal.

### Les méthodes et outils pour l'appariement et les calculs de différentiels

L'appariement consiste à "reconnaître" dans deux jeux de données les couples d'objets servant à représenter un même objet géographique dans chacun des jeux. Appliqué à deux bases de données différentes, cela permet d'établir des relations entre leurs représentations pour faire par exemple de la multi-représentation dans un SIG. Utilisé sur deux jeux d'une même base de données, mais dans des versions différentes, l'appariement permet de sélectionner les objets qui n'ont pas la même représentation dans les deux jeux de données et d'identifier ainsi ceux qui ont changé. Pour pouvoir comparer deux données, les outils d'appariement utilisent des outils de calculs de différentiels. Il s'agit d'explicitier la différence existant entre les deux et de la qualifier. Si ces données représentent deux états différents d'un même



► “objet”, le calcul du différentiel caractérise alors la différence existant entre l'état-avant et l'état-après. Il sert ainsi à obtenir des éléments explicites sur les changements intervenus sur les données. Les méthodes et outils pour l'appariement et les calculs différentiels sont aujourd'hui développés principalement par les producteurs de bases de données, comme éléments de processus de production. Ces techniques restent complexes à mettre en œuvre et beaucoup sont de l'ordre de la recherche. De fait, elles sont encore peu accessibles aux utilisateurs.

## L'apport des métadonnées

Les modèles pour implémenter les mises à jour ne modélisent pas explicitement les événements : il n'y a pas d'information pour les représenter en tant que tels. Les techniques de calculs différentiels et d'appariements peuvent servir à déterminer les changements opérés entre deux états d'une même donnée, mais ils ne saisissent que des informations directement déductibles des données. La structure définie par le modèle ne permet en revanche pas d'intégrer d'autres informations caractéristiques (par exemple, la cause du changement : technique (ex. : correction d'erreur, changement de spécifications) ou thématique (ex. : destruction d'un tronçon de route liée à la réalisation d'un rond-point, construction d'un tronçon pour desservir une future zone industrielle), leur durée réelle (la période de temps séparant deux versions de la base peut être d'un an alors que le changement s'est effectué sur un mois, sur deux

ans) etc. L'introduction de métadonnées peut pallier ce manque.

Les métadonnées, qualifiées souvent de “données sur les données” regroupent, généralement sous forme d'un fichier complémentaire, des données contenant des informations utiles à la description de la base de données ou des données qu'elle contient. Il existe plusieurs niveaux de métadonnées. Au niveau détaillé, on peut introduire des informations sur les changements effectués sur les données pour préciser leurs causes ou leur durée mais aussi qui a collecté l'information ou qui l'a introduit dans la base de données, etc.

## Vers l'étude de phénomènes géographiques avec les SIG

Les modèles d'implémentation des mises à jour, les outils et méthodes pour reconstituer des informations sur les changements et les métadonnées constituent un ensemble de moyens complémentaires pour mieux gérer les évolutions d'une base de données. Introduits dès la conception des bases, ils permettent de saisir des informations sur la dimension temporelle, plus nombreuses et plus élaborées. Leur mise en œuvre associée augmente les capacités des outils SIG à intégrer le temps dans les analyses et à le traiter de façon automatique. Or, il s'agit bien aujourd'hui pour les SIG de développer ces capacités. En effet, l'observation des phénomènes géographiques et plus encore leur compréhension impliquent de pouvoir suivre leurs mouvements et d'obtenir des informations sur la façon dont ils se produisent. Pour devenir de véritables outils d'aide à la

décision, les SIG doivent aller au-delà de la mise en observation de données géographiques. Ils doivent mettre en évidence des variations, participer aux simulations, montrer des effets.... Cela passe d'abord par l'intégration de la dimension temporelle, puis par le développement de méthodes d'analyses. Alors, et sous ces conditions, les SIG devraient devenir des outils privilégiés pour tous ceux qui tentent de comprendre, d'anticiper et de maîtriser des phénomènes géographiques. ●

## Bibliographie

**Bordin P, 2006**, *Méthode d'observation multi-niveaux pour le suivi de phénomènes géographiques avec un SIG*, thèse de doctorat en Sciences de l'information géographique, Université de Marne-la-Vallée, 12/12/2006

**Peuquet D, 2002** *Representations of space and time*, Guilford, New-York, 2002

**Langran G 1993** *Time in geographic information systems* Taylor et Francis, London, 1993

## Contact

**Patricia BORDIN**

Laboratoire de Géomatique Appliquée,  
ENSG Marne-la-Vallée

patricia.bordin@ensg.eu

## ABSTRACT

*GIS, which are the systems dedicated to the management and treatment of geographical data, have not yet become real tools in decision making. Indeed, to help follow, analyze and understand the phenomena that take place on the ground, it is necessary to study at the same time their geographical and temporal aspects. Now, partly because they were originally dedicated to map making, but also all because of the technical difficulties involved in temporal modelling, GIS often lead to rough representations of evolutions. Yet, there are solutions to improve their performances. In the following article, we will see how some can be efficient a posteriori and others during the modelling process, when data are gathered.*

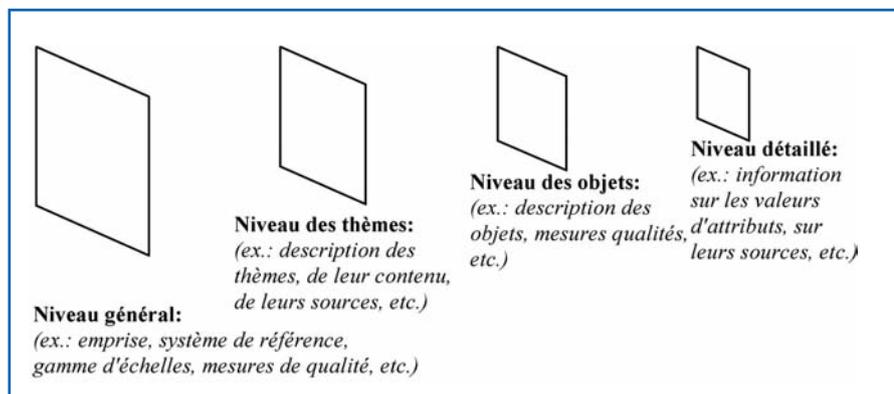


Figure 4. Différents niveaux de métadonnées