

# Modélisation Spatio-temporelle des Connaissances d'un Système d'Information Géographique

A. Mokrane<sup>1</sup>, O. Laouamri<sup>2</sup>, G. Dray<sup>1</sup>, P. Poncelet<sup>1</sup>

<sup>1</sup>*Groupe Connaissance - LGI2P - EMA  
Parc scientifique G.Besse Nîmes 30000, France*

{[abdenour.mokrane](mailto:abdenour.mokrane@ema.fr), [gerard.dray](mailto:gerard.dray@ema.fr), [pascal.poncelet](mailto:pascal.poncelet@ema.fr)}@ema.fr

<sup>2</sup>*LCIS - INPG - ESISAR*

*50, rue Barthélemy de Laffemas Valence 26902, France*

[oussama.laouamri@esisar.inpg.fr](mailto:oussama.laouamri@esisar.inpg.fr)

**Résumé**—L'objectif principal d'un SIG (Système d'Information Géographique) est la représentation de diverses connaissances géographiques et le raisonnement à partir de ces connaissances dans des domaines d'application très variés. Les modèles logiques actuels représentent la spatialité et la temporalité de l'information géographique par des modèles de BD (Bases de Données) spatio-temporelles. Ces modèles gèrent efficacement les informations géographiques spatio-temporelles, cependant ils ne permettent pas le raisonnement ou la prise de décision à partir des connaissances géographiques spatio-temporelles. En effet, ces modèles ne comportent pas de connaissances explicites et les connaissances ainsi que les règles d'inférence sont enfouies dans les bases de données. AROM (Allier Relations et Objets pour Modéliser) est un des systèmes de représentation de connaissances par objets. L'objectif de cet article est d'étendre ce système pour une gestion avancée des connaissances d'un SIG, en lui intégrant deux couches : spatiale et temporelle, ainsi qu'une série d'opérateurs permettant de formuler des requêtes spatiales, temporelles et spatio-temporelles. Nous illustrons notre modèle d'AROM étendu sur l'application du cadastre.

**Mots Clés:** — Modélisation spatio-temporelle, Système d'Information Géographique, AROM étendu.

## 1 INTRODUCTION

Un Système d'Information Géographique (SIG) permet le stockage, l'interrogation, la manipulation, le partage, la diffusion et la restitution d'informations à composantes géographiques [1], telle que des parcelles ou des rues. Ces informations géographiques se caractérisent par leurs aspects multidimensionnels. Elles sont définies par trois types de données : des données thématiques décrivant un objet géographique, des données spatiales exprimant la géométrie, la localisation et les relations topologiques ainsi que des données temporelles exprimant la dynamique et l'évolution des deux premières dimensions : thématique et spatiale [1,2,4]. Cette dualité de dimensions entraîne une gestion complexe de données tridimensionnelles, au niveau de la modélisation conceptuel, logique et physique. La modélisation au

niveau conceptuel de l'information géographique consiste à modéliser l'information indépendamment de toute implantation. A ce niveau on modélise le discours du monde réel sans se soucier de sa mise en œuvre. Au niveau de la modélisation logique, l'information est représentée par le modèle logique des Systèmes de Gestion de Bases de Données (SGBD). Enfin, on s'intéresse au niveau de la modélisation physique à la représentation de l'information sur le support physique de stockage, d'acquisition ou de visualisation.

Nous nous intéressons en particulier au niveau de la modélisation logique de l'information géographique à référence spatial et temporel, dans le but de modéliser les évolutions du monde réel et de traiter des données spatio-temporelles. Les modèles logiques actuels [1,3,4,5,6,7] représentent la

spatialité et la temporalité de l'information géographique par des modèles de bases de données spatio-temporelles. Ces derniers sont généralement implémentés par des bases de données de type relationnel, relationnel étendu, objet, objet-relationnel ou hybrides (Bases de Données-XML), avec une intégration de trois couches spatiale, temporelle et spatio-temporelle. Ces types de modèles ont été concrétisés par divers environnements de gestion de bases de données spatio-temporelles, tel que GODOT, ONTOS ou ROSE algebra [3,4,5,7,8]. Ces modèles gèrent efficacement les informations géographiques spatio-temporelles, cependant ils ne permettent pas une représentation explicite des connaissances géographiques spatio-temporelles et le raisonnement ou la prise de décision à partir de ces connaissances. En effet, ces modèles ne comportent pas de connaissances explicites et les connaissances ainsi que les règles d'inférence sont enfouies dans les bases de données.

Afin de répondre à cette problématique, nous proposons dans cet article une extension du système de représentation de connaissances par objet *AROM* (*Allier Relations et Objets pour Modéliser*) [9]. A cet effet, nous intégrons au système *AROM* deux couches : spatiale et temporelle, pour une gestion avancée des connaissances d'un SIG ainsi qu'une série d'opérateurs permettant de formuler des requêtes spatiales, temporelles et spatio-temporelles. Cette extension d'*AROM* permet une modélisation objet des connaissances géographiques spatio-temporelles et une formulation de requêtes complexes, afin d'aider la communauté des décideurs (gestionnaires de réseaux de transport, planificateurs de l'urbanisme, chargés de l'aménagement du territoire, etc.) en matière de raisonnement et de prise de décision.

L'article est organisé comme suit. Nous présentons dans la section 2 le système *AROM*. La section 3 et 4 détaillent notre extension spatio-temporelle du système *AROM* et du Langage de Modélisation Algébrique (*LMA*) associé à ce système. Dans la section 5, nous illustrons notre modèle d'*AROM étendu* sur une application du cadastre. Nous terminons dans la section 6, par une conclusion et une perspective d'avenir de notre travail.

## 2 PLATE FORME AROM

*AROM* (*Allier Relations et Objets pour Modéliser*) [9] est un système qui reprend et supporte la plupart des caractéristiques des Systèmes de Représentation de Connaissances par Objets (SRCO). De plus, *AROM* intègre quatre spécificités très importantes, à savoir :

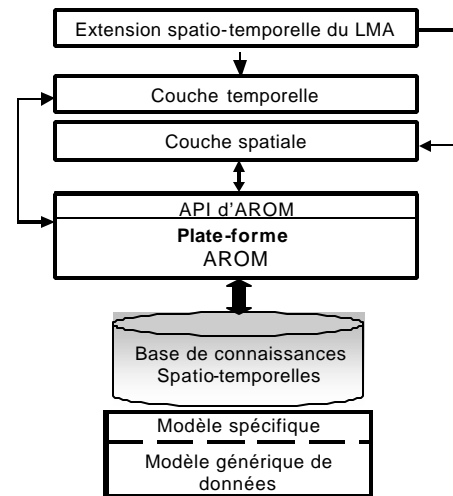


Fig. 1: Architecture d'AROM Spatio-Temporel.

- 1) *AROM* fait une distinction explicite entre les classes et les associations dans la mesure où il considère les associations comme des entités à part entière. Cette distinction permet de traiter les associations au même titre que les classes, le concepteur peut donc les organiser hiérarchiquement par une relation de spécialisation et peut leur associer des contraintes pour le maintien de la cohérence. Ce concept nous aide à développer le modèle objet des connaissances géographiques spatio-temporelles.
- 2) *AROM* supporte le *Langage de Modélisation Algébrique (LMA)*, ce langage permet de renforcer les mécanismes d'inférence classiques. Le *LMA* peut servir à la fois à l'écriture d'équations pour définir un objet du Modèle de Connaissances (MC) relativement aux autres, de contraintes sur les objets afin de maintenir la cohérence du MC et de requêtes pour une navigation par le contenu.
- 3) *AROM* offre également une interface de modélisation graphique qui permet au concepteur de représenter le MC d'une manière similaire à *UML*. Par le biais de cette interface, le concepteur ou l'utilisateur a la possibilité d'instancier à tout moment son MC d'une manière graphique.
- 4) Enfin, *AROM* complète ses possibilités d'utilisation en offrant une *API Java* qui consiste en un ensemble de packages permettant à l'utilisateur d'exploiter le MC par programmation. Cette dernière possibilité est très avantageuse car elle permet d'exploiter la puissance d'*AROM* à travers le langage de programmation *JAVA* en s'affranchissant des limites de son interface de modélisation

graphique, permettant ainsi d'aboutir à des applications puissantes et souples.

### 3 EXTENSION SPATIO-TEMPORELLE D'AROM

Le modèle de données spatio-temporelles écrit en *AROM* est basé sur deux couches : spatiale et temporelle (cf. Fig. 1). Ce modèle fournit une hiérarchie extensible de classes spatiales et temporelles, à partir desquelles un modèle de connaissances d'une application d'un SIG Spatio-Temporel spécifique peut être établi en utilisant le concept de l'héritage.

La couche spatiale et la couche temporelle représentent le noyau qui concrétise la spatialité et la temporalité du modèle de connaissances et l'extension du *LMA*.

Nous supposons à ce niveau qu'un objet spatio-temporel *O* est tel que  $O = \{ST, SP, T\}$  où *ST* est le composant attributaire de *O* estampillé par le temps ; *SP* est le composant spatial de *O* estampillé par le temps ; et *T* est le composant temporel de *O* représentant son temps valide. La période valide d'un objet est l'intervalle de temps dans lequel sa sémantique est vraie en réalité modélisée.

#### 3.1 Modèle spatial

La spatialité est la faculté de décrire les relations entre les objets et l'espace [10]. L'approche suivie est celle qui permet d'expliciter la géométrie sous forme d'objets qui peuvent être localisés dans

l'espace à partir d'un ensemble d'objets de base : point, ligne et polygone. Nous décrivons cette faculté à l'aide d'une hiérarchie de classes et de généralisation (cf. Fig. 2). A chaque classe de type spatial est associé un ensemble d'attributs représentant la géométrie adéquate (la partie spatiale du modèle de connaissances). Le comportement de chaque classe est implémenté séparément du modèle de connaissances. Il est implémenté au niveau de la couche spatiale sous forme de codes *Java* ajouté à la plate-forme *AROM*. Il s'agit des opérateurs métriques et topologiques (superficie, distance, longueur, disjonction, adjacence, croisement, recouvrement, inclusion, égalité, etc.).

Ce modèle comporte des classes génériques et des classes plus spécifiques. Notamment, il existe trois sortes de classes : les classes qui représentent la géométrie statique (stable), les classes qui représentent la géométrie dynamique (l'introduction de la temporalité au niveau de la géométrie de chaque classe) et les classes qui représentent les systèmes à référence spatiale.

Le premier et le deuxième ensemble de classes ont la même superclasse : la classe *Location* qui représente l'information à référence spatiale. Plus précisément, les informations peuvent être localisées dans l'espace soit par un système à références spatiales (classe *SRS*), soit par coordonnées (classe *Coordinates*), soit par géodésie (classe *Geodesy*) ou d'autres dont la définition est à la charge de l'utilisateur.

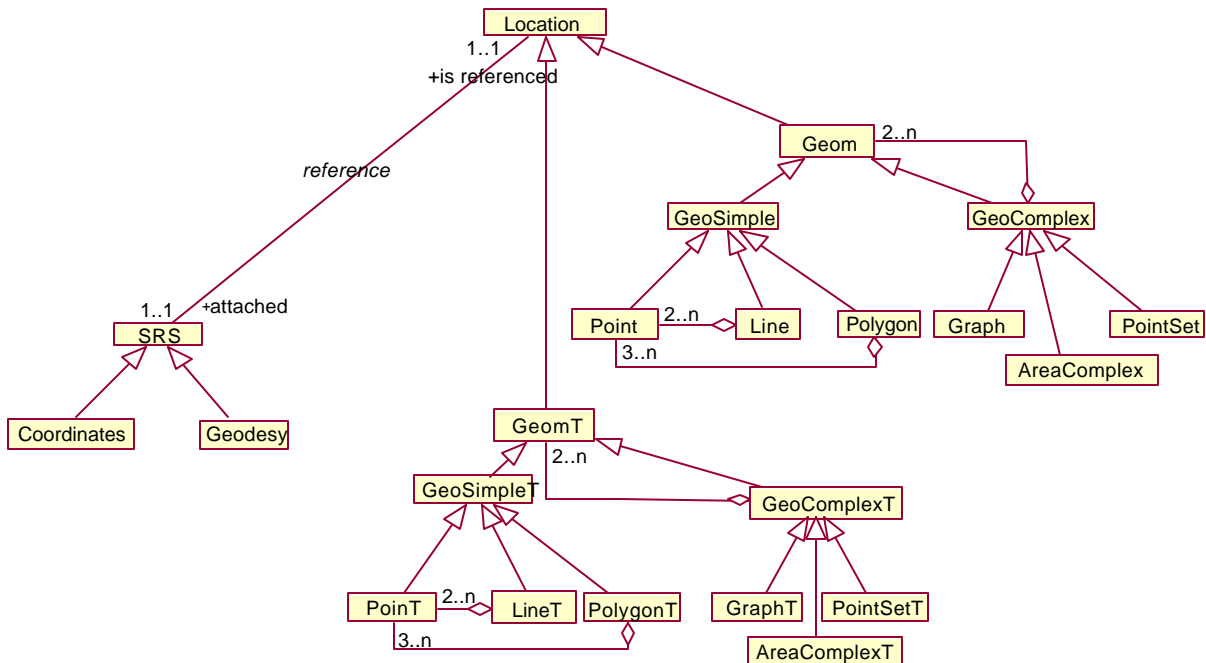


Fig. 2: Modèle spatial.

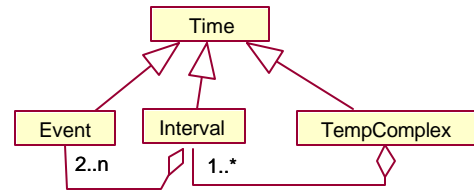
Les classes *GeomT* et *Geom* représentent les types spatiaux sans préciser la géométrie. La première représente la géométrie qui change au cours du temps et la deuxième représente la géométrie stable, i.e. qui ne change pas au cours du temps. La classe *GeoSimple* représente les classes spatiales simples. Les sous-classes de *GeoSimple* sont : la classe *Point* qui représente la géométrie d'un point, la classe *Line* qui représente la géométrie d'un polygone (composée de deux ou plusieurs *Points*) et la classe *Polygon* qui représente la géométrie d'un polygone (composée de trois ou plusieurs *Points*).

La classe *GeoComplex* permet de décrire toute composition de classes spatiales simples ou complexes. Les sous-classes de *GeoComplex* sont : la classe *PointSet* qui est un ensemble de points (exemple : arbres d'une forêt), la classe *Graph* qui est un ensemble de lignes (exemple : réseau routier) et la classe *AreaComplex* qui est un ensemble de surfaces simples (exemple : un ensemble d'îlots). Il en va de même pour l'autre ensemble de classes *GeoSimpleT*, *PointT*, *LineT*, *PolygonT*, *GeoComplexT*, *GraphT*, *PointSetT*, *AreaComplexT*, avec l'ajout de la dynamique de la géométrie (évolution spatiale).

**3.2 Modèle temporel**

Ce modèle (cf. Fig. 3) décrit la granularité temporelle et les types temporels de base. La classe *Time* représente la valeur temporelle qui peut être soit : un instant (classe *Event*), un intervalle de temps (classe *Interval*) ou une liste d'intervalles (classe *TempComplex*). Comme le modèle spatial, le modèle temporel a aussi un comportement implémenté au niveau de la couche temporelle. Il s'agit de tous les opérateurs métriques et

topologiques (granularité, temps valide, durée, intersection temporelle, égalité, inclusion, etc.).

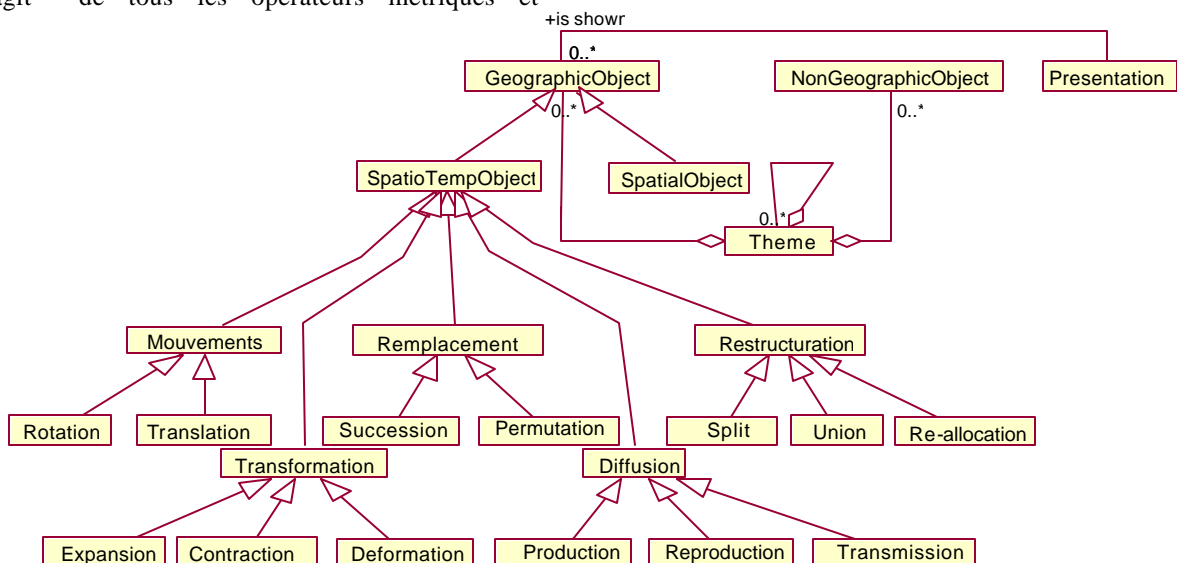


**Fig. 3: modèle temporel.**

**3.3 Modèle géographique spatio-temporel**

Ce modèle décrit l'information géographique multidimensionnelle (cf. Fig. 4). La classe principale est la classe *GeographicObject* qui représente l'entité géographique sans préciser l'orthogonalité choisie (spatiale ou spatio-temporelle). Les classes *SpatialObject* et *SpatioTempObject* sont des sous-classes de la classe *GeographicObject*, la première représente des entités géographiques à référence spatiale (orthogonalité spatiale) et la deuxième représente des entités géographiques spatio-temporelles (orthogonalité spatio-temporelle). La dimension structurelle (attributaire ou thématique) est ajoutée par héritage. La classe *SpatialObject* se compose d'un ou plusieurs objets géométriques (classe *Geom*). La classe *Geom* représente la géométrie de l'entité géographique en question.

La multiplicité de cette agrégation a pour but de concrétiser la possibilité de la représentation multiple (ou alternative) s'il s'agit d'une entité géographique qui a des géométries alternatives ou multiples. La classe *SpatioTempObject* se compose d'un ou plusieurs objets géométriques évolutifs dans le temps (classe *GeomT*) et d'une ou plusieurs valeurs temporelles (classe *Time*).



**Fig. 4: Modèle géographique spatio-temporel.**

Pour enrichir notre modèle, nous spécialisons la classe *SpatioTempObject*. Cela consiste à ajouter des sous-classes qui décrivent les processus d'évolution spatio-temporelle de base. La classe *Presentation* a pour but de représenter les spécificités de présentation (visualisation) cartographique (symboles cartographiques, échelles et couleurs).

La classe *Theme* représente l'entité thème. Cette classe regroupe des entités géographiques (la classe *GeographicObject*), des entités non géographiques (la classe *NonGeographicObject*) et/ou d'autres thèmes.

### 3.4 Démarche de raffinement

Le modèle de connaissances proposé est bien entendu générique et recouvre un ensemble considérable de domaines d'applications. Afin de pouvoir le raffiner (l'adapter) à un modèle spécifique, nous définissons une démarche de raffinement qui prend en considération les concepts d'*AROM*. La démarche consiste à :

- 1) Choisir la classe adéquate selon la dimension de la connaissance à modéliser.
- 2) Associer la spatialité, la temporalité ou la spatio-temporalité aux classes géographiques par héritage de l'association concernée.
- 3) Enrichir les nouvelles classes par les informations (attributs, contraintes, etc.) de la modélisation.

## 4 EXTENSION SPATIO-TEMPORELLE DU LMA

Le *Langage de Modélisation Algébrique (LMA)* d'*AROM* est doté de nombreux opérateurs scalaires (+, \*, sum, etc.) et attributaires (opérateurs de navigation : in, set, etc.). Cependant, il n'est pas adapté pour exprimer spécifiquement des requêtes spatio-temporelles. De ce fait, nous étendons le langage *LMA* existant afin de supporter ce genre de requêtes en lui intégrant une série d'opérateurs. Nous proposons dans les sections suivantes les différents opérateurs spatiaux, temporels et spatio-temporels de base.

### 4.1 Opérateurs spatiaux

En accord avec [1,3,4,8], nous classons les opérateurs spatiaux en trois catégories : opérateurs d'orientation, métriques ou topologiques. Nous considérons seulement les opérateurs exécutables sur les modèles vectoriels puisque notre choix de modèle physique s'est porté vers ce type de modèle. Nous décrivons dans la suite de cette section les opérateurs spatiaux que nous intégrons au langage de requêtes *LMA* d'*AROM*.

#### 4.1.1 Opérateur pour rechercher l'objet spatial

L'opérateur *GetSP* retourne l'objet spatial de son paramètre *A* (objet géographique). L'objet complexe retourné contient la représentation spatiale de *A*. Il est composé d'attributs non-spatiaux, permettant de maintenir des informations spécifiques sur la représentation (comme l'échelle ou le système de mesure) et d'un attribut spatial qui énumère les objets géométriques décrivant la géométrie de *A*.

#### 4.1.2 Opérateurs d'orientation

Les opérateurs d'orientation vérifient s'il existe un rapport spécifique d'orientation entre deux ensembles d'objets géométriques.

- *North(A,B)*: il retourne vrai si tous les éléments de *A* ont une relation du *NORD* avec tous les éléments de *B*,
- *East(A,B)*: même chose qu'avec l'opérateur *North* mais dans la direction *EST*.

#### 4.1.3 Opérateurs métriques

Les opérateurs métriques retournent une grandeur scalaire qui représente une propriété intrinsèque des objets analysés. Nous considérons les opérateurs suivants:

- *Area(A)*: calcule la zone occupée par l'objet *A* (polygone).
- *Length(A)*: il retourne la longueur de *A* (polyline).
- *Perimeter(A)*: il retourne le périmètre de *A* (des polygones).
- *Distance(A,B)*: il calcule la distance entre deux ensembles d'objets géométriques (points, lignes, ou polygones).

#### 4.1.4 Opérateurs topologiques

Les opérateurs topologiques de base sont la disjonction, l'adjacence, le croisement, le recouvrement, l'inclusion et l'égalité [1,3,11].

- *Disjoint(A,B)*: il retourne vrai si tous les éléments de *A* ont une relation de disjonction avec tous les éléments de *B* ; sinon retourne faux.
- *Touch(A,B), Overlap(A,B), Cross(A,B)* : ces opérateurs retournent vrai s'il existe une paire (*a*, *b*) de *A* et *B* ayant respectivement une relation d'adjacence, de recouvrement ou de croisement (intersection); sinon retournent faux.
- *Inside(A,B)*: cet opérateur caractérise l'inclusion, il retourne vrai si *A* est inclus dans *B*.
- *Equal(A,B)*: il retourne vrai si *A* et *B* ont la même géométrie.

## 4.2 Opérateurs temporels

### 4.2.1 Opérateurs métriques

- $GetT(A)$ : retourne l'objet temporel de  $A$ .
- $Begin(A), End(A)$ : ils retournent respectivement le temps de départ et de fin de  $A$ . Ces opérateurs concrétisent le temps valide, la temporalité de  $A$  dans ce cas est un intervalle.
- $Day(A), Month(A)$  et  $Year(A)$ : ces opérateurs retournent respectivement le jour, le mois et l'année correspondants à  $A$ .
- $When(A)$ : retourne le temps valide de  $A$ .
- $Interval(s,e)$ : il retourne la durée de l'intervalle  $[s..e]$ .
- $IntersT(A,B)$ : il retourne la valeur temporelle correspondant à l'intersection entre  $A$  et  $B$ .

### 4.2.2 Opérateurs topologiques [12,13]

$Ts$  ( $\cdot$ ),  $Te$  ( $\cdot$ ) dénotent respectivement le temps de départ et de fin de l'opérande.

- $BeforeT(A,B)$ : il retourne vrai si  $Te(A) < Ts(B)$ , sinon retourne faux.
- $OverlapsT(A,B)$ : il retourne vrai si  $\exists t (t \in A \wedge t \in B)$ , sinon retourne faux.
- $EqualT(A,B)$ : retourne vrai si  $A=B$ , sinon retourne faux.
- $Contains(A,B)$ : il retourne vrai si  $\forall t \in B, t \in A$ , sinon retourne faux.
- $Meets(A,B)$ : il retourne vrai si  $Ts(B) - Te(A) = I$  (où  $I$  représente un instant), sinon retourne faux.

## 4.3 Opérateurs spatio-temporels

Ces opérateurs étendent les opérateurs spatiaux définis dans la section 4.1 en ajoutant la dimension temporelle.

### 4.3.1 Opérateurs comportementaux

A ce niveau, nous nous intéressons aux changements d'états de l'objet spatio-temporel.

- $GetVersion(A,T)$ : il retourne les versions (états) valides de l'objet  $A$  à l'instant ou l'intervalle  $T$ . Notamment, il retourne une liste de tuples  $\{ST_i, SPI_i, TI_i\} / \forall TI_i, TI_i \in T$ . Cet opérateur concrétise l'aspect évolutif de l'objet géographique. Le type d'évolution se définit au niveau de la définition du schéma de données par héritage (processus spatio-temporels [13]).
- $IsState(A,T,state)$ : il retourne vrai si l'état passé en paramètre est l'état de  $A$  à l'instant ou l'intervalle  $T$ , sinon il retourne faux. Les états possibles sont : état statique (stable), état changeant, état suspendu, etc [2]. Le paramètre  $state$  est donc de type énuméré, il décrit les états de changement.

Par exemple,  $IsState(Voiture, TI, STATIC)$

correspond à la question « est-ce que l'entité géographique *Voiture* est en état stable ou non à l'instant  $TI$  ? ».

### 4.3.2 Opérateurs métriques spatio-temporels

Ces opérateurs ont comme paramètre un ou deux objets spatio-temporels et une valeur temporelle :

- $AreaST(A,T)$ : il calcule la zone valide occupée par l'objet  $A$  à l'instant  $T$ . Notamment, il retourne la paire {superficie, temps}.
- $LengthST(A,T)$ : il retourne la longueur valide de  $A$  à l'instant  $T$ . notamment, il retourne la paire {longueur, temps}.
- $PerimeterST(A,T)$ : il retourne le périmètre valide de  $A$  à l'instant  $T$ . Notamment, il retourne la paire {périmètre, temps}.
- $DistanceST(A,B,T)$ : il calcule la distance valide entre deux ensembles d'objets géométriques pour tous les intervalles d'intersection temporelle entre  $A$ ,  $B$  et  $T$ .

### 4.3.3 Opérateurs topologiques spatio-temporels

$DisjointST$ ,  $TouchST$ ,  $OverlapST$ ,  $CrossST$ ,  $InsideST$  et  $EqualST$ : ces opérateurs retournent vrai si la relation topologique en question  $Disjoint$ ,  $Touch$ ,  $Overlap$ ,  $Cross$ ,  $Inside$  ou  $Equal$  (cf. section 4.1.4) entre deux objets  $A$  et  $B$  (passés en paramètres) est constante pendant un intervalle de temps  $T$  (passé en paramètre) sinon retournent faux. Donc, ils indiquent si les géométries des deux objets  $A$  et  $B$  sont relativement constantes pendant l'intervalle de temps  $T$ .

## 4.4 Autre opérateur

$IsKindOF(A,type)$ : il retourne vrai si l'objet  $A$  est de même type que le paramètre passé en argument, sinon retourne faux. Par exemple,  $IsKindOF(A, "Polygon")$  est équivalent à « est-ce que l'objet  $A$  est un polygone ? ».

## 4.5 Requête spatio-temporelle en LMA

Une requête exprimée en *LMA* correspond à une expression formée à partir des opérateurs définis dans la section 4. Nous considérons les trois types de requêtes suivantes : requêtes spatiales, requêtes temporelles et requêtes spatio-temporelles. Les requêtes spatiales traitent les relations spatiales entre les objets et elles sont formulées à l'aide d'opérateurs spatiaux. Elles retournent des informations calculées ou des objets spatiaux. Les requêtes temporelles contiennent au moins un opérateur temporel mais aucun opérateur spatial. Elles traitent des relations temporelles entre les objets et les temps valides des objets. Les requêtes spatio-temporelles traitent des relations spatiales au cours du temps. Elles retournent des objets spatiaux, des valeurs métriques valides à un instant spécifique

ou le temps valide des relations spatio-temporelles.

Dans la section suivante, nous illustrons notre modèle et l'intérêt de nos opérateurs à travers des exemples de requêtes s'appuyant sur l'extension du *LMA*.

## 5 ILLUSTRATION : APPLICATION DU CADASTRE

### 5.1 Enoncé du problème

Considérons un ensemble de parcelles qui doivent être équipées en conduites d'eau, ainsi que des routes qui coupent ces parcelles.

### 5.2 Modèle objet des connaissances

Initialement, la modélisation associe trois classes *AROM* aux données géographiques : la classe *Parcelles*, la classe *Conduites* et la classe *Routes*. Nous proposons que les conduites et les routes soient représentées par des polygones et les parcelles par des polygones. Les conduites sont extensibles au cours du temps. La figure 5 illustre le modèle objet des connaissances écrit en *AROM étendu*, associé à cette application du Cadastre. Le modèle générique de connaissances étant placé au-dessus du trait gras. Le modèle placé au-dessous du trait gras correspond au modèle obtenu après spécialisation (raffinement). La notation est celle d'*AROM* dont les rectangles représentent les classes, les rectangles arrondi représentent les classes d'associations, les flèches représentent les liens de l'héritage et les lignes représentent les rôles avec leurs cardinalités. La spatialité est attachée à la géométrie adéquate à travers la spécialisation des associations d'agrégations. Par exemple, l'association générique *Component\_of\_S* est spécialisée en deux nouvelles associations spécifiques, *Component\_of\_S\_Routes* pour attacher la géométrie linéaire à la classe *Routes*, *Component\_of\_Parcelles* pour attacher la géométrie polygonale à la classe *Parcelles*. Comme la classe *Conduites* est une entité géographique spatio-temporelle qui s'étend au cours du temps, nous l'héritons de la classe *Expansion* (extension) et nous lui associons la spatialité par la spécialisation de l'association *Component\_of\_ST*.

La temporalité est associée à la valeur temporelle adéquate (selon la granularité choisie) à travers la spécialisation de l'association *Component\_of\_Temp*. Par exemple, la classe *Conduites* est associée à la valeur temporelle *Event* (instant) par le biais de l'association *Component\_of\_Temp\_Conduites*. Nous pouvons expliciter au niveau du schéma la topologie spatiale et temporelle ou les relations topologiques par des associations, mais cela va surcharger le schéma et le développeur par l'ajout de la

sémantique nécessaire. A cet effet, nous concrétisons ces relations sous forme d'opérateurs spatiaux, temporels et spatio-temporels. Ces opérateurs se basent sur des algorithmes qui manipulent directement le modèle physique (coordonnées, valeurs métriques, etc.).

### 5.3 Exemples de requêtes

#### 5.3.1 Exemple de requête spatiale

« Quelles sont les conduites d'eaux de la parcelle A ? » (requête exprimée en langage naturel).  
 « set (c in Conduites : Inside (GetSP(c), GetSP(A)) = true) » (requête exprimée en *AROM étendu*).

#### 5.3.2 Exemple de requête temporelle

« Quelles sont les conduites d'eau qui ont été construites en 1980 ? » (requête exprimée en langage naturel).  
 « set (c in Conduites : OverlapsT (GetT(c), 1980) = true) » (requête exprimée en *AROM étendu*).

#### 5.3.3 Requête spatio-temporelle

« Quelles sont les parcelles dont les conduites d'eau ont été construites entre 01/01/50 à 01/01/80 ? » (requête exprimée en langage naturel).  
 « set (p in Parcelles : c in Conduites : InsideST (c, p, Interval(01/01/50, 01/01/80))=true) » (requête exprimée en *AROM étendu*).

## 6 CONCLUSION

La modélisation logique des connaissances géographiques spatio-temporelles est un niveau difficile de modélisation. Fournir un modèle générique de connaissances géographiques spatio-temporelles extensible et un langage de requêtes spatio-temporelles riche va offrir une aide considérable pour accomplir la tâche de modélisation à ce niveau. Nous avons proposé dans cet article un modèle générique de représentation et de gestion de connaissances géographiques spatio-temporelles. Nous avons concrétisé ce modèle par l'extension du système de représentation de connaissances par objet *AROM*, en lui ajoutant deux couches : spatiale et temporelle, ainsi qu'une série d'opérateurs permettant de formuler algébriquement des requêtes spatio-temporelles. L'illustration de notre modèle d'*AROM étendu* sur l'application du cadastre, nous confirme la capacité de ce modèle pour supporter et représenter les connaissances géographiques spatio-temporelles. Une perspective d'avenir de notre travail consiste à développer un langage visuel de pictogrammes adapté à la notation d'*AROM étendu*.



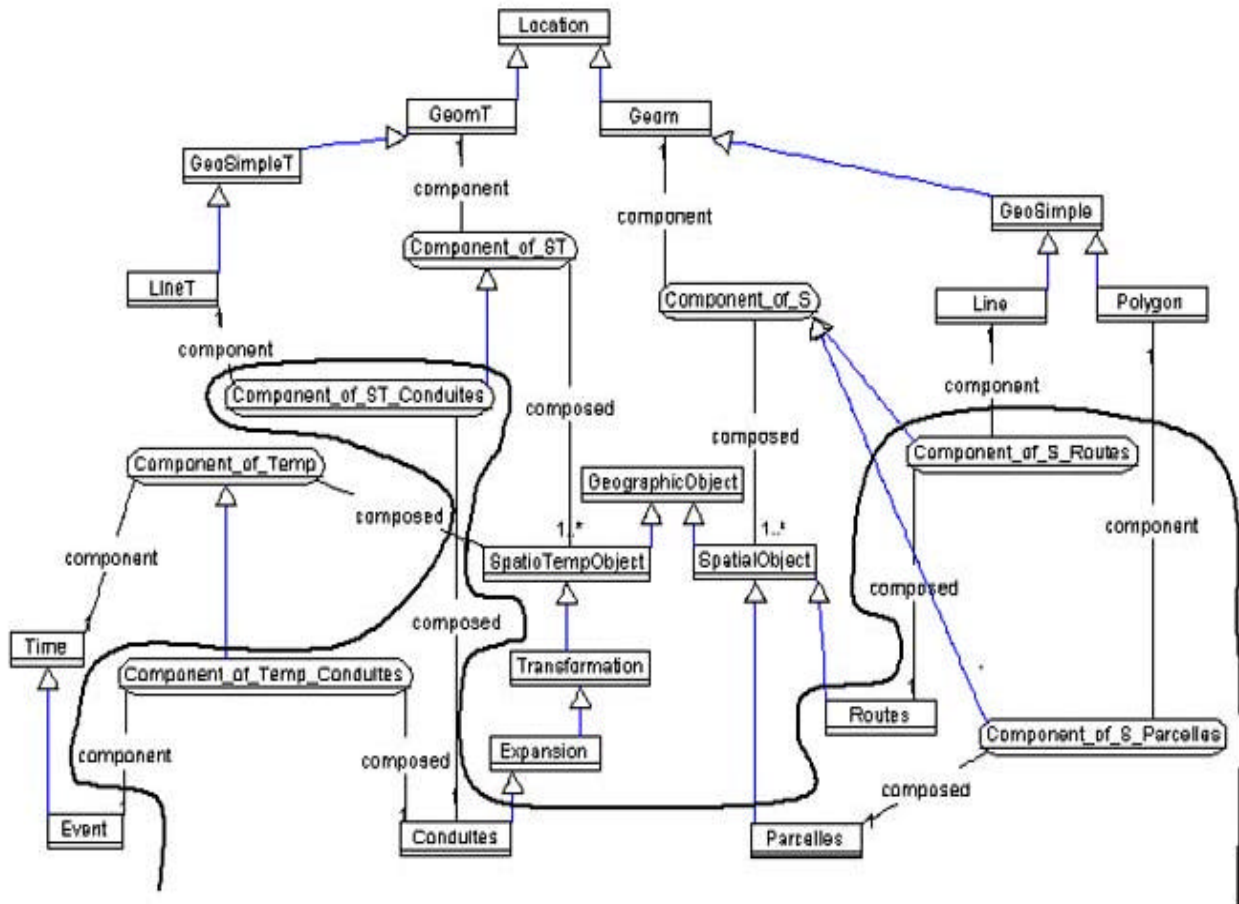


Fig. 5: Modèle objet des connaissances de l'application du Cadastre écrit en AROM étendu.

## REFERENCES

- [1] M. Scholl, A. Voisard, J-P. Peloux, L. Raynal et P. Rigaux. *SGBD Géographique, Spécificités* : ITP Pub, Paris, 1996.
- [2] A. Renolen. "Conceptual Modelling and Spatiotemporal Information Systems: How to Model the Real World". *ScanGIS '97. Red. Hauska, Hans*, pp 35-52, Stockholm, 1997.
- [3] R. Laurini and F. Milleret-Raffort. *Les bases de données en géomatique* : Edition HERMES, Paris, 1993.
- [4] P. Rigaux, M. Scholl and A. Voisard. *Spatial Databases With Application to GIS: 2<sup>nd</sup> edition*, Morgan Kaufmann Pub, 2002.
- [5] M. Koubarakis, T. Sellis, A.U. Frank, S. Grumbach, R.H. Güting, C.S. Jensen, N. Lorentzos, Y. Manolopoulos, E. Nardelli, B. Pernici, H-J. Schek, M. Scholl, B. Theodoulidis and N. Tryfona. "Spatio-Temporal Databases: The CHOROCHRONOS Approach", Springer-Verlag, 2003.
- [6] W.P. Norman, A.F. Alvaro, G. Tony. "Spatio-Temporal Databases: Contentions, Components and Consolidation". *DEXA Workshop*, pp 851-855, 2000.
- [7] C. B. Medeiros, F. Pires. "Databases for GIS". In *ACM SIGMOD Record*, 23(1), pp 107-115, march 1994.
- [8] Y. Tao and D. Papadias. "Spatial Queries in Dynamic Environments". *ACM Transactions on Database Systems (TODS)*, 28(2), pp 101-139, 2003
- [9] M. Page, J. Gensel, C. Capponi, C. Bruley, P. Genoud, D. Ziebelin, D. Bardou and V. Dupierriis, "A new approach in object-based knowledge representation : the AROM system", In *Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEAAI&ES 2001)*, Budapest, Hungary, juin 2001.
- [10] P. Christine, S. Stefano, Z. Esteban, D. Pier, P. Corinne and V. Christelle. "MADS ou l'information spatio-temporelle à porté de ses utilisateurs". *Quatrièmes rencontres de ThéoQuant*, Besançon, France, février 1999.
- [11] C. Parent, S. Spaccapietra, E. Zimányi, P. Donini, C. Plazanet, C. Vangenot, N. Rognon and P.A. Crausaz "MADS, modèle conceptuel spatio-temporel". *Revue internationale de géomatique*, pp 317-352, 1998.
- [12] J.F. Allen. "Maintaining Knowledge about Temporal Intervals", *Communications of the ACM*, 26(11), pp 832-843, November 1983.
- [13] C. Claramunt and M. Thériault. "Managing Time in GIS: An event-oriented approach, Recent Advances on Temporal Databases", Springer-Verlag, 1995.