



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

# **Etude de l'application d'INTERLIS à l'échange de données pour la gestion de l'entretien routier**

Travail pratique de diplôme



Rainer Oggier

Fevrier, 1998

Professeur: F. Golay  
Assistant: O. Paschoud

Institut de système d'information à référence spatiale  
Département de Génie Rural

# INTERLIS pour l'échange de données routières

---

Oggier Rainer

---

**Abstrait: Etude de l'application  
d'INTERLIS à l'échange de données pour  
la gestion de l'entretien routier...**



## 1.0 INTRODUCTION

---

INTERLIS est un mécanisme d'échange de données qui a été créé initialement pour l'échange de données de la mensuration officielle. L'interface de la mensuration officielle (IMO) a été formulée avec le langage INTERLIS dans l'ordonnance technique de la mensuration officielle (OTEMO). Bien qu'INTERLIS ait été créé pour des géomètres, le formalisme utilisé doit pouvoir être appliqué à n'importe quel modèle de données pour n'importe quelle base de données.

En Suisse, l'Office Fédéral des Routes (OFT) a réalisé une banque de données routières (STRADA-DB) pour améliorer la gestion de l'entretien, des constructions et de l'exploitation des routes nationales et cantonales. Cette banque de données a été répartie dans plusieurs bases de données parce que les propriétaires des routes (communes, cantons) sont aussi concernés par l'entretien des routes. De plus, ils peuvent étendre leur base de données par des outils personnalisés selon leurs besoins individuels. STRADA-DB doit agir comme noyau commun de toutes les bases de données routières de tous les propriétaires. Pour atteindre ce but, un transfert de données doit être garanti. Ainsi, le mécanisme d'échange de données est fondamental pour le fonctionnement de tous les systèmes d'information répartis et la distribution optimale de la responsabilité entre les gérants de données.

Ce travail traite de la problématique de l'échange de données à l'aide d'INTERLIS pour les échanges de données de STRADA-DB. Dans ce cadre, une étude des principales interfaces d'échanges de données routières a été réalisée.

Le besoin en données routières a augmenté ces dernières années. Certaines entreprises ayant créé des systèmes de navigation des véhicules, aimeraient une base de données qui leur permettrait de visualiser un parcours pour que le véhicule arrive au point désiré le plus rapidement possible. Bosch et Philipps ont créé un format pour l'échange de données utiles pour la navigation automobile, appelé GDF (Global Data Format) qui a été repris par le centre européen de normalisation (CEN) et déclaré comme la norme d'échange de données routières. Parallèlement, toutes les administrations nationales européennes ont élaboré un format pour leurs propres besoins, appelé RADEF (Road Administration Data Exchange Format). Pour éviter des redondances de levés de données entre les différents intervenants intéressés, la CEN et ISO sont en train de fixer un modèle standardisé qui définit les éléments de base pour décrire des informations spatiales.



## 2.0 OBJECTIFS DU TRAVAIL

But général

Le but principal est d'examiner l'adéquation d'INTERLIS pour l'échange de données routières. Après une étude des données routières et des particularités dans la conception de STRADA-DB, ce travail présente une liste de critères pour évaluer d'autres modèles routiers qui pourraient être utilisés pour le transfert de données de STRADA-DB. Il s'agit de comparer le format RADEF utilisé pour l'échange de données routières entre diverses administrations routières, et GDF créé pour la navigation automobile. Ces formats sont relativement statiques et ils peuvent transférer les entités prédéfinies et rien d'autre. INTERLIS, par contre, propose de dériver un format à partir d'une description du modèle conceptuel. Il ne définit pas cette description, mais il propose un formalisme utile pour décrire tous les modèles conceptuels. Est-ce que ce formalisme suffit pour décrire des modèles conceptuels routiers? Est-ce que le format qui est produit à l'aide de cette description arrange les données pour que l'écriture des traducteur puisse être faite efficacement? Pour répondre à la première question, nous créerons une autre liste de critères pour finalement évaluer le mécanisme d'échange INTERLIS. A l'aide d'un test le format va être examiner en programmant les traducteurs nécessaires.

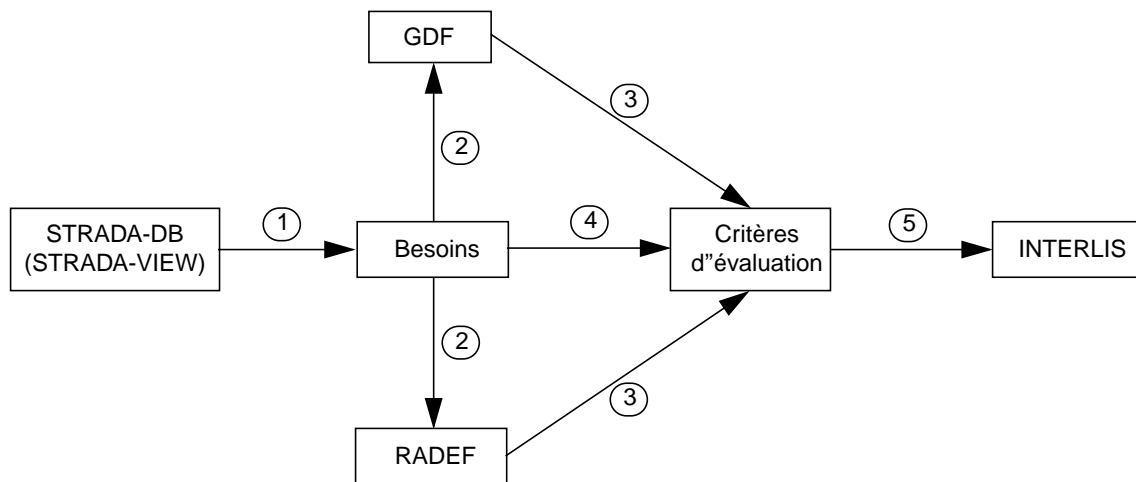


FIGURE 1.

Démarche méthodologique. Les chiffres correspondent aux numéros de sous-objectifs décrits ci-dessous.

### Les sous-objectifs

L'objectif se compose de huit sous-objectifs:

1. Etablir une liste d'exigences envers le mécanisme d'échange relatif à l'entretien routier, notamment STRADA-DB
2. Evaluer des formats d'échange de données routières existants (RADEF, GDF).
3. Mettre en évidence les exigences supplémentaires dues aux formats RADEF et GDF.
4. Etablir une liste de critères d'évaluation du mécanisme d'échange et évaluer l'importance de ces critères pour diverses activités.
5. Evaluer INTERLIS.
6. Discuter les possibilités de la représentation des données de STRADA-DB dans le schéma conceptuel d'INTERLIS.
7. Elaborer un schéma conceptuel à l'aide du formalisme INTERLIS-DDL basé sur le concept de STRADA-DB.
8. Tester ce schéma pour un jeu de données routières quelconque en écrivant les traducteurs nécessaires.

## 3.0 ACTIVITÉS ET OUTILS DE LA GESTION ROUTIÈRE

---

### 3.1 Généralités

Le but de ce chapitre est de décrire les activités et les devoirs du Système de Gestion d'Entretien des routes (SGE), ainsi que le rôle de STRADA-DB.

Système de Gestion d'Entretien (SGE)

Le 'Guide d'introduction et d'exploitation' définit le SGE comme suivant: 'Le système de Gestion de l'Entretien (SGE) représente le processus de direction, de coordination et de contrôle d'un ensemble complet d'activités en vue de l'entretien des routes, en utilisant le mieux possible les ressources disponibles tout en respectant l'environnement, c'est-à-dire en maximisant les bénéfices pour la collectivité.' Au vu de la détérioration de la chaussée des différentes routes et de l'augmentation énorme du trafic, il est nécessaire de confier des crédits aux autorités compétentes pour la rénovation des routes. Cette demande en crédit est énorme, et les investissements doivent être faits de façon à ce que les mesures aux endroits urgents puissent être réalisées immédiatement, tout en garantissant leur fonctionnement à long terme. Le SGE a aussi pour tâche de déterminer ces endroits nécessitant une intervention urgente en définissant des critères d'évaluation de l'état des routes (viabilité, sécurité, satisfaction, conservation, coûts, etc.). Ainsi, devant manipuler un grand nombre d'informations, le SGE doit faire appel à des outils modernes de gestion, donc aux outils informatiques.

### 3.2 STRADA-DB

#### 3.2.1 Généralités

Généralités

Les activités relevant du SGE se basent sur un système d'information formé par des banques de données maintenues par différents gestionnaires. Pour qu'un tel système puisse être utilisé efficacement, une sémantique claire et une structure de données commune doivent être convenues entre les différents intervenants. La sémantique est définie dans les normes SN élaborées par l'Union des Professionnels Suisse de la Route (VSS) et approuvées par l'Office Fédérale des Routes (OFR). Ce paragraphe définit la base de données STRADA-DB et décrit ses particularités.

Définition de STRADA-DB

'Le projet STRADA-DB (DatenBank für STRAssen-DAten: banque de données routières) comprend le noyau commun des banques de données routières des cantons (BDR), de l'OFR, ainsi que d'autres services

cantonaux et fédéraux. Il a pour objectif de former la base du Système d'Information du SGE.' (VSS: STRADA-DB, 'Guide d'introduction et d'exploitation') Les cantons sont responsables pour la construction, l'exploitation et la maintenance des routes cantonales et nationales. Tandis que l'Office Fédéral des Routes est chargé de l'inspection des routes nationales et de la coordination intercantonale pour l'entretien de ces routes. Comme les tâches des différentes autorités portent partiellement sur les mêmes objets, un échange de données sur ces objets rend une gestion coordonnée plus efficace. L'organisation d'un système d'information, dont les informations sont gérées par plusieurs acteurs géographiquement éloignés, exige que ce système soit réparti. En plus, on veut assurer d'une part une certaine liberté dans la conception des bases de données spécifiques à chaque gestionnaire (cantons, OFR, commune), et d'autre part la capacité de transmettre des données sans pertes. STRADA-DB est la structure commune des données de toutes les bases de données routières des cantons, de l'OFR et d'autres offices fédéraux.

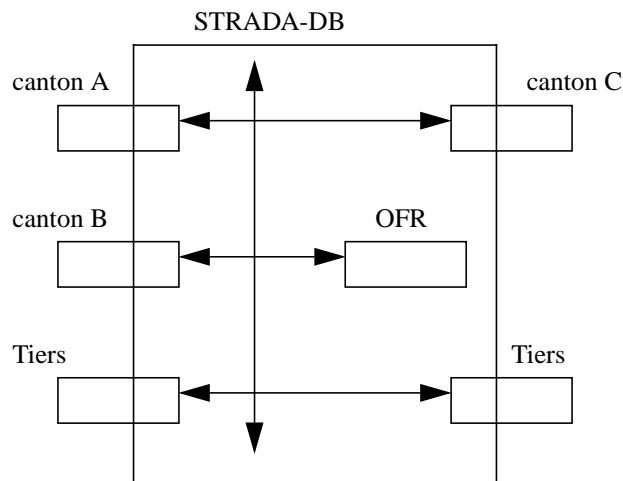


FIGURE 2.

**STRADA-DB comme noyau commun de toutes les banques de données routières (cf. Guide d'introduction et d'exploitation STRADA-DB)**

Objectifs

STRADA-DB a pour but d'offrir un outil informatique moderne qui est un soutien optimal à la décision dans le domaine de la maintenance routière. Il est le composant de base du système d'information du SGE.

Cet outil doit être ouvert, transparent et extensible de point de vue de sa conception pour que chaque utilisateur puisse s'accrocher avec ses propres unités conceptuelles. La structure de données et les applications doivent assurer l'intégrité des données. Elle doit être adaptée à l'utilisation spécifique en séparant les données et les applications.

Contraintes Le paysage politique de la Suisse se présente comme un système fédéré où une autorité nationale surveille les activités des cantons, gère les affaires administratives d'importance nationale et s'occupe de la coordination intercantonale. Les cantons eux-même sont souverains par la loi. Un projet au niveau fédérale, comme le STRADA-DB, doit respecter les particularités des cantons.

### 3.2.2 La structure de STRADA-DB

Généralités Dans la norme SN 640'900 'Management de la maintenance routière, principes', trois groupes de données sont spécifiés: les clés d'identification des entités, les caractéristiques ou propriétés alphanumériques des objets et les données de repérage.

Le catalogue de données est structuré de la manière suivante:

1. Système de repérage et réseaux
2. Description de la route et de sa structure
3. Evènement sur la route
4. Entretien courant
5. Entretien constructif et rénovation
6. Contrôles et relevés d'état
7. Données des modèles et résultats de simulation
8. Données de management

En plus, on utilise les notions définies pour le diagramme Entité-Association. On parle d'entités s'il s'agit d'un type d'objet d'information, de relations s'il s'agit d'une liaison entre plusieurs entités, et des attributs, s'il s'agit d'informations supplémentaires d'une entité.

Système de repérage Le système de repérage est essentiel pour la compréhension de STRADA-DB. La restriction de ne repérer que des points proches d'un axe de route a incité à ne pas utiliser le système de coordonnées conventionnel X, Y et Z. Car STRADA-DB n'étant pas un système d'information géographique, il n'y a aucun moyen de contrôler le repérage d'un point en utilisant les coordonnées cartésiennes. En plus, un spécialiste qui est en train de lever une caractéristique de la route, ne peut pas fournir les coordonnées cartésiennes directement sans avoir des outils de mesures appropriés et chers.

Ainsi, les inventeurs de STRADA-DB ont finalement décidé d'introduire un système de repère curviligne relatif suivant de l'axes des routes. Ceci avait aussi l'avantage de restreindre le domaine de définition des coordonnées. Si le lieu d'un accident était connu, on ne saurait à priori pas si cette paire de coordonnées correspondait à la réalité. Par contre, en indiquant des informations relatives à un axe, il serait possible de dire tout-de-suite si le point à repérer se trouve sur la route ou non. En fait, on restreint le plan bidimensionnel entièrement décrit avec les coordonnées cartésiennes à un problème unidimensionnel. Il ne faut

pas penser que l'ensemble des points du plan n'est pas représentable avec une paire de coordonnées curvilignes, mais que la description de quelques points en coordonnées curvilignes ne fait aucun sens: par exemple un point ayant un écart d'un kilomètre d'un axe n'a pas de rapport avec cet axe.

Pour STRADA-DB, on a créé un système de repérage, appelé système de repérage de base (SRB) en matérialisant des points de repère (PR) au bord des routes. Ces points ont été implantés tous les kilomètres dans la zone rurale et approximativement tous les 300 mètres dans la zone urbaine et dans des zones où la géométrie de la route l'exige. Chaque PR est accompagné d'un signe précisant la position du PR par rapport au signe, le numéro du PR et le propriétaire du point. De plus, la projection du PR sur l'axe est marquée par un carre jaune sur la chaussée. Si le PR se trouve à l'extrémité de l'axe, il est représenté par un cercle jaune sur la chaussée. L'axe est défini comme l'axe d'entretien, c'est-à-dire que pour une route à deux voies il se trouve au milieu de la chaussée et pour les autoroutes au bord gauche en direction du trafic. Une autoroute a donc deux axes.

La position d'un point est déterminée en espace en précisant l'axe (par exemple: BL.N2+), le numéro du PR, la distance de référence du lieu par rapport à la projection du PR sur l'axe le long de l'axe et l'écart. Ce dernier est la distance perpendiculaire à l'axe entre l'axe et le lieu.

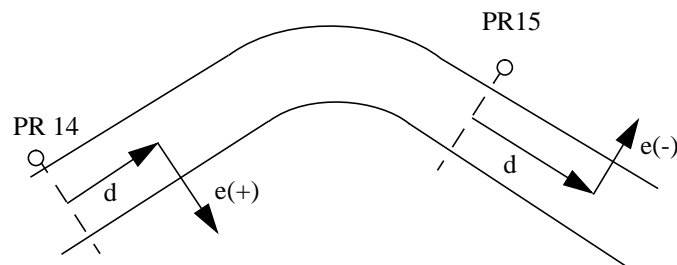


FIGURE 3.

**Les paramètres nécessaires pour qu'un point soit défini dans le système de repérage de base. (d: distance longitudinale, e: écart)**

Il existe parfois des situations où une route est discontinue, par exemple la N1 (A1) est interrompue par des routes nationales nord-sud. C'est pour cela que la notion de segment a été introduite, qui est définie par la portion continue de l'axe. Un axe peut avoir plusieurs segments, mais il n'existe pas de segments qui appartiennent à plusieurs axes.

Réseaux d'exploitation

Les réseaux d'exploitation permettent la constitution de différentes vues du réseau routier. Ces réseaux sont attachés au système de repérage de base. Ils permettent de définir des 'objets' géométriques et des propriétés topologiques en fonction d'un objectif particulier. Le SGE

utilise ces réseaux pour représenter les parcours d'entretien, les itinéraires de transports spéciaux, le réseau des routes cantonales, etc. .

Les réseaux d'exploitation se réfèrent dans le système de référence de base à l'aide des noeuds. Un noeud représente, en général, l'intersection de deux routes, une limite administrative ou une limite d'un projet. En plus, des noeuds simples et des noeuds complexes sont distingués. L'aspect de l'échelle qui entre en jeu pour la définition de ces deux notions.

Prenons un exemple, une personne responsable pour le déneigement des routes doit dégager une jonction de plusieurs autoroutes. Il ne lui suffit pas de savoir où se trouve cette jonction, mais elle est plus intéressée à connaître tous les raccordements existants pour finalement trouver un trajet optimal. Les intersections de routes et les raccordements ayant une existence physique sont appelés noeuds simples. Pour un touriste, les raccordements exacts ne sont pas très intéressants, mais il veut savoir où se trouve la jonction qui lui permet de changer de route. Ces noeuds ne doivent pas forcément avoir une existence physique. Le lieu où une route passe au-dessus d'une autre au moyen d'un pont ne permet pas au conducteur de changer de route. Il suffit simplement de savoir que le noeud simple se trouve à proximité. Ces noeuds 'virtuels' sont appelés noeuds complexes, ils représentent tous les noeuds simples ayant une liaison directe avec ses deux routes à un niveau plus global.

Les éléments géométriques entre deux noeuds sont appelés tronçons. Ils sont définis conformément à une règle de découpage, comme par exemple classification nationale, découpage en réseaux d'entretien, etc. . Les groupes de tronçons sont définis par un ensemble de tronçons qui satisfont un critère supplémentaire, par exemple les numéros des routes, les numéros de la commune, etc. . Noeuds, tronçons, groupes de tronçons constituent la base des réseaux d'exploitation.

Comme il peut exister plusieurs réseaux d'exploitation, ils ne peuvent pas être utilisés comme système de référence de base.

Lors de la création d'un réseau d'exploitation, il est souhaitable de définir sa topologie. Cette tâche consiste à préciser des propriétés déterminantes, comme la continuité, la connexité, l'absence de recouvrement, l'ordre de succession et l'orientation. Ces restrictions doivent rendre la base de données cohérente.

#### Clés d'identification

Pour assurer l'intégrité du système d'information réparti, une identification des différentes banques de données, des utilisateurs ou des groupes d'utilisateurs et des entités au niveau conceptuel est réalisée de façon à ce que cette clé d'identification soit unique dans toute la Suisse. L'Union des Professionnels Suisse de la Route (VSS) est responsable que tous les propriétaires et toutes les banques de données, tous les codes et textes structurés communs reçoivent une clé unique. Un propriétaire d'une banque de données peut créer pour ses propres besoins

des banques de données subordonnées. La clé d'identification se composerait de la clé du propriétaire et de la clé de la banque de données définie par le propriétaire, par exemple NE:01. Les deux parties de la clé s'appellent clés partielles. Si l'on veut accéder à la banque de données principale, la clé du propriétaire sera répétée, par exemple NE:NE. Dans ce cas, le propriétaire s'appelle NE, et la banque de données s'appelle NE:NE.

Pour les clés conceptuelles, le même principe s'applique. Mais en plus, on prend comme hypothèse que le propriétaire peut gérer les clés de sa banque de données sans ambiguïté. En précédant la clé interne par la clé du propriétaire, l'identification de l'entité est unique.

Gestion des originaux et des copies

Dans le cadre de STRADA-DB, il a été convenu que toute la modification ne peut être faite que sur l'original du jeu de données et un original peut être dupliqué. Sur les copies aucune modification n'est autorisée. Le problème se trouve dans le transfert des originaux. Comment est-il possible de transférer l'original, et comment peut-on garantir que l'original est unique en toute la Suisse pour un jeu de données déterminé? La clé du jeu de données contient la clé de la base de données où l'original se trouve. Pour une copie, la clé contient la clé de la base de données indiquant où l'original se trouvait à l'époque où la copie a été créée, parce qu'une copie n'est pas mise à jour. La clé ne peut être changée que par le propriétaire de l'original. En résumé: si la clé du propriétaire ne fait pas partie de la clé du jeu de données qu'il possède, il s'agit d'une copie, sinon de l'original. Quand le propriétaire veut transférer l'original à un autre gestionnaire, il remplace tout simplement la partie de la clé contenant son identité par l'identité de la base de données cible et il transfère le jeu de données. Le destinataire s'aperçoit directement s'il s'agit de l'original en comparant la clé d'identification de sa base de données avec la partie de la clé décrite ci-dessus.

Listes de codes

Pour avoir un certain contrôle d'intégrité pendant la saisie des informations, on a restreint le domaine des valeurs admissibles pour quelques attributs. La solution générale consiste à créer une liste des valeurs admissibles pour un attribut donné. Souvent ces listes contiennent des codes pour chaque valeur ce qui rend la saisie de données plus efficace. Il est plus simple pour une personne sur le terrain, qui lève des objets, de pouvoir entrer un code court au lieu de mettre un texte plus long, surtout si celui-ci se répète souvent. Naturellement, elle doit avoir connaissance de cette liste. L'inconvénient de ces listes de codes reste dans l'utilité si elles sont trop longues. Pour ce cas, on a inventé des catalogues de texte dans STRADA-DB.

Catalogues de textes

Les listes de codes sont très statiques. Une valeur, qui n'est pas sur la liste, n'est pas admissible même si elle a un sens. On pourrait simplement déterminer le domaine de définition de l'attribut comme texte libre, mais un contrôle ne serait plus possible. En plus, une interrogation s'avérerait impossible, parce que l'utilisateur aurait, par exemple, la possibilité de décrire avec 'bitume', 'goudron', 'couche bitumineux'

ou 'AB\_11' la même caractéristique de l'attribut *Type de couche*. Le catalogue de textes est un compromis de ces deux formes 'extrême' de description d'une caractéristique. Il s'agit, en fait, d'un système de codes structurés. Ces catalogues de texte rassemblent des connaissances spécifiques au domaine routier dans STRADA-DB.

Un catalogue de texte est une table, dans laquelle chaque colonne contient des textes élémentaires décrivant la même propriété. Par exemple, le catalogue *Type d'Axe* contient une colonne *Nature* qui possède des textes élémentaires comme *voie de circulation*, *cours d'eau*, ... . Toutes les combinaisons plausibles entre les différents textes élémentaires des différentes colonnes sont décrites dans une liste, et elles sont accompagnées par un code. Il existe des règles de combinaisons de textes élémentaires, soit entre des valeurs particulières des différentes colonnes, soit entre des colonnes entières. En plus, la suite des textes élémentaires peut être fixée.

**EXEMPLE:**

Catalogue: Type de point de repère physique

Domaine (Colonne):

Matériau: (cheville, borne, construction spéciale, etc.)

Emplacement: (sur la chaussée, dans la berme centrale, etc.)

Fonction primaire: (spécifique BDR, de la mensuration officielle, etc.)

Fonds: (sur propriété privée)

Exemples de textes composés:

Cheville BDR sur la chaussée.

Borne de la mensuration officielle hors accotement sur propriété privée.

Construction spéciale BDR dans la berme centrale.

La saisie de texte complexe peut se faire de la façon suivante:

- Le texte composé peut être directement choisi en entrant son code qui se trouve dans une liste imprimée de tous les codes des catalogues au préalable.
- L'application conduit l'utilisateur d'une colonne à l'autre et il peut choisir un texte élémentaire dans chaque colonne.
- L'utilisateur choisit librement la suite des colonnes dans lesquelles il veut choisir un texte élémentaire.

Les catalogues peuvent être modifiés ou complétés par les gestionnaires.

Langue

En attachant une clé d'identification aux textes élémentaires, se composant d'un numéro de texte et d'un code de langue, on peut les faire apparaître dans n'importe-quelle langue. Quand on compose des règles de composition, on appelle les textes élémentaires par leur numéro ce qui permet d'accéder aux textes élémentaires en différentes langues ayant une signification pareille (par exemple 501: 'Kiesschicht', 'Cou-

che de graviers'). De même pour les listes de codes. Ainsi tous les codes sont indépendants de la langue dans laquelle les textes élémentaires sont écrits.

#### Historique

La gestion de l'historique est essentielle dans STRADA-DB. Elle permet au SGE une planification à long terme des mesures d'entretien, ainsi que d'établir des statistiques sur les investissements en fonction de l'espace dans le passé. Pour renseigner sur la période de validité d'une information, trois attributs sont réservés, un pour la date de début de validité, un pour la date de fin de validité et un pour la date de référence. Par date de référence, on entend le moment où le gestionnaire a fait connaissance de cette information, ce qui est souvent la date de saisie de l'information.

Il y a trois scénarios possible: un évènement ponctuel déterminé par une date et une heure précise (par exemple un accident), une activité dans le passé déterminée par un intervalle dans le passé et un état ayant une durée indéterminée. Si la date du début de validité est égale à la date de fin de validité, mais plus petite que la date de référence, il s'agit d'un évènement. Si la date du début de validité est inférieure à la date de fin de validité et les deux dates sont encore inférieures à la date de référence, il s'agit d'une activité. Un état est défini par une date de référence à l'intérieur de l'intervalle formé par la date du début de validité et la date de fin de validité. On dit alors que l'état est effectif. Il existe des états dont la date de référence est antérieure à l'intervalle de validité, il s'agit d'une prédiction ou d'un projet.

La gestion d'un historique est uniquement nécessaire pour les états, car les évènements et les activités se passent une fois et ne peuvent pas évoluer. Par contre les états dans le passé doivent être gérés. Il ne peut pas avoir un intervalle de temps dans lequel aucun état était valable pour un objet donné. Le changement d'état entraîne la création d'une nouvelle version de cet objet. En plus, les mutations dues au changement du système de repérage n'impliquent aucune création de versions, car elles peuvent être déduites de l'historique du système de repérage de base. Les corrections des données dues à des fautes ne font pas l'objet de l'historique.

L'historique pose quelques problèmes au niveau logique. Au niveau conceptuel, on identifie les objets par une clé conceptuelle, mais au niveau logique, le système informatique génère des clés lui-même, appelées Base-ID, composées de la clé de la base de données mère et du numéro donné par le système informatique. La modification d'un attribut d'une entité ne change pas l'entité. Donc, le Base-ID est toujours valable pour cette entité. Si l'on veut acquérir une entités et surtout les attributs de cette entité à une date donnée, il faut faire une requête qui recherche la version de cette entité pour laquelle la date souhaité se trouve à l'intérieur de l'intervalle de validité. Cette requête est effectivement faite pour STRADA-DB. Donc, le BASE-ID ne détermine pas entièrement tous les attributs d'une entité. C'est la raison pour laquelle

on a ajouté le numéro de la version au BASE-ID pour avoir un identifiant unique indépendant du système.

L'introduction du numéro de la version dans la clé entraînerait une dénormalisation parce que la relation entre deux entités posait des difficultés comme le montre la figure 4.

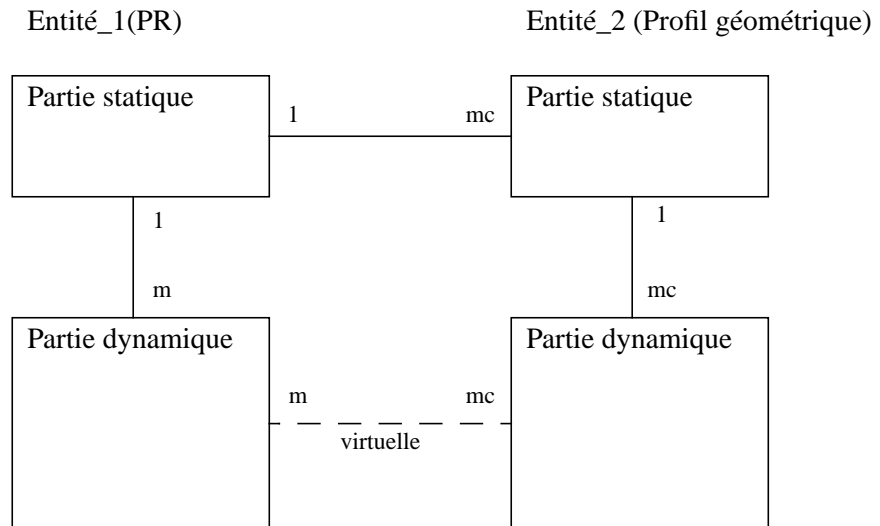


FIGURE 4.

**Relations entre deux entités au niveau logique.**

Prenons l'exemple du *point de repère* (PR) et du *profil géométrique* de la route. Il peut exister plusieurs profils qui sont repérés par rapport à un seul PR, mais un *profil géométrique* ne peut pas être repéré par deux PR. Chacune de ces entités possède une partie statique qui se compose surtout des attributs administratifs, et une partie dynamique (par exemple les attributs décrivant la nature des PR) qui peut changer à tout moment. La relation d'une version de *profil géométrique* est référencé par exactement un PR et indépendamment de la façon dont il est matérialisé, il doit s'agir du même PR. C'est-à-dire que la partie statique du PR entre en relation 1:1 avec la partie dynamique du profil en travers. Les deux parties dynamiques n'ont qu'une relation virtuelle m:mc. STRADA-DB propose de créer deux tables pour chaque entité et de faire la liaison entre la partie dynamique du *profil géométrique* et la partie statique du PR, comme représenté à la figure 5. En pratique, les deux tables sont clairement identifiées par le Base-ID et le numéro de la version. Pour faire la liaison, on n'ajoute que le BASE-ID de l'entité entrant en relation 1:1 (Ici PR) dans l'autre table (*Profil géométrique*). Cette nouvelle structure ne correspond plus à la deuxième forme normale, parce que les attributs ne dépendent plus de la clé entière, c'est-à-dire que la partie statique dépend que du BASE-ID et pas du BASE-ID et du numéro de la version.

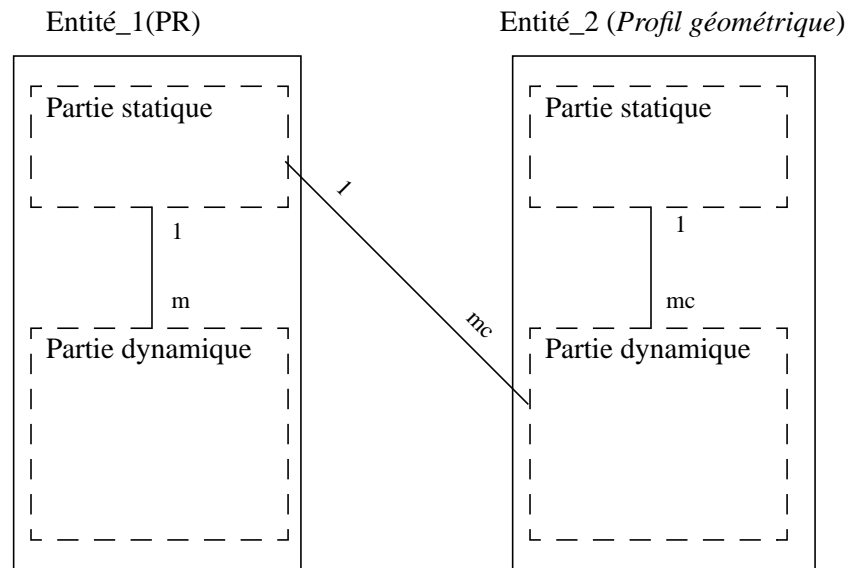


FIGURE 5. Modélisation d'une relation entre deux entités (*Profil géométrique* et PR).

### 3.2.3 Contrôle de cohérence

#### Généralités

STRADA-DB étant une base de données complexe, possède beaucoup de contraintes. En général, le domaine de définition de la valeur d'un attribut est restreint, il y a des relations entre différents attributs dont la combinaison peut être contradictoire. Pour garder la base de données cohérente, des contrôles doivent être effectués, surtout pour l'intégration de nouvelles données.

#### Définitions

On peut effectuer des opérations au niveau de la base de données directement, c'est-à-dire qu'on peut insérer ('insert'), corriger ('correct'), sélectionner ('select'), effacer ('delete') et écrire l'historique ('history') avec SQL directement dans la base de données ORACLE. Ces opérations sont appelées opérations de la BD. Mais elles ont dû être élargies pour pouvoir garantir la cohérence de la base de données. L'outil *Application Programming Interfaces* (API) fournit des fonctions, appelées opérations API, qui réalisent les opérations de la BD et les tests de cohérence nécessaires. Les opérations API de base portent le même nom que les opérations de la BD: 'insert', 'correct', 'select', 'delete' et 'history'.

Parmi les fonctions API se distinguent les fonctions de test et les fonctions de manipulation. Les fonctions de test garantissent l'intégrité de la base de données en empêchant toutes les opérations qui ne respectent pas les contraintes d'intégrité. Les fonctions de manipulation réalisent le même but en modifiant des données, par exemple l'augmentation du numéro de la version.

Etats d'intégrité

Pour l'échange d'un nombre réduit de données, les tests de cohérence peuvent se faire manuellement. Par contre, l'intégration d'un grand nombre de données dans STRADA-DB exige une validation automatique des données. Avant l'intégration des données dans STRADA-DB, l'intégrité des nouvelles données est garantie par le API dans presque tous les cas.

Les API fournissent l'état d'intégrité des données se composant de l'intégrité structurelle et de l'intégrité sémantique. Malgré des tests rigoureux, on ne peut pas toujours assurer l'intégrité structurelle car on exige aussi qu'il soit possible d'importer des données ne remplissant pas les tests d'intégrité, notamment des textes élémentaires non existants. D'autre part, le gérant peut se donner le privilège de manipuler les données avec SQL et d'ainsi contourner les API. En plus, dans des cas spéciaux, les API ne peuvent pas toujours exécuter tous les tests nécessaires, ainsi si la validité temporelle n'est contrôlée que par rapport à la version antérieure. L'intégrité sémantique doit être contrôlée aussi par l'utilisateur, par exemple le chevauchement de deux bandes latérales.

Il existe trois codes d'états d'intégrité et un quatrième (3) est proposé pour mieux distinguer les défauts sémantiques et les défauts structuraux:

0	valid
1	semantic invalid
2	invalid
3	invalid, semantic invalid

TABLE 1.

Les états d'intégrité

Dépendances

Dans le contexte des contraintes d'intégrité, la compréhension des dépendances entre différents objets est essentielle. Ceci parce que ces dépendances définissent l'ordre des contrôles de cohérence à suivre. Le lieu d'un accident, par exemple, ne vaut rien si l'exactitude du point de repère auquel est attaché l'emplacement n'est pas garantie. Les objets possédant des objets dépendants sont appelés masters. Les dépendances directes entre deux objets sont appelées des dépendances master-détail, par exemple *Axe - Point de repère*. Si deux relations ont une liaison très vague, il s'agit d'une dépendance master-master, par exemple *Point de repère - Structure de la chaussée* (Couche). Dans ce rapport, on désigne les objets supérieurs par masters et les objets subordonnés par détails.

Une clé conceptuelle unique

Dans STRADA-DB, on utilise le principe *Unique par substitution* pour garantir que la clé conceptuelle d'un objet est unique à tout instant. On suppose que la clé conceptuelle du master est déjà univoque, lorsqu'on contrôle un objet subordonné. Dans une dépendance master-détail, la clé conceptuelle du master fait partie de la clé conceptuelle du détail. En substituant la clé partielle provenant du master par le BASE-ID de

celui-ci, l'unicité du master est garantie et le contrôle se réalise avec ce BASE-ID et la clé partielle du détail.

Pour illustrer le concept de la substitution, voici un exemple:

Relation	nom logique	Explication	Substitution
Axe	CKO_OWNER	Responsable de la clé conceptuelle	CKO_OWNER
	CK	Clé conceptuelle	CK
	POS_CODE	droite, gauche, rien, (!!Autoroute)	POS_CODE
Ref-Point	AXE.CKO-OWNER	Clé étrangère	AXE_BASEID
	AXE.CK	Clé étrangère	
	AXE.POS_CODE	Clé étrangère	
	CK	clé partielle due à Ref.Point	CK

TABLE 2.

**Le concept de la substitution**

Test *not nul*

Il existe des attributs qui exigent toujours une valeur. Il s'agit des clés conceptuelles, du début de validité, de la date de référence et du code de la version (effectif ou prédit). Le test *not nul* est supposé trouver les tuples faisant défaut de cette exigence.

Test sur les attributs temporels

Ce test compare la date de référence, le début de validité, la fin de validité et le code de la version. Selon le code (effectif, prédiction), l'ordre temporel des valeurs des attributs change.

Contrôle de la validité de deux objets dépendants

Le contrôle de la fin de validité temporelle entre deux objets d'information n'est pas réalisé. Par contre, la consistance entre le début de validité de deux objets dépendants est contrôlée par les API. Le premier contrôle n'a pas été réalisé afin d'obtenir une plus grande flexibilité pour terminer la validité d'un objet. Par ce 'défaut', la situation peut apparaître que le master a été terminé, mais que le détail est encore valide. Dans ce cas, STRADA-DB prend la dernière version de l'objet supérieur. Le *test relationship time* contrôle d'abord l'existence du master et ensuite si l'intervalle de validité du détail est contenu dans l'intervalle du master pour les opérations 'insert', 'history' et 'correct'. Ce test se réalise dans la direction du détail vers le master. Mais comme déjà mentionné, ce test n'est pas réalisé dans l'autre direction. Le *test relationship object down* contrôle toutes les relations avec des objets subordonnés avant d'effacer leur master.

Test de la séquence

Le test de la séquence s'applique lors de l'insertion, de la correction ou de l'effacement d'une version effective d'un objet. Ce test implique d'abord un test d'unicité qui contrôle l'unicité d'un objet à l'intérieur d'une version en tenant compte de sa période de validité. Naturellement, deux objets avec la même clé conceptuelle pour un intervalle de temps donnée ne peuvent pas exister. Lorsqu'on insère une nouvelle version d'un objet, la validité de la version antérieure doit être terminée, c'est-à-dire que la fin de validité de la version antérieure doit reprendre la valeur du début de validité de la version postérieure. Si l'on change le

code de version de 'prédiction' à 'effective', on crée une nouvelle version et le test de la séquence s'applique. Si l'on change le code de version de 'effective' à 'prédiction', on efface une version. La fin de validité ne peut être modifiée par l'utilisateur que lorsqu'il s'agit de la version la plus actuelle. Pour toutes les autres versions, la fin de validité doit être modifiée par le système parce qu'il n'est pas possible de prédire où la version à insérer va s'intégrer. Cette version ne doit pas forcément être la version la plus actuelle.

#### Verrouillage

La solution du verrouillage est choisie dans beaucoup de systèmes d'information géographiques: on exporte une partie d'une base de données à modifier sur un système moins lourd que la base de données mère et on verrouille cette partie dans la base de données mère pour que personne d'autre ne puisse modifier la même partie (appelé Check Out). Ainsi, on évite des incohérences dues à des modifications simultanées. Le même scénario ne peut pas se produire pour STRADA-DB car les modifications ne peuvent être faites que sur l'original. Malgré cela, le problème des modifications simultanées existe. Un utilisateur peut modifier des données à l'aide d'une application, pendant qu'un autre fait des modifications avec les API. Exemple: Un utilisateur X efface un axe et fait encore d'autres modifications. Entre temps, un autre utilisateur y crée un point de repère se référant à cette axe et exécute cette opération dans la base de données mère avant que l'utilisateur X ait exécuté les siennes. La base de données mère se trouve dans un état incohérent, puisque le point de repère se réfère à un axe non existant. Ce problème peut être éliminé en verrouillant l'axe lorsqu'on exécute la commande 'insert'. Ce verrouillage est aussi possible pour les opérations 'history' et 'correct'.

Il existe deux types de verrouillage. Le verrouillage simple ne concerne que les objets modifiés, sauf pour 'history' où toutes les versions de l'objet sont verrouillées. C'est le verrouillage de l'objet. L'autre type est le verrouillage de la relation. Le master d'un objet modifié alors verrouillé.

### 3.2.4 STRADA-VIEW

#### Définition

Comme STRADA-DB est purement alphanumérique, il est pénible de se situer par rapport à des objets marquants du paysage. On a besoin d'une carte ou d'un plan d'ensemble en arrière-plan pour pouvoir se situer. Au niveau informatique, c'est justement STRADA-VIEW qui fait cette superposition.

STRADA-VIEW est un outil de représentation et d'analyse des informations géographiques contenues dans STRADA-DB. En plus, il permet de représenter des unités paysagères n'ayant pas de liaisons directes avec STRADA-DB, mais qui sont importantes pour pouvoir se situer.

STRADA-VIEW est créé à l'aide du logiciel ARCVIEW de ESRI. Il accède les informations de STRADA-DB via ODBC (Open DataBase Communication). Pour des raisons techniques, un sous-ensemble de données de STRADA-DB a été copié dans ACCESS et STRADA-VIEW acquiert les données alphanumériques d'ACCESS. Cet accès indirect aux données de STRADA-DB pose un problème de mise à jour des données utilisées par STRADA-VIEW.

Insertion de la géométrie

Cet outil type système d'information géographique ajoute la géométrie des routes. Il s'agit tout d'abord d'une géométrie de représentation, qui n'a pas le but de corriger les données de repérage dans STRADA-DB. La géométrie des axes a été digitalisée à partir des plans d'ensemble 1:10'000, c'est-à-dire que des lignes avec des paires de coordonnées X, Y ont été créées. De plus, toutes les longueurs des axes sont calculées. En tenant compte du rapport entre la longueur totale des axes dans la géométrie de représentation et la somme de la longueur des secteurs (distance entre deux PR) dans STRADA-DB les coordonnées X, Y et Z des PR sont calculées et insérées dans STRADA-VIEW. Connaissant les PR en coordonnées absolues, tous les objets géographiques (point, ligne et surface) peuvent être calculés.

Axe tendu

Si l'on veut représenter une seule variable sur plusieurs axes, une carte géographique peut nous satisfaire. Par contre, si l'on veut représenter plusieurs variables d'un seul axe, les cartes géographiques ne le permettent pas sans introduction d'une nouvelle variable synthétique. Par exemple, une carte géographique représente bien l'état de la chaussée des axes de routes dans le canton de Vaud. Par contre, si l'on veut connaître l'état de la chaussée, le profil géométrique ou les réparations effectuées les dernières années sur une route cantonale dans le canton de Vaud, les caractéristiques sont difficilement représentables sur une carte, parce que ces variables se trouvent au même endroit géographique. C'est pour cela que le concept de l'axe tendu a été introduit. Il s'agit d'une représentation non de la géométrie mais de l'axe comme droite. Chaque variable est représentée par un graphique en fonction de la distance à partir du début de l'axe.

## 4.0 CADRE THÉORIQUE D'ÉCHANGE DE DONNÉES

---

### 4.1 CEN TC 287 et ISO TC 211

#### Généralités

Le Centre Européen de normalisation CEN TC287 a comme but de créer un standard dans le domaine du transfert des données géographiques. Il décrit par des normes comment définir, décrire et transférer la réalité géographique. Au niveau international, c'est l'ISO TC211 (International Standardization Organization), qui s'occupe des normes pour la création d'une structure standardisée des données géographiques. Une standardisation est surtout nécessaire pour l'échange de données au niveau international. Mais cette tâche est très ardue parce que tous les pays aimeraient maintenir leur description de données et leur format. Ces normes sont souvent des compromis entre différents pays. Les solutions proposées, non encore mise en vigueur, sont appelées pré-normes.

Par informations géographiques, on entend des informations sur un phénomène directement ou indirectement associé à un lieu. Elles se composent de l'aspect sémantique, spatial et de qualité. En définissant toutes les entités et les relations entre elles, on crée un schéma conceptuel. Mais comme il n'existe pas de modèle conceptuel représentant parfaitement la réalité, le CEN définit un schéma d'application comme une description du modèle conceptuel créée pour un projet particulier. Dans la catégorie de schéma d'application, on distingue encore les schémas internes créés pour un système informatique spécifique, et les schémas externes créés pour un transfert spécifique. Le CEN TC 287 ne définit pas les schémas d'application, mais il fournit des règles pour leur création.

#### Le transfert des données

La figure 6, ci-dessous, montre l'organisation de la préparation des données pour un transfert quelconque. Le schéma d'application, ses métadonnées et le jeu de données qui satisfait le schéma d'application, sont les éléments à transférer. Le schéma d'application et les métadonnées sont codées dans le dictionnaire de données en utilisant les règles définies par le schéma de langage. Ce codage est nécessaire pour avoir un schéma d'application standardisé sur lequel on peut tester la cohérence. Le schéma d'application devrait être écrit en EXPRESS pour pouvoir le traduire automatiquement dans le dictionnaire de données. Le schéma de langage spécifie les notions d'EXPRESS à utiliser pour définir le dictionnaire. Le dictionnaire représente les métadonnées pour le transfert. Le jeu de données est codé selon le schéma de codage. Ce codage est nécessaire pour que les données soient indépendantes de la langue utilisée et que les différents transferts soient similaires. A partir des métadonnées pour le transfert et les données codées, on peut préciser quelles données et quelles métadonnées sont à transférer. Il peut être

souhaitable de n'exporter qu'une partie de données. Cette procédure doit se dérouler selon le schéma de transfert. Le format d'échange est finalement défini par les règles de codage.

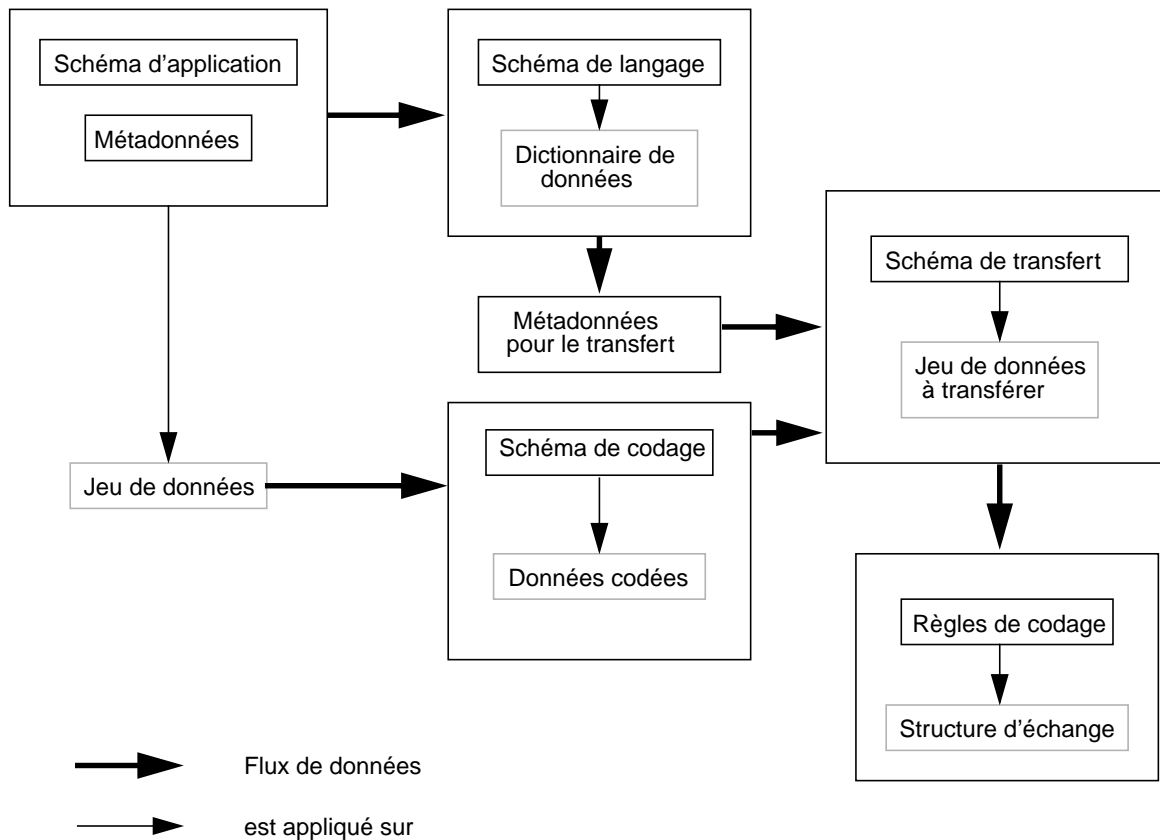


FIGURE 6.

Diagramme de la préparation des données pour le transfert

## 4.2 INTERLIS

### Définitions

INTERLIS ne se distingue pas fondamentalement de la procédure du CEN, et il s'agit surtout de pouvoir situer l'INTERLIS dans le mécanisme décrit ci-dessus. Par INTERLIS on entend le processus d'échange de données comprenant un schéma écrit en INTERLIS-DDL (Data Description Language) et produisant un format de données. INTERLIS est une interface entre plusieurs systèmes qui permet l'échange de données géographiques, alphanumériques et leurs relations en respectant la cohérence des données.

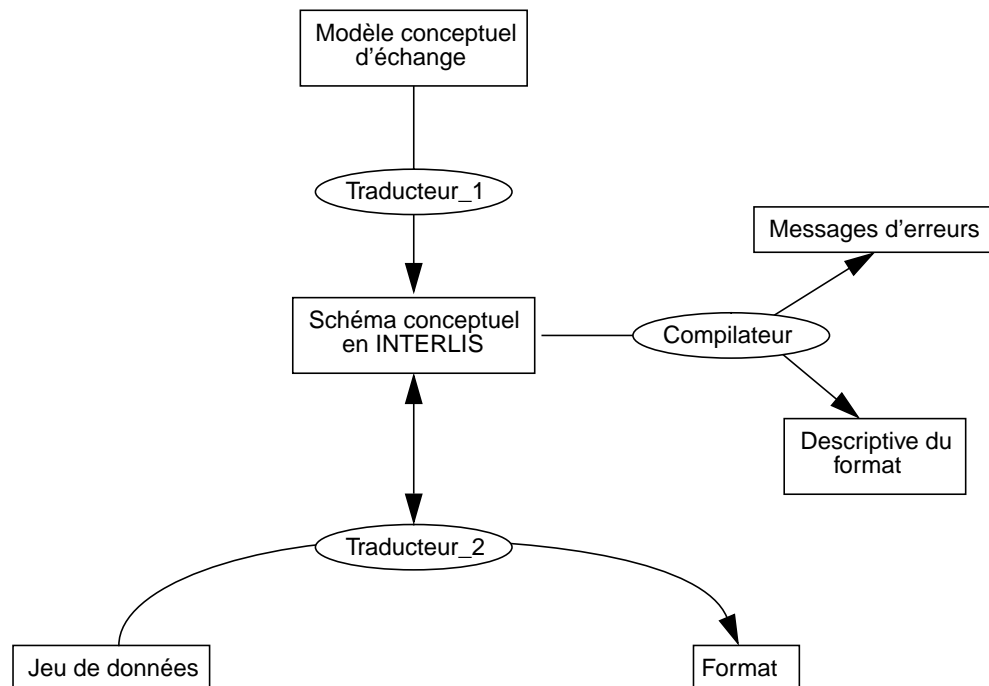


FIGURE 7.

Procédure

**Procédure de transfert des données selon INTERLIS.**

La figure 7 décrit surtout l'export des données. On admet que une communauté d'information définie comme la communauté de tous les utilisateurs ayant la même sémantique, s'est mis d'accord pour un modèle conceptuel commun dit de communication. Si cet accord n'est pas possible, il ne vaut pas la peine d'établir un échange standardisé entre les participants d'une communauté. Ce modèle doit être décrit avec le formalisme INTERLIS-DDL. Le schéma résultant doit être correctement formalisé. Le compilateur, aussi appelé Parser, peut détecter toutes les fautes et les écrire dans un fichier. Il crée encore un fichier descriptif du format, qui facilite la lecture du format généré à l'aide du schéma conceptuel ILI (Abréviation pour INTERLIS). Un autre traducteur, cette fois informatique, transforme le jeu de données en format ITF (INTERLIS Transfer Format).

Comparaison de la procédure de INTERLIS avec celle des organisations internationales.

Comparé au mécanisme d'échange du CEN, le schéma en INTERLIS-DDL peut être considéré comme le dictionnaire, sauf que ce schéma inclut aussi les métadonnées du transfert. En plus, le schéma conceptuel n'est pas inclus dans le format, mais traité séparément. Si l'on trouve un format sans fichier ILI, il ne sert à rien. Par contre, l'avantage est de ne pas avoir à transférer le schéma conceptuel pour chaque jeu de données. Le fichier ILI n'est pas codé, il décrit le fichier ITF. Pour le CEN toutes les informations, les métadonnées du transfert et les données, sont codées. Un codage et un décodage selon une norme ISO est nécessaire,

sinon on ne comprend pas ce que la structure de données représente. Avec INTERLIS, la thématique du fichier de données est connu si l'on a le schéma conceptuel qui n'est pas codé. En plus, la procédure du CEN resp. ISO est beaucoup plus compliquée, elle possède plusieurs étapes de codage. Ceci est dû à l'utilisation d'EXPRESS ayant une centaine de mots clés (INTERLIS: env. 30). A chaque étape, un schéma en EXPRESS modifié - du à des restrictions décrites dans le schéma de langage - définit les unités de base pour décrire les informations correspondantes. Les unités de base d'INTERLIS sont décrits dans la documentation 'Mécanisme d'échange de données pour systèmes d'information du territoire'. Dans la procédure du CEN resp. ISO les schémas sont décrits dans une prénorme et ils ont un caractère normatif dès que les membres du CEN approuvent la prénorme.

## 5.0 ECHANGE DE DONNÉES DANS LE CADRE ROUTIER

### 5.1 Besoins

L'échange entre deux STRADA-DB est essentiel parce que l'Office Fédéral des Routes étant responsable de la coordination entre les différents cantons, a besoin des données des cantons et vice versa. Ce transfert de données est déjà réalisé par un format propriétaire de STRADA-DB. Par contre, les échanges entre STRADA-DB et les logiciels bureautiques, et entre STRADA-DB et les systèmes d'information ne sont pas du tout réglés (Figure 8). La solution qui consiste à utiliser des formats propriétaires n'est pas souhaitée parce que pour tous les échanges entre STRADA-DB et les différents systèmes extérieurs, on aurait besoin autant de formats que de systèmes extérieurs. Pour éviter de devoir créer un format propre pour chaque nouveau système, un format standard valable pour n'importe quel mécanisme d'échange est utilisé.

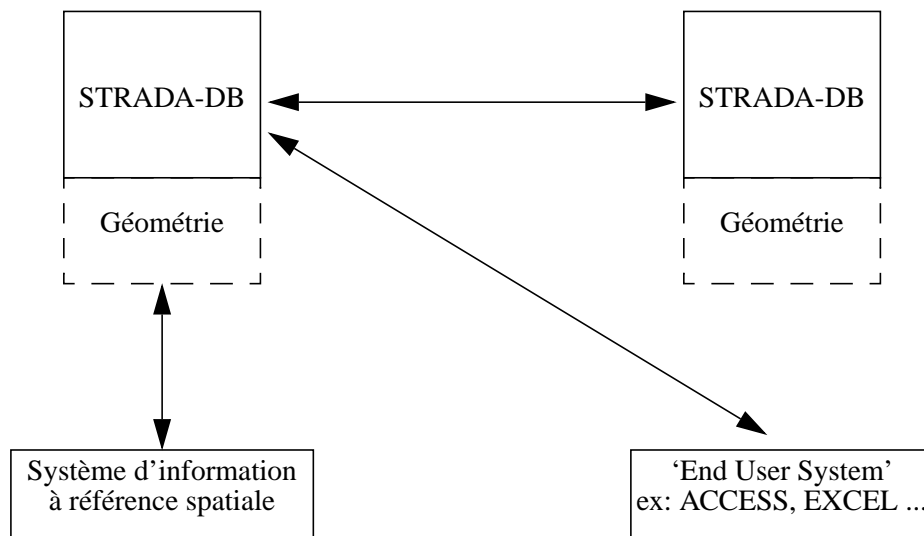


FIGURE 8.

#### Besoins en échange de données de STRADA-DB

L'échange de données standard doit aussi garantir un échange de données géométriques. Bien que la géométrie ne fasse pas partie de STRADA-DB, il faut en tenir compte. Ceci parce qu'il est absolument nécessaire de combiner la géométrie et les données alphanumériques

dans une seule base de données (STRADA-DB), vu les possibilités offertes par les systèmes d'information géographiques.

## 5.2 Analyse de l'existant

### 5.2.1 Road Administration Exchange Format (RADEF)

Définition	Le Road Administration Exchange Format (RADEF) a été établi par les pays de l'Europe de l'ouest et leurs administrations des routes publiques (PRA: Public Road Administrations) pour faciliter l'échange des données des routes principales (highways). Elles sont stockées dans des bases de données des pays divers.
Objectifs	Le but de RADEF est de pouvoir diffuser les données à des utilisateurs comme les administrations locales, régionales et nationales et aux organisations. On sait que les données sont stockées dans différents formats. Pour garantir un échange, une approche commune est indispensable. Vu la diversité des administrations, une centralisation des données n'était pas souhaitée. Le processus de transfert de données décrit par les PRA doit permettre à l'utilisateur d'entrer directement dans le site du PRA, de choisir les données souhaitées avec un outil de requête et de les transférer chez lui au format RADEF. Un système de sécurité doit être mis en place pour que les utilisateurs qui n'ont pas le droit de s'introduire dans le PRA, mais qui sont quand même dans le réseau, n'aient pas d'accès à ces données. RADEF comprend deux parties: Le dictionnaire de données qui décrit la structure des données, et le RADEF Tool qui est l'outil de requête.
Dictionnaire de données	<p>Le dictionnaire de données contiennent les termes, les définitions, les catégories et les relations approuvées par les pays participant à RADEF pour divers types de données. Cette partie est appelée données originales de RADEF. Chaque PRA peut compléter la liste des données en comparant les données avec les leurs dans leurs bases de données. Le PRA doit garantir aux ayant droit l'accès aux données définies dans le dictionnaire.</p> <p>Le dictionnaire de données exige les unités structurales suivantes: La base de données doit contenir une liste de toutes les administrations des routes publiques et une table qui énumère tous les domaines qui regroupent les entités ayant une caractéristique commune. Il doit avoir des tables contenant les entités, leurs codes, leurs descriptions et les domaines auxquels elles appartiennent. Les entités doivent avoir des relations avec les tables de la base de données de la PRA, où les données se trouvent. Dans le dictionnaire sont aussi des tables qui contiennent tous les attributs, leurs codes et leurs descriptions. Dans ces tables, il doit aussi être mentionné si les valeurs d'un attribut viennent d'une liste de codes défini par le RADEF ou si elles résultent de mesures. Dans ce dernier cas, les unités de mesures doivent être annotées muni d'un facteur de</p>

	conversion permettant de calculer les paramètres en unités RADEF définis dans une liste de codes.
RADEF Tool	Chaque PRA doit avoir un RADEF Tool qui agit comme interface entre l'utilisateur qui veut extraire des données et le site du PRA. Dans RADEF Tool il existe une table dans laquelle tous les essais d'extraction des données sont enregistrés. En plus, il enregistre toutes les données sélectionnées, les domaines, les entités et les attributs dans une table. Le RADEF Tool doit aussi permettre de traiter les données sélectionnées. En plus, il doit informer l'utilisateur de la grandeur du fichier dans laquelle se trouvent les résultats des traitements, et il doit donner à l'utilisateur la possibilité d'omettre des données peu importantes.
Reachout	Pour garantir que le RADEF-Tool soit consulté par les utilisateurs ayant vraiment le droit, un système de contrôle de mots de passe doit être installé.
Les domaines	Il existe dix domaines: <ul style="list-style-type: none"><li>• Le réseau des routes: Il définit les routes et un système de repérage curviligne.</li><li>• Les restrictions: Ce domaine décrit les restrictions physiques pour le flux de trafic.</li><li>• Le trafic: Il décrit le flux du trafic mesuré en différentes campagnes.</li><li>• Les structures: Par structure, on entend les ouvrages d'art, comme des ponts ou des tunnels.</li><li>• Les équipements: Ce domaine décrit des éléments ponctuels, comme des panneaux.</li><li>• Les accidents: Ce domaine contient les personnes intervenues dans un accident et leurs blessures, ainsi que les conditions temporelles, l'état de la route et les conditions d'éclairage pendant l'accident.</li><li>• Le domaine des conditions: Il décrit la qualité de la chaussée déterminée par des campagnes de mesures.</li><li>• Géométrie de la route: Il s'agit du profil longitudinal qui ne tient pas compte de la troisième dimension. La description du trajet se limite à une suite de droites et d'arcs de cercle.</li><li>• Domaine de parcours: Il a été introduit pour classifier les routes selon les besoins de PRA.</li><li>• Network Enquiry Domain: Il permet de stocker une vue individuelle des données, c'est-à-dire le résultat d'un traitement spatial.</li></ul>
Système de référence	Dans le RADEF, un système de coordonnées cartésiennes est utilisé pour définir la position des arcs dans le domaine de la géométrie des routes. En fait, ces arcs sont définis par les coordonnées curvilignes du

point de départ et du point final, et les coordonnées cartésiennes d'un point intermédiaire. Les PRA's peuvent utiliser deux systèmes de coordonnées curvilignes, un système équivalent à celui de STRADA-DB (système français) en définissant des points de référence, ou par un système défini par des noeuds (système anglais). Les éléments constitutifs des systèmes curvilignes ne possèdent pas de coordonnées absolues, donc cartésiennes. Par conséquent, un transfert avec RADEF n'est possible que si la base de données cible est alphanumérique. Il n'est pas possible de transférer les données RADEF vers un système d'information géographique comme information géographique. Par information géographique, on entend les données nécessaires pour repérer une entité dans un système de coordonnées absolu. La seule entité qui est repérée dans le système de coordonnées cartésiennes est le point intermédiaire d'un arc. Malheureusement, le point intermédiaire n'a pas de coordonnées curvilignes, avec lesquelles on pourrait déduire la position relative de ce point par rapport aux entités en coordonnées curvilignes, notamment le début et la fin de l'arc à laquelle le point intermédiaire est attaché. Par conséquent, l'arc n'est pas défini du tout. Pour une administration des routes il s'agit finalement d'évaluer l'importance de la position des arcs pour savoir s'il faut introduire un système de référence de coordonnées absolues. En tous les cas, le domaine défini comme expliqué n'est pas utilisable.

### 5.2.2 Geographic Data Format (GDF)

#### Généralités

Le *Geographic Data Format* est un format créé par Philipps et Bosch pour le transfert des données à des systèmes d'information de navigation des automobiles. Ces *Driver Information Systems* peuvent contenir plusieurs fonctionnalités qui déterminent les données nécessaires. Ils peuvent offrir un *gazetteer* qui permet au conducteur de spécifier une localisation par son adresse, un outil de positionnement qui référence le véhicule constamment, une planification des parcours qui permet de trouver le chemin optimal entre deux points, un système de guidance qui donne au conducteur des instructions, une planification des excursions qui fournit les informations souhaitées pour une destination (hôtel, restaurant, etc.), des informations générales comme le temps, les attractions touristiques, etc., et des fonctions cartographiques. Le GDF, qui a été repris par le Centre Européen de Normalisation CEN TC 278, décrit un modèle de données satisfaisant au mieux aux exigences de la navigation automobile.

#### Type de standards

Pour assurer des échanges de données efficaces, il existe deux approches: On peut standardiser le produit final, c'est-à-dire que la forme dans laquelle les données sont stockées sur le media utilisé par le système lui-même par exemple le CD. Cette solution a été choisie par les Japonais qui ont créé un standard basé sur un CD-ROM. L'autre solution consiste à standardiser un produit semi-fini en décrivant les données qui peuvent être contenues dans une base de données routières générales. Il est évident que Philipps et Bosch, travaillant déjà sur leur propre système de navigation, ne pouvaient plus standardiser un produit

	<p>fini. En plus, ils travaillaient avec des processeurs de familles différentes (Intel et Motorola). Initialement, GDF contenait des informations utilisées par les systèmes de navigation de ces deux entreprises.</p>
Historique	<p>A l'époque, il y avait qu'un seul standard qui pouvait être utilisé comme base, le NTF anglais. C'est la raison pour laquelle Philipps et Bosch, travaillant ensemble dans le projet DEMETER (Digital Electronic Mapping of European Territory), ont choisi le NTF comme base. Pourtant, le NTF ne satisfaisait pas tout à fait aux exigences des systèmes de navigation. D'une part le NTF était trop générique, ne définissant que les concepts de 'feature' et 'attribut', d'autre part, NTF ne permettait ni de représenter des relations non-spatiales entre deux entités, ni de spécifier le système de référence. La première version GDF 1.0 est apparue en 1988. Pendant les années suivantes, des nombreux projets européens utilisaient, développaient et validaient GDF. Le format GDF a été repris par le CEN TC 278 en 1994. La version la plus actuelle est GDF 3.0 de 1995, pas encore approuvée par le CEN.</p>
Les volumes	<p>Le GDF contient huit volumes:</p> <ul style="list-style-type: none"><li>• Feature Catalogue (Catalogue des entités)</li><li>• Attribute Catalogue (Catalogue des attributs)</li><li>• Relationship Catalogue (Catalogue des relations)</li><li>• Feature Representation Scheme (Schéma cartographique des entités)</li><li>• Quality Measuring Specification (Spécification de l'estimation de la qualité)</li><li>• Global Data Catalogue (Catalogue des métainformations)</li><li>• Logical Data Structure (Structure du modèle logique)</li><li>• Media Record Specification (Spécification sur la manière d'enregistrer les données)</li></ul>
	<p>Pour la description des données au niveau conceptuel, les quatre premiers volumes sont essentiels.</p>
<i>Feature Catalogue</i>	<p>Dans le Feature Catalogue, on distingue les thèmes d'entités et les classes d'entités, aussi appelées tout court entité. Les thèmes d'entités sont encore génériques et ils font simplement une distinction entre des différentes entités par rapport à une propriété spécifique. On distingue le thème des routes et des ferrys, le thème des unités administratives, le thème des colonies, le thème des bâtiments, le thème des ponts et des tunnels, le thème des routes fluviales et le thème des chemins de fer. Pour la création des entités (classes d'entités), il faut satisfaire les conditions suivantes:</p> <ul style="list-style-type: none"><li>• Le candidat doit être identifié par un nom d'un langage naturel.</li><li>• Le candidat doit avoir un lieu dans l'espace. Il n'est pas possible de transférer des entités alphanumérique, comme par exemple 'mariage'.</li></ul>

- Deux candidats de la même classe d'entités doivent être distincts dans l'espace (Interdiction de chevaucher).
- Le candidat doit occuper un lieu pour un intervalle de temps. Les objets en mouvement, comme une voiture, sont exclus.
- Le candidat doit avoir un propre jeu de types d'attributs. Sinon, on met le candidat dans la classe d'entité ayant les mêmes types d'attributs.

L'identifiant de chaque classe d'entité décrit ci-dessus peut agir comme une clé conceptuelle. Par contre, pour l'échange de données, chaque entité doit porter un identifiant numérique. Ce code est entièrement défini dans la documentation du GDF 3.0.

GDF fait la distinction entre les entités simples et les entités complexes. L'entité simple est composée des primitives géométriques tels que les noeuds, les arrêts et les faces, par exemple *Element de la route*. Les entités complexes se composent de plusieurs entités simples ou eux-mêmes complexes, par exemple l'*Intersection* créée avec les *Elements de la route* et des *Jonctions*.

### Système de repérage

GDF utilise deux systèmes de référence: un système absolu cartésien pour la description des données informatives n'ayant pas de liaison étroite avec les réseaux des routes et un système de coordonnées curvilignes. Le système de coordonnées curvilignes est utilisé pour des données étroitement liées à la géométrie de la route. Les détails de ce système sont décrits ci-dessous.

### Attribut Catalogue

Le *Attribute Catalogue* décrit des types d'attributs, aussi appelés tout simplement attributs, et le domaine dans lequel la valeur d'un attribut doit être définie. Les attributs ne possèdent pas de lieu, ils sont caractéristiques pour une entité et ne constituent pas une relation entre celui-ci avec une autre entité. Les relations sont définies par le *Relationship Catalogue*. Les attributs peuvent être simples ou complexes. Un attribut simple a un seul composant et les attributs complexes en ont plus qu'un seul. Les composants individuels sont appelés sous-attributs. Il n'est pas interdit qu'un sous-attribut soit composé d'autres sous-attributs.

### Les sous-attributs restrictifs

Il existe des attributs, simples ou complexes, qui peuvent être combinés avec des sous-attributs définissant la période de leur validité ou un ensemble d'objets pour lesquels l'attribut est valable. Par exemple, une restriction peut être valable pour les véhicules lourds, mais pas pour les voitures. Sans ce sous-attribut, l'attribut principal est toujours valable.

### Attributs fragmentés

En général, un attribut décrit une entité dans son ensemble. Si l'on veut restreindre la validité de cet attribut à une partie de l'entité on introduit des attributs fragmentés (en anglais: *segmented attributs*). Au lieu de créer plusieurs entités ayant les mêmes valeurs pour tous les attributs sauf l'attribut en question, on ajoute à cet attribut deux autres: *Position de* et *Position à*. Ces positions contiennent une distance curviligne le long de la géométrie d'un élément linéaire. Si les attributs *Position de* et

*Position à* sont vides, l'attribut en question est valable pour toute l'entité.

Les systèmes de navigation sont des systèmes d'information qui essaient de communiquer directement avec le conducteur. Comme l'humain ne peut pas s'orienter à l'aide des coordonnées absolues, l'utilisation d'un système curviligne se justifie. Toutes les distances dans ce système se réfèrent à une des extrémités d'une entité linéaire, c'est-à-dire aux noeuds. Par conséquent, tous les utilisateurs doivent s'entendre à utiliser les mêmes noeuds.

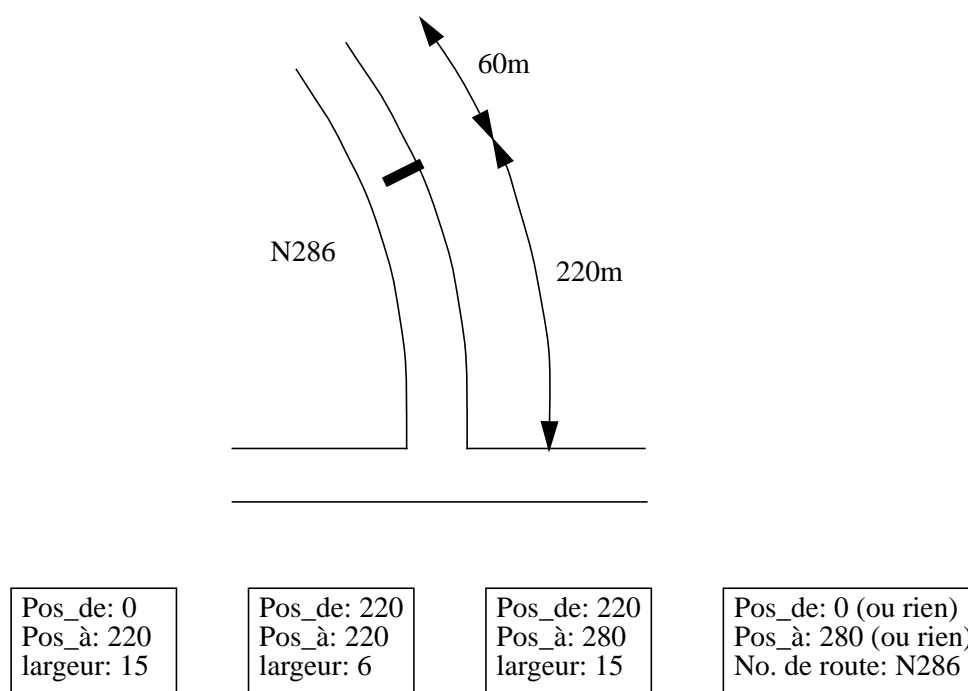


FIGURE 9.

#### Un exemple des attributs fragmentés

Les distances curvimétriques peuvent être données d'une manière absolue ou relative. Une fragmentation absolue met en relation la position avec une extrémité de l'entité à laquelle l'attribut se réfère. La fragmentation relative référence la position par rapport à une extrémité physique d'un ensemble d'entités formant une unité linéaire, par exemple l'extrémité d'une *Route* et non l'extrémité de l'entité *Elément de la route*. La position du noeud étant le 'début' de l'entité linéaire, c'est-à-dire le noeud le plus proche de l'extrémité physique de l'objet (*Route*), doit figurer dans le champ réservé pour le sous-attribut *Chainage Offset*. Pour l'exemple de la *Route*, la *Route* est définie comme l'ensemble des entités linéaires entre deux *Intersections*. Les *Intersections* sont des entités composites comprenant des *Jonctions* et des *Eléments de route*.

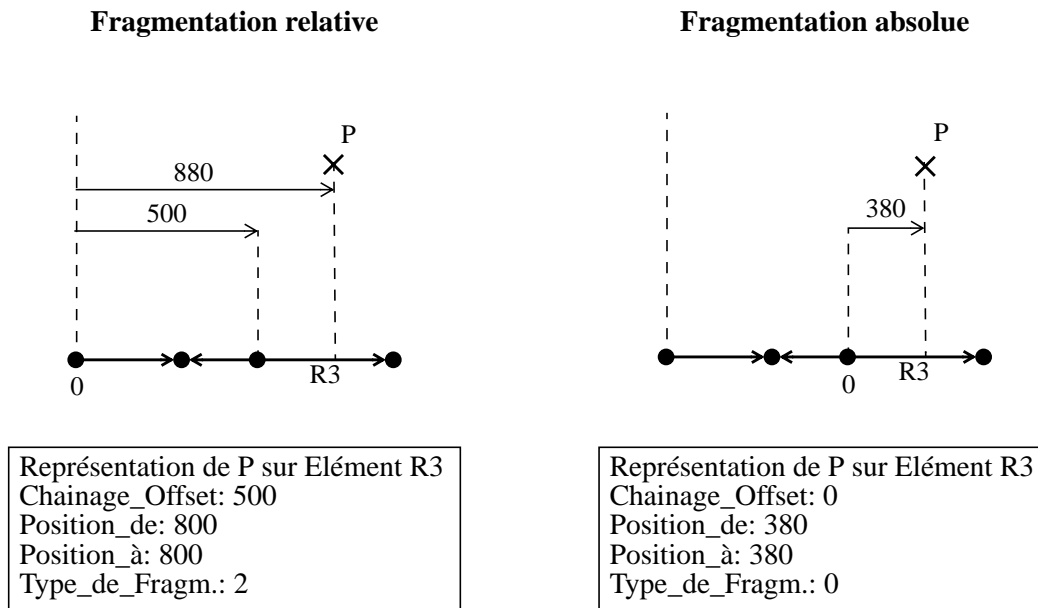


FIGURE 10.

**Différence entre la fragmentation relative et la fragmentation absolue.**

Un sous-attribut supplémentaire est nécessaire pour préciser la méthode utilisée: fragmentation absolue en direction de l'entité (0), fragmentation absolue opposée à la direction de l'entité (1), fragmentation relative dans la direction de l'entité (2) et fragmentation relative opposée à la direction de l'entité (3). Dans GDF, chaque entité linéaire peut avoir sa direction sans tenir compte de la direction des entités précédentes ou suivantes.

*Relationship Catalogue*

Une relation sémantique est une liaison entre deux ou plusieurs entités qui ne sont pas nécessairement de classes différentes. Les relations ayant une structure égale et une signification égale sont regroupées dans des types de relations. La plupart des relations sont binaires, c'est-à-dire que dans une relation deux entités interviennent. Mais il existe des cas où plusieurs entités interviennent, pas exemple la relation *Manoeuvre prohibée* exige les entités *Elément de route*, une *Jonction* et un autre *Elément de route*. L'ordre de ces entités dans la relation est significatif.

On peut distinguer quatre catégories de relations: Il existe des relations en liaison avec le trafic, notamment les manoeuvres interdites ou restreintes. D'autres relations permettent de décrire des liaisons logiques qui ne sont pas équivalentes aux liaisons spatiales, par exemple un bâtiment qui a son entrée vers une autre route que celle qui est la plus proche. Le troisième type de relations a été créé pour décrire quel *Elément de route* passe par dessus de quel *Elément de route*. Cette relation est appelé 'X au-dessus de Y par un brunnel (BRigde et tUNNEL)', où X et Y remplace des *Eléments de route*, *Eléments de route fluviale*, *Eléments de chemin de fer*. La dernière catégorie décrit l'appartenance

d'une entité à une autre, par exemple *Elément de route* à une unité administrative. Beaucoup de relations sont homogènes, c'est-à-dire qu'elles lient deux occurrences du même type d'entité, par exemple les manoeuvres interdites (U-turn prohibited). Ce type de relations est très important dans le GDF, parce qu'elles sont une aide utile pour le conducteur utilisant la navigation automobile.

### *Feature Representation Specification*

Un choix important est de définir la manière de représenter une entité sur une interface graphique. Elle doit être définie par les primitives géographiques: les points, les lignes et les surfaces. On aurait pu intégrer cette spécification dans le Feature Catalogue, mais les concepteurs ne voulaient pas forcément imposer une représentation cartographique. Par conséquent, le transfert d'un jeu de données à une base de données alphanumérique est possible avec le GDF en ignorant cette spécification.

Dans la plupart des cas, toutes les occurrences de la même classe d'entité sont représentées par la même primitive. Mais il y a des exceptions: les *Jonctions* peuvent être considérées comme des points ou des surfaces selon l'échelle.

Les moyens de représentations sont catalogués dans trois niveaux. Le Level 0 contient la géométrie d'une carte en terme de primitives cartographiques: noeuds, arrêts et faces. Tous ces éléments sont représentés sur un plan, on dit que leur graphe est planaire. Le Level 1 contient les primitives géographiques: les points, les lignes et les surfaces. Les éléments de Level 0 obtiennent une signification dans le monde réel. Le Level 1 ne doit plus être planaire puisque il existe de relations tenant compte de la troisième dimension, notamment les *Jonctions* dont les *Eléments de route* sont séparés en hauteur. Dans le Level 2, les entités sont observées à une échelle plus globale. Les détails d'une entité décrits au Level 1 sont simplifiés et des entités composites (Complex feature) sont créées, par exemple l'entité *Intersection* est composée des *Eléments de route* et des *Jonctions*.

### *Quality Description Specification*

Le contrôle de qualité consiste d'une part à détecter tous les défauts du format et à assurer la consistance structurelle et sémantique, d'autre part. Le contrôle structurel se limite à vérifier les partenaires qui interviennent dans une relation, l'existence de ces entités et l'existence des relations nécessaires. Au niveau sémantique, ce sont les valeurs des attributs qui sont contrôlés par rapport à leurs domaines des valeurs admissibles préalablement définis dans la documentation GDF et par rapport à la contrainte *not nul*.

Pour vérifier la cohérence entre les différents niveaux de représentations (Levels), des tests rigoureux sont nécessaires. Ces tests ne sont pas appliqués à toutes les données, mais juste sur un échantillon tiré auparavant. Au level 0, on contrôle la géométrie, c'est-à-dire qu'on compare les éléments géométriques avec la géométrie de l'objet correspondant dans le monde réel, et on affiche toutes les erreurs. Pour les levels 1 et

2, on a besoin d'un test topologique. Le surplus ou la lacune en entités complexes ou simples est testé.

*Global Data Set*

Ce volume définit toutes les métadonnées nécessaires pour clairement définir le transfert et pour éviter des mauvaises interprétations. En outre, on définit le système de référence pour assurer l'indépendance des données par rapport au pays. Ces métainformations doivent permettre la transformation de toutes les coordonnées, transférées à leur propre système de projection et leur propre système de coordonnées.

Critiques

GDF utilise d'une part des coordonnées cartésiennes pour repérer des éléments géométriques d'une manière absolue. D'autre part, GDF permet de transférer des données curvilignes attachées au système absolu par des noeuds à l'aide des attributs fragmentés. L'introduction des attributs fragmentés met en question la complétude de la représentation graphique. Par définition, un attribut n'a pas de lieu, ainsi, il n'est pas représentable géographiquement. La liaison d'un attribut avec une entité peut donner naissance à une visualisation de la valeur d'un attribut. Prenons l'exemple de la figure 9. L'utilisateur veut savoir où sont les *Eléments de routes* qui ont une largeur supérieure à dix mètres. Cette requête sur les entités fournit, entre autres, l'élément N286, bien que cet élément ne satisfasse pas à la condition. Pour améliorer le résultat, on pourrait imaginer une requête sur tous les attributs fragmentés, mais cette méthode se révélerait assez lourde si l'on devait parcourir tous les attributs fragmentés de toutes les entités, surtout pour des traitements nombreux. Ce problème peut être contourné facilement par la création des entités nécessaires. En fait, cette restriction de largeur a lieu à un endroit bien défini dans l'espace, comme le GDF l'exige pour la création d'une entité. Le système de coordonnées pourrait être maintenu en appliquant le repérage similaire à la fragmentation relative.

GDF parle de la direction d'une entité linéaire, par rapport à laquelle la direction de mesures des distances curvimétriques est définie. La direction d'une arrête est définie par une suite des noeuds, la direction d'une ligne est définie par un code précisant si la direction est la même que la direction de l'arrête ou opposée à celle-ci. Il serait mieux de standardiser la direction de mesure en attachant une direction à la *Route*, qui se propage aux *Eléments de la route*.

GDF sépare les entités en trois niveau selon leur complexité cartographique. Les entités de base décrites au niveau 0 sont agrégé et forment de nouvelles entités au niveau 1. Dans le niveau 2, il ne s'agit plus d'une agrégation des entités du niveau 1, mais d'une généralisation de celles-ci. Les entités au niveau 2 ont une géométrie plus simple que les entités au niveau 1. Pour la suite de ces réflexions, on devrait créer encore un autre niveau qui généraliserait les entités décrites au niveau 2, c'est-à-dire une généralisation des objets complexes. L'efficacité de la création d'un autre niveau est très douteux, parce qu'il y a peu d'entités complexes représentant d'autres entités complexes et toutes les autres entités devraient être reprises dans le niveau supérieur. La différentia-

tion entre le niveau 0 et le niveau 1 est assez reconnue dans le monde des systèmes d'information géographiques, puisqu'on veut clairement faire une différence entre les primitives topologiques (équivalent aux primitives cartographiques dans le langage de GDF) et les primitives géographiques. L'introduction d'un niveau 2 peut faire l'objet de plus de discussions, parce qu'une entité complexe peut aussi être vue comme une entité avec sa géométrie généralisée en relation 1:n avec les entités du niveau 1 avec une géométrie plus détaillée, par exemple une *Route* a un ou plusieurs *Eléments de routes*. On aurait l'avantage de pouvoir regrouper les entités plusieurs fois dans un même niveau. Ceci peut aussi être fait au niveau 2 dans GDF, mais l'utilité du niveau 2 serait remis en question, parce que cette relation pourrait aussi être réalisée au niveau 1. On pourrait imaginer d'attacher une échelle à la représentation des entités, c'est-à-dire à la géométrie. Selon l'échelle de l'affichage, on utiliserait la géométrie de la route ou des Eléments de la route.

Conclusion

Il est souhaitable d'abandonner les attributs fragmentés et le niveau 2 dans le schéma de représentation.

### 5.2.3 Comparaison RADEF - GDF pour le transfert de données de STRADA-DB

RADEF et GDF décrivent un format d'échange pur. Les deux méthodes ont été créées pour satisfaire à un échange dans un but particulier. RADEF a été fait pour avoir un standard entre différents administrateurs des routes en Europe, GDF initialement pour avoir un standard dans la navigation automobile. Les échanges se faisant uniquement à l'aide d'un format sont très statiques. Toutes les tables et leurs structures sont définies par ce format. Bien que GDF réserve des codes pour des besoins individuels, l'utilisateur n'a que la possibilité de transférer des types d'information ayant un code. Si l'utilisateur n'a pas besoin de toutes les informations définies par les codes, il n'utilise pas le code correspondant. La comparaison de deux formats peut se faire au niveau du contenu ou au niveau du codage. Dans ce travail on ne compare pas le codage, puisqu'il se situe au niveau physique. Par contre, une comparaison des données transférables avec les deux formats montrera que la définition des données routières est très variable.

Dans ce travail, l'intérêt principal reste STRADA-DB. Est-t-il possible de transférer des données de STRADA-DB avec GDF ou avec RADEF? Si oui, quelles sont ces données?

Eléments STRADA-DB	Ev. RADEF	Ev. GDF	Commentaires
<b>Types d'information</b>			
Entités alphanumériques			
Entités géographiques			

**Echange de données dans le cadre routier**

Éléments STRADA-DB	Ev. RADEF	Ev. GDF	Commentaires
Attributs			
<b>Identification</b>			
OID			RADEF: le pays + le numéro
Identifiant multiple			
Clés conceptuelles			
<b>Éléments constitutifs de SRB</b>			
Axes			
Point de repère			
Point de repère d'aide			
<b>Réseaux d'exploitation</b>			
Éléments de base (Noeud,...)			GDF: dépend du thème
Entités complexes			
Règle de découpage			
<b>Données routières</b>			
Profil longitudinal			GDF: largeur de la route
Profil transversal			
Description de la chaussée			
Accidents			RADEF,GDF: appelé restriction
Chantiers			
Entrave			
Données administratives			
<b>Système de repérage</b>			
Coordonnées cartésiennes			RADEF: que distance longitudinale
Coordonnées curviligne (F)			
Coordonnées curviligne (GB)			
<b>Éléments géométriques</b>			
Point			RADEF: Entités Arc inutilisable
Ligne			
Ligne orientée			
Droite			
Arc			
Clothoïde			
Surface			
<b>Attributs temporels</b>			RADEF: que pour les accidents
<b>Légende:</b>			
		représenté:	entièrement
			partiellement
		représenté:	nul part

TABLE 3.

Evaluation des format RADEF et GDF

Bien que les différents formats soient décrits ci-dessus, il faut faire quelques remarques par catégories de données de STRADA-DB, écrit en gras dans le tableau 3:

- **Types d'information:** GDF ne représente que les entités représentable sur une carte parce qu'un automobiliste n'a ni l'intérêt, ni le temps de parcourir des tables.
- **Identification:** Les deux formats indentifient un objet par un numéro. Comme un numéro n'est pas parlant pour un utilisateur habitué à identifier un objet par son nom, on appelle ce numéro *Objet Identifier* (OID), l'identifiant donné par le système d'information. Théoriquement, ce numéro peut aussi être donné par l'utilisateur, mais comme on ne pourrait plus transférer le vrai OID, ce numéro n'est pas classé dans les clés conceptuelles.
- **Eléments constitutifs de système de repérage de base (SRB):** Le point de repère d'aide n'est que informatif et ne détermine pas directement le SRB. RADEF permet de transférer tous les éléments.
- **Réseaux de d'exploitation:** Les deux formats décrivent divers réseaux. RADEF a plus de flexibilité dans sa structure. On peut définir soi-même un réseau. Dans GDF, les réseaux sont fixés auparavant.
- **Données routières:** Bien que RADEF ait été fait pour les administrateurs des routes, les exigences en données routières de STRADA-DB sont très différentes. Pour les informations communes, la sémantique est différente. Par exemple, les catégories de condition de chaussée sont totalement différentes dans les deux systèmes.
- **On ne peut pas transférer directement le SRB.** RADEF transfère que les distances longitudinales, mais pas l'écart. GDF n'utilise pas du tout le SRB. Par contre une transformation des données SRB dans un système curviligne anglais, même sans passer par le système de coordonnées absolu, permet au moins de transférer les distances curvilignes. Pour le transfert de l'écart, on n'est pas obligé de passer par le système de coordonnées absolu parce que le GDF ne gère pas cette information.
- **Eléments géométriques:** RADEF, décrivant qu'une seule coordonnée, ne peut pas contenir des éléments surfaciques.
- **Attributs temporels:** Les activités et les évènements peuvent être accompagnés par un intervalle de validité. Le transfert des états n'est pas possible.

La table 3 donne une synthèse. Il ne suffit pas de compter les points forts de chaque format et d'en juger quel format est meilleur, parce que certains critères sont fortement corrélées avec d'autres, par exemple les éléments constitutifs sont naturellement décrits dans un format lorsque celui-ci utilise le système de référence français.



## 6.0 INTERLIS-DDL

### Généralités

Un transfert peut se baser uniquement sur un format, comme GDF, RADEF et il existe aussi des transferts basés sur un schéma conceptuel. Les transferts basés sur les formats ont l'avantage d'être créés relativement facilement et rapidement, mais ils ne sont souvent utilisables que pour un système source et un système cible donné. Ces formats se changent souvent en modifiant la liste des données, puisqu'on ne veut pas transférer toujours les mêmes données. Des organisations internationales ont fixé un format standard pour les transferts dans leurs domaines d'activités. Dès qu'un transfert basé sur des formats est standardisé, il devient très statique, c'est-à-dire que les utilisateurs ne peuvent plus ajouter des données au format selon leurs propres besoins. Les transferts, qui se basent sur des schémas décrivant un modèle conceptuel, sont plus ouverts. On définit un formalisme standardisé pour décrire un modèle conceptuel et les règles d'utilisation. Par ce modèle conceptuel, on entend un modèle qui est commun à l'intérieur d'une communauté d'informations. Ces transferts sont dynamiques, parce qu'une communauté peut elle-même définir un modèle. Si l'on définit le modèle une fois pour toute une communauté, le transfert devient statique. Il est parfois souhaitable d'avoir un transfert statique pour garantir l'échange de données surtout pour une grande communauté. Un exemple d'un transfert statique est l'interface de la mensuration officielle (IMO) décrit dans l'Ordonnance Technique de la Mensuration. C'est un exemple d'application d'INTERLIS, contrairement à la fausse interprétation qu'INTERLIS est égale à l'IMO. Le formalisme utilisé dans l'IMO s'appelle INTERLIS-DDL (INTERLIS Data Description Language).

La particularité d'INTERLIS est de pouvoir transmettre des données géométriques et alphanumériques. Il est dit 'objet-relationnel', 'relationnel' ce qui concerne la présentation des entités et des relations, 'objet' ce qui concerne les éléments géométriques. Un modèle relationnel pure ne permet que de représenter des attributs uni-dimensionnels. Pour les données alphanumériques ça suffit parfaitement parce que la conversion dans un modèle relationnel exige seulement le modèle conceptuel soit en troisième forme normale. Mais pour les données géométriques, la réservation d'un seul champ par attribut ne suffit pas parce qu'on ne sait souvent pas le nombre de coordonnées d'un élément géométrique. INTERLIS permet de résoudre ce problème.

### Versions

Actuellement, il n'existe que la version 1.0 de INTERLIS sur le marché. Elle définit l'INTERLIS-DDL et le format de transfert (ITF: INTERLIS Transfer File) qui est dérivé du schéma conceptuel. Elle permet de transférer des jeux de données complets. On s'est rendu compte que la version 1.0 ne satisfaisait pas aux besoins du transfert de données. Une synchronisation de deux bases de données, par exemple, exige de pouvoir faire des transferts d'une partie d'un jeu de données. L'effacement de l'ancienne base de données et l'import de tous les jeux de données de l'autre base de données est une méthode lourde. Par con-

séquent, on a introduit INTERLIS dite incrémentielle. Cette version n'existe qu'en prototype et n'est pas encore en vente. La version incrémentielle ne change rien au niveau du formalisme. Le transfert d'une partie d'un jeu de données doit utiliser le même schéma conceptuel que le transfert de tout le jeu.

#### Éléments de INTERLIS-DDL

Un transfert doit être identifié par le mot clé TRANSFER. Il est suivi par le domaine de définition des valeurs valables globalement, qui est introduit par le mot DOMAIN. INTERLIS-DDL permet même de transférer plusieurs schémas conceptuels dans un seul transfert, elles sont identifiées par le mot clé MODEL. A l'intérieur de chaque schéma, on peut distinguer différentes couches d'informations (TOPIC) qui sont indépendantes entre elles. On a choisi de garantir cette indépendance pour avoir la possibilité de transférer une couche individuelle sans devoir transférer d'autres couches. Dans une couche, un autre domaine de définition des valeurs, local cette fois, peut être créé. Il est suivi par une liste de tables (TABLE), dans lesquelles tous les attributs sont énumérés. Pour entrer dans les détails, veuillez-vous référer à la documentation de base (INTERLIS, Mécanisme d'échange de données pour systèmes d'information du territoire, 1991).

#### Attributs

Une table est identifiée normalement par une clé conceptuelle. L'unicité de la clé conceptuelle est précisé par le mot clé IDENT. Il peut exister des tables sans clé conceptuelle (NO IDENT) contrairement à une entité dans un diagramme entité-association. Mais elles doivent être en liaison 1:1 avec une table possédant une clé conceptuelle, par exemple les tables *Noeud* et *Localisation du noeud*. Une relation est représentée par un attribut, qui pointe sur une autre table préalablement définie. Il s'agit d'une clé étrangère (NomAttribut -> NomTable). Avec cette méthode, on ne peut décrire que des relations 1:1. Un attribut ne peut pointer que sur une seule table. S'il s'agit d'une relation m:m, m étant un entier petit, on peut créer m attributs qui se réfèrent à la même table. Au niveau du format, chaque table est identifiée par un numéro, appelé Transfer ID (TID), valable uniquement pour le transfert. Les valeurs des clés étrangères sont les TID des tables sur lesquelles les clés pointent. On pourrait aussi imaginer utiliser des clés conceptuelles pour faire la relation, mais comme une clé conceptuelle peut être composée de plusieurs attributs, il est beaucoup plus simple d'utiliser le TID. On ne doit réserver qu'un seul champ par clé étrangère.

Un attribut peut être alphanumérique ou géométrique. Un modèle purement relationnel a beaucoup de peine à représenter un attribut multivalué, comme la géométrie. Une polyligne, par exemple, peut contenir plusieurs points, mais, en général, on ne sait pas le nombre de points dans la polyligne lorsqu'on crée un schéma conceptuel. C'est la raison pour laquelle on a inventé INTERLIS-DDL. Dans le schéma, un attribut géométrique est traité comme un autre, mais accompagné par le type de géométrie (ex: POLYLIGNE). Dans le format, une table supplémentaire est créée pour chaque attribut géométrique (sauf le point). Cette table contient toutes les coordonnées des points de la figure et éventuel-

lement des informations supplémentaires pour chaque point (VERTXINFO).

#### Éléments géométriques

Dans INTERLIS-DDL, on peut définir un point par une paire de coordonnées (ou triplet). Dans le format, deux (trois) attributs sont créés qui correspondent au deux (trois) coordonnées. Les coordonnées sont définies par rapport au système de coordonnées nationales. Il s'agit d'un système gauche ayant l'axe Y dans la direction est-ouest et l'axe X dans la direction nord-sud. Par polyligne (POLYLINE) on entend une ligne brisée ou une combinaison de droites et d'arcs. On distingue deux types d'éléments surfaciques: surface (SURFACE) et lot (AREA). La différence principale entre une surface et un lot est la condition topologique supplémentaire pour un lot. Deux lots ne peuvent pas se chevaucher. Il engendre une autre table dans le format, qui porte le nom de la table principale et le nom de l'attribut géométrique. Dans celle-ci, tous les éléments linéaires avec toutes les coordonnées des points d'appui sont énumérés. Pour les lots, cette table précède la table principale dans le format. Les coordonnées du centroïde de l'objet doivent figurer dans la table principale. Le traducteur qui transforme le format ITF en un format propre à un système cible doit avoir la capacité de restituer l'objet en comparant les coordonnées du centroïde avec celles des points d'appui des lignes pour former un lot avec les lignes les plus proches du centroïde formant un polygone fermé. Il est évident que cette procédure fournit des bons résultats que si les lots ne se chevauchent pas. L'avantage est de ne pas enregistrer chaque lignes deux fois, puisque une ligne fait partie de deux lots, sauf les lignes sur le périmètre. La table supplémentaire engendrée par les surfaces succède à sa table principale et toutes les lignes possèdent une clé étrangère pointant chacune sur une occurrence de la table principale.

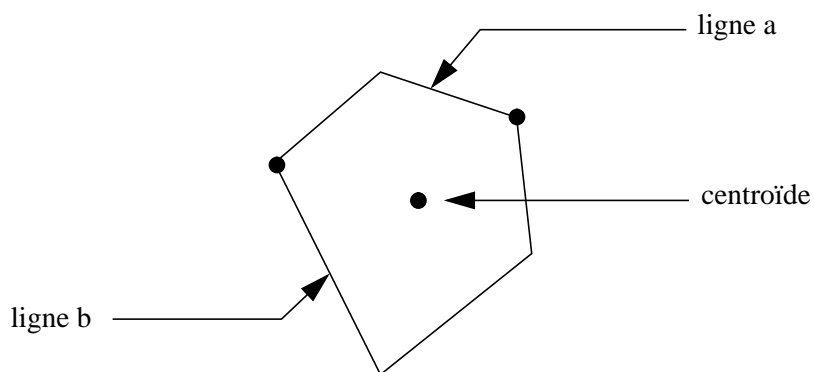


FIGURE 11.

La définition d'un centroïde.

Les polygones, les surfaces et les lots sont souvent accompagnés par le commentaire 'WITHOUT OVERLAPS' et un numéro. Ce commentaire est dû à la définition d'un arc. Une ligne peut contenir des droites, définies par les coordonnées de deux points, et des arcs, définis par les coordonnées de deux points aux extrémités de l'arc et les coordonnées d'un point intermédiaire faisant partie de l'arc. Avec cette définition de l'arc, des imprécisions produisent des fausses liaisons de l'arc avec les droites qui l'engendrent. Un arc, qui est tangent à la droite qui le précède, peut ne plus être tangent après le transfert. C'est une perte d'information due au transfert. Pour éviter des erreurs trop graves, l'utilisateur peut préciser la tolérance de 'chevauchement' des arcs avec ce commentaire.

**Contraintes d'intégrité**

Les contraintes d'intégrité peuvent être mentionnées à l'aide des explications (*// ... //*). L'utilisateur a la liberté totale d'exprimer des contraintes. Le fait de ne pas avoir à formaliser les contraintes peut poser des problèmes pour la traduction des formats ITF à partir de ILI, puisque les notations sont propres à chaque utilisateur.

**Les exploitations et les vues**

Une exploitation (DERIVATIVE) définit la représentation des données, elle peut être une fonction ou une représentation cartographique. Elle n'ajoute pas de nouvelles informations au schéma. Les vues (VIEW) laissent à l'utilisateur la possibilité de changer le format standard proposé par INTERLIS.

**INTERLIS incrémentiel**

La version incrémentielle ne change rien au niveau d'INTERLIS-DDL. Elle n'exige que d'avoir un identifiant global (GID) au lieu d'avoir un identifiant valable uniquement pour le transfert (TID). Un GID est un identifiant qui est stable et univoque. L'identification est univoque si un identifiant est univoque à l'intérieur d'une table dans une communauté d'informations. Elle est stable si l'identifiant ne change pas pendant un cycle de vie d'un objet. L'identifiant ne peut pas être utilisé par un autre objet, même si l'on efface l'objet. Au niveau du format, quelques commandes ont été introduites: 'DELO', pour effacer une occurrence et 'UPDO' pour mettre à jour une table. Le mot clé 'OBJE' pour créer une occurrence est maintenu dans cette version. A première vue, on pourrait s'apercevoir d'une redondance dans les opérations, puisqu'on pourrait remplacer la mise à jour d'une occurrence par une suite de 'DELO' et 'OBJE'. Malgré cela, on a insisté sur le 'UPDO' parce qu'il est extrêmement lourd de contrôler toutes les références extérieures pendant l'effacement et de les reconstituer avec la récréation d'une autre occurrence.

Comme la version incrémentielle ne transfère que l'incrément, il faut savoir dans quelle base de données on veut intégrer l'incrément et l'état de celle-ci. En tenant compte de l'état de cette base de données, on choisit les mots clé 'OBJE', 'DELO' ou 'UPDO'. Ce type d'information est enregistré dans une section (SCNT) dans laquelle on trouve toutes les métainformations comme la version de INTERLIS, le genre de transfert (FULL, INITIAL, UPDO), le nom de l'émetteur, du récepteur, de la

base de données, le nom de l'état initial et final de la base de données. Malheureusement, ces données ne se trouvent que dans le format. Partant du schéma conceptuel, on ne sait pas quelles métadonnées le format exige.



## 7.0 CRITÈRES D'ÉVALUATION / EXIGENCES DES BASES DE DONNÉES ROUTIÈRES

---

### 7.1 Définition des critères d'évaluation

#### Généralités

Si l'on compare les transferts basés sur des formats et les transferts basés sur des schémas, on se rend compte que les critères d'évaluation utilisés pour l'évaluation des formats ne peuvent pas être appliqués, puisque toutes les données peuvent en principe être décrites par un schéma. Il s'agit d'abord de créer une liste de critères généraux qui permettent d'évaluer le formalisme, essentiel pour les schémas, et de les décrire tel qu'aucune autre interprétation ne puisse être faite. Il existe des critères qui ne sont pas essentiels pour tous les types de transferts. Une deuxième tâche consiste à déterminer l'importance de chaque critère pour les activités de transfert possible. Par la suite, une évaluation des formalismes est réalisée.

Nous avons distingué onze catégories de critères:

1. Transfert général
2. Métadonnées
3. Identification
4. Domaines des valeurs
5. Données
6. Type de relation
7. Système de référence
8. Contraintes formalisés
9. Topologie
10. Historique
11. Transfert des méthodes

#### Transfert général

Dans cette catégorie, on met des critères qui concernent tout le mécanisme de transfert.

#### Indépendance des logiciels

Il est important que le transfert soit possible pour n'importe quel logiciel. C'est la raison principale pour laquelle on a introduit des transferts basés sur des schémas. Comme STRADA-DB est un mandat de la Confédération, elle ne doit favoriser aucun logiciel. En plus, elle doit respecter le traitement des données routières des cantons utilisant leurs propres systèmes.

#### Schéma représentatif et simple

D'une part, le schéma doit être compréhensible sans devoir consulter de documentation. Ceci facilite aussi la création des schémas. D'autre part,

	<p>le schéma doit représenter tous les modèles conceptuels. La traduction d'un modèle à un schéma doit conserver la structure.</p>
Formalisme basé sur un standard	<p>Les notions utilisées dans un schéma doivent correspondre à un standard (inter)national pour garantir un échange avec des administrateurs et des utilisateurs non catalogués par le SGE.</p>
Format simple	<p>Un format doit être simple à créer en lisant le schéma. Mais il doit aussi être lu facilement. Le traducteur, qui transforme le format en un autre à la base de données cible, doit effectuer le moins de lecture possible du format pour reconstituer toutes les données et leurs relations.</p>
Transfert en ASCII	<p>Le transfert des données proprement dit doit être fait en format ASCII. Bien que le format ASCII soit plus long que le format binaire, il est préféré parce que le format binaire est relativement compliqué et illisible par des éditeurs.</p>
Transfert dynamique	<p>Le transfert doit permettre d'ajouter des structures et des données. STRADA-DB fournit un noyau commun pour toutes administrations cantonales. Mais les cantons doivent avoir la possibilité de transférer aussi leurs propres données routières.</p>
Verrouillage	<p>Le transfert doit résoudre le problème de verrouillage. Il doit garantir que toutes les opérations dans la base de données originale réalisées par des applications extérieures soient séquentielles, c'est-à-dire une deuxième opération ne peut pas commencer avant que la première soit terminée.</p>
Requêteur	<p>Il est souhaitable de pouvoir sélectionner des données qu'on veut transférer de l'extérieur. Le transfert doit contenir une interface non seulement entre deux systèmes d'information, mais aussi entre le système source et l'utilisateur du système cible. Cette exigence peut être satisfaite par RADEF, parce qu'il permet d'avoir un contact on-line avec le système source.</p>
<b>Métadonnées</b>	<p>Le formalisme et la structure dans le schéma doit permettre de décrire des métadonnées, qui ont un caractère global, dans une section indépendante des données proprement dites. Par global, on entend des données qui sont valables pour tout le transfert. Elle contient, entre autres, l'identifiant du transfert, sa version, la date du transfert, le système de référence géodésique avec toutes les informations nécessaires pour la transformation des données géographiques. En plus, les métadonnées doivent être décrites pour chaque jeu de données, comme la nature des données (originale, copie).</p>
<b>Identification</b>	<p>Par identification, on entend l'identification d'une occurrence par une clé conceptuelle au niveau conceptuel. Mais au niveau logique, toutes les occurrences doivent être identifiées par un numéro.</p>
OID composite	<p>Le transfert de l'OID peut être garanti par pratiquement tous les mécanismes de transfert. Par contre, STRADA-DB possède une OID compo-</p>

	site, dont les deux composants sont le BASE-ID et la Version. On veut aussi avoir la possibilité de transférer l'historique d'un objet.
Clé conceptuelle composite	Le formalisme d'un schéma doit donner la possibilité de former une clé conceptuelle en indiquant les clés partielles.
<b>Domaines des valeurs</b>	Ces critères restreignent le domaine des valeurs qu'un attribut peut avoir. C'est une contrainte sur un attribut.
Domaines de base	On appelle domaine de base tous les domaines qui sont supporté par la plupart des base de données et des langages de programmation. Il s'agit des nombres entiers, des nombres décimaux, des textes et des dates.
Listes de codes	Les listes de codes définis dans STRADA-DB doivent être entièrement transmissibles.
Catalogues de textes	Les catalogues de textes contiennent toutes les informations professionnelles dans STRADA-DB. On n'attend pas du transfert de pouvoir transmettre la gestion des catalogue de textes, mais le contenu des textes composés et le dynamisme des catalogues doivent être transférés.
Structure à l'intérieur d'un attribut	Une date a normalement une structure à l'intérieur, qui précise dans quel ordre il faut insérer l'année, le mois et le jour. Il s'agit de la date utilisée souvent dans la pratique. STRADA-DB définit la date autrement, parce qu'elle s'intéresse aussi à l'heure et à la seconde. En plus, le BASE-ID exige aussi une structure interne, il contient l'identifiant du responsable des clés et le numéro d'objet.
<b>Données</b>	On ne comprend pas les données elles-mêmes, mais plutôt le type de données. Contrairement aux formats purs, la question à se poser est comment on peut représenter un type de données, c'est-à-dire le concept derrière ces données.
Éléments constitutifs du SRB	Il s'agit des éléments qui sont matérialisés sur le terrain, et qui forment les éléments essentiels pour le système de repérage de base (SRB). Ce sont les axes et les points de repère. Les points de repère d'aide ne font pas partie du système de repérage de base puisqu'ils ne sont qu'une aide pour les gens travaillant sur le terrain.
Éléments de base des systèmes d'exploitation	Dans la terminologie de STRADA-DB, il s'agit des noeuds, des tronçons et des groupes de tronçons.
Éléments complexes	Un élément complexe est une entité qui généralise un groupe d'entités. Dans STRADA-DB, il n'y a que des noeuds complexes. Contrairement au format GDF, les super-entités ayant une géométrie composée des géométries des sous-entités ne sont pas complexes.
Entités alphanumériques	Une entité alphanumérique est un objet qui n'est pas fixé à un lieu et qui, par conséquent, n'est pas repérée. Le GDF, par exemple, n'est pas capable de transférer les entités alphanumériques.

Entités géographiques	Une entité géographique occupe un lieu d'une façon durable tel que ça vaut la peine de l'enregistrer comme telle dans une base de données. Elle peut être un point, une ligne ou une surface.
Point	STRADA-DB gère aussi des accidents. Elles ont un lieu bien précis, donc ponctuel.
Lignes	Dans le SRB, on peut définir une ligne en donnant aux points d'appui des coordonnées curvilignes.
Droite	Une droite est définie par des coordonnées des points extrêmes.
Arcs	Par arc, on entend le segment d'un cercle. Dans le métier routier, un segment de cercle doit être tangent aux droites qui l'entourent.
Clothoïde	La clothoïde est un élément principal aux routes. Elle définit le tracé entre une droite et un arc. Elle a été introduite pour offrir aux conducteurs automobiles un confort optimal. Une clothoïde diminue la courbure linéairement de la droite jusqu'à l'arc de cercle.
Surfaces	STRADA-DB peut définir des surfaces trapézoïdales pour décrire l'infrastructure et la superstructure de la chaussée des routes. RADEF, par exemple, ne peut pas décrire une surface parce qu'elle ne gère qu'une seule coordonnée curviligne.
Objets avec plusieurs géométries	Dans deux cas, une entité peut avoir deux objets: Un axe peut se composer de plusieurs segments, qui ne forme pas une ligne continue. Les professionnels des routes aiment travailler avec des géométries de routes dans lesquelles la planimétrie et l'altimétrie sont séparés.
Type de relation	Un transfert ouvert doit permettre de modéliser n'importe quel type de relation (1:1, 1:n, n:n).
Relation homogène	Une relation homogène est une relation qui joint deux occurrences d'une même entité, par exemple la relation <i>Se composer de</i> lie un noeud simple avec un noeud complexe. La notion de 'relation homogène' vient de la terminologie de GDF.
<b>Système de référence</b>	Pour toutes les bases de données routières, il est important de pouvoir utiliser plusieurs systèmes de repérage.
Systèmes curvilignes	Les systèmes français et anglais doivent être représentés par le transfert. Les systèmes curvilignes sont plus importants parce que même un transfert vers une base de données alphanumérique a besoin des informations relatives de la position des entités.
Systèmes cartésiens	L'utilité d'un système absolu ne se révèle que dans un système d'information géographique. Vu les possibilités de traitement et de représentation des données spatiales, STRADA-DB doit se développer vers un système qui combine les données alphanumériques avec des représentations cartographiques.

<b>Contraintes standardisées</b>	Un transfert doit au moins communiquer au destinataire quelles contraintes existent dans les données pour que le destinataire puisse faire les tests correspondants. Il existe des types de contraintes, qui se répètent très fréquemment pour toute sorte de modèle conceptuel à transférer. Il est souhaitable de formaliser les contraintes fréquentes, ce qui facilite aussi l'introduction des tests de cohérence directement dans le traducteur, qui convertit les données exprimées dans un format neutre à un format propriétaire.
Test not null	Cette contrainte figure pratiquement dans chaque base de données. Il existe des attributs mandataires, qui doivent avoir une valeur, et les attributs, qui peuvent avoir aucune valeur.
Unicité	Toutes les clés conceptuelles doivent être uniques.
Inégalités	On doit avoir la possibilité d'exprimer des relations d'inégalité entre des attributs. Dans STRADA-DB, la date du début de validité d'une entité doit toujours être plus petite que la date de la fin de validité.
Contraintes entre deux entités	Il peut exister des attributs dont la valeur dépend de la clé primaire de l'entité dans laquelle ils se trouvent et une autre entité. Ces attributs sont souvent exprimés comme des clés étrangères. Mais, on devrait aussi avoir la possibilité d'exprimer des contraintes d'un attribut qui n'est qu'indirectement influencé par une autre entité. Par exemple, il existe des contraintes entre le <i>Profil géométrique</i> et l' <i>Usage de la chaussée</i> . Une route de trois mètres de largeur ne peut pas avoir plus que deux bandes de roulement.
(x or y)	Dans STRADA-DB, on peut identifier une entité par la clé logique, BASE-ID et la version, ou par la clé conceptuelle. Mais au moins, une clé doit être définie. Il est souhaitable de pouvoir mentionner que les deux clés sont optionnelles, mais que au moins une clé doit être définie. On pourrait aussi imaginer une contrainte qui peut être optionnelle. Si la clé logique prend une valeur, elle doit être unique, mais si elle ne possède pas de valeur, c'est la clé conceptuelle qui doit être unique.
Type de relation	On devrait mentionner le type de relation qu'une entité a avec une autre. Ce critère n'est pas à confondre avec le domaine de critères Type de relation. Celui-ci évalue la possibilité de créer une relation.
<b>Topologie</b>	La topologie des entités graphiques est très importante pour le transfert de STRADA-DB à un système d'information géographique, parce que un défaut de topologie peut être visualisé sur l'écran et on peut l'apercevoir relativement rapidement. Ces défauts peuvent être présents même dans une base de données alphanumérique, mais ils peuvent être éliminés en ajoutant des contraintes entre les attributs concernés. Par contre, un système d'information devrait détecter les défauts topologiques lui-même. C'est pour cela que la contrainte topologique doit être explicitement mentionnée.
Continuité	Tous les segments d'axe sont continus, parce que si une ligne a une interruption, elle se compose automatiquement de deux segments.

Connexité	Si les entités élémentaires continues forment une entité complexe aussi continue, par conséquent les entités élémentaires sont connexes. Cette contrainte n'est pas essentielle pour STRADA-DB parce qu'il n'existe pas d'entité qui exige cette restriction. Une axe de route peut se composer d'éléments de route non connexes.
Absence de recouvrement	Tant que les deux premières propriétés topologiques concernent les éléments linéaires, l'absence de recouvrement restreint les lieux des surfaces. Par absence de recouvrement, on évite que deux surfaces se chevauchent. Dans GDF, toutes les entités possèdent un lieu distinct, par définition. Pour STRADA-DB, cette propriété n'est pas essentielle puisqu'il n'existe pas de contraintes à ce sujet.
Ordre de succession	Dans la navigation automobile, il est essentiel de savoir l'ordre de succession des panneaux ou des éléments de route. L'ordre détermine le parcours. Si l'on veut savoir le chemin le plus court pour livrer des boisons à tous les restaurants par exemple, il ne suffit pas d'avoir une liste des routes qu'on doit prendre, par contre l'ordre de succession des routes est important.
Orientation	L'orientation d'un segment de routes définit la direction dans laquelle on a mesuré les données de repérage dans STRADA-DB.
Graphe plan	En principe, la topologie des éléments de route correspond à un graphe plan. Dans GDF, dès qu'il y a des noeuds fictifs (séparation du niveau de deux routes), le graphe n'est plus plan.
Historique	Le transfert doit permettre d'accompagner un attribut avec des attributs temporels définissant la validité de l'attribut précédent. La détermination temporelle doit couvrir tous les intervalles possibles dans STRADA-DB.
Transfert des méthodes	Il est parfois souhaitable de transférer des contrôles à effectuer d'une manière rigoureuse, mais pour cela il faudrait redéfinir le but du transfert. Est-ce qu'on veut transférer les données avec sa structure ou des programmes?
Interpolation	L'interpolation n'est importante que pour l'exportation des données STRADA-DB aux systèmes d'information géographiques. Une base de données alphanumérique gère les entités ponctuelles, comme le <i>Profil géométrique</i> , comme STRADA-DB. Mais les systèmes d'information géographiques doivent savoir ce qui se passe entre deux enregistrements de <i>Profil géométrique</i> . Il s'agit souvent d'une interpolation linéaire pour des données de STRADA-DB.
IF THEN ELSE	Avec cette commande, reconnue dans le monde des langages de programmation, on pourrait décrire tous les contrôles nécessaires avant l'intégration définitive des données STRADA dans un autre système.

Domaines	Critères	Activités			
		Transfert initial	Transfert d'un jeu de données		
			vers STRADA-DB	vers BD alpha.	vers SIG
<b>Généralités</b>	Indépendance des logiciels				
	Schéma représentatif et simple				
	Format basé sur un standard				
	Format simple				
	Transfert en ASCII				
	Transfert dynamique				
	Verrouillage				
	Requêteur				
<b>Métadonnées</b>					
<b>Identification</b>	Object Identifier (OID) composite				
	Clé conceptuelle composite				
<b>Domaines des valeurs</b>	Domaines de base				
	Listes de codes				
	Catalogues de textes				
<b>Données</b>	Structure interne d'un attribut				
	Eléments constitutifs du système de repérage de base				
	Eléments de base des systèmes d'exploitation				
	Eléments complexes				
	Entités alphanumériques				
	Entités géographiques				
	Point				
	Ligne				
	Droite				
	Arc				
	Clothoïde				
	Surface				
	Objet avec plusieurs géométries				
	Superentités / Subentités				
	<b>Type de relations</b>	1:1			
		0:n			
		1:n			
n:n					
Relations homogènes					
<b>Système de référence</b>	Système curviligne				

Domaines	Critères	Activités			
		Transfert initial	Transfert d'un jeu de données		
			vers STRADA-DB	vers BD alpha.	vers SIG
<b>Contraintes standardisées</b>	Système cartésien				
	Test not nul				
	Unicité				
	Inégalités				
	Contraintes entre entités (X or Y)				
	<b>Topologie</b>	Type de relation			
		Continuité			
		Connexité			
		Absence de recouvrement			
		Ordre de succession			
Orientation					
<b>Historique</b>	Graphe plan				
	Evènements et activités				
<b>Transfert des méthodes</b>	Etats				
	Interpolation				
	IF THEN ELSE				

<b>Légende</b>		important
		souhaitable
		pas important

**TABLE 4.**

**Les exigences au transfert en fonction des activités**

On ne tient compte que des activités sortant de STRADA-DB. On distingue les transferts initiaux, dans lesquels le transfert du schéma conceptuel est très important, et les transferts partiels. Un transfert parfait et standardisé doit recouvrir tous les besoins des activités.

## 7.2 Evaluation d'INTERLIS

Domaines	Critères	INTERLIS	Commentaires	
<b>Généralités</b>	Indépendance des logiciels			
	Schéma représentatif et simple			
	Format basé sur un standard			
	Format simple			
	Transfert en ASCII			
	Transfert dynamique			
	Verrouillage			
	Requêteur			
<b>Métadonnées</b>				
<b>Identification</b>	Object identifier (OID) composite		Compilateurs existants ne le supportent pas	
	Clé conceptuelle composite			
	Domaines de base			
	Listes de codes			
	Catalogues de textes			
	Structure interne d'un attribut			
<b>Données</b>	Eléments constitutifs du système de repérage de base		Relation homogène	
	Eléments de base des systèmes d'exploitation			
	Eléments complexes			
	Entités alphanumériques			
	Entités géographiques			
	Point			
	Ligne			
	Droite			
	Arc			
	Clothoïde			
	Surface			
	Objet avec plusieurs géométries			
	Superentités / Subentités			
	<b>Type de relations</b>	1:1		
		0:n		
		1:n		
n:n				
Relations homogènes				
<b>Système de référence</b>	Système curviligne			
	Système cartésien			

Domaines	Critères	INTERLIS	Commentaires
<b>Contraintes standardisées</b>	Test not nul		attributs pas optionnels
	Unicité		
	Inégalités		
	Contraintes entre entités (X or Y)		
	Type de relation		
<b>Topologie</b>	Continuité		
	Connexité		
	Absence de recouvrement		
	Ordre de succession		
	Orientation		
<b>Historique</b>	Graphe plan		
	Evènements et activités		
<b>Transfert des méthodes</b>	Etats		
	Interpolation		
	IF THEN ELSE		

Légende		
		offert
		faisable
		pas offert

**TABLE 5.**

**Evaluation d'INTERLIS**

Formalisme simple

Le formalisme du schéma conceptuel INTERLIS-DDL est très simple. Avec une trentaine de mots clés, un modèle conceptuel peut être décrit. C'est l'avantage d'INTERLIS-DDL par rapport à d'autres formalismes standardisés. EXPRESS, qui a été favorisé par le CEN dans une évaluation des formalismes pour décrire des modèles conceptuels, a besoin d'approximativement trois cent mots clés. Dans cette évaluation, le CEN n'a évalué que la modélisation au niveau conceptuel, mais pas au niveau logique. C'est-à-dire que seul INTERLIS-DDL faisait partie de cette évaluation. INTERLIS-DDL a été créé par des professionnels dans le domaine des systèmes d'information géographiques, notamment de la mensuration technique. C'est la raison pour laquelle INTERLIS-DDL donne les possibilités de représenter des entités géographiques, leur lieu géographique et leur forme. Ces éléments n'étaient à priori pas représentables avec les langages proposés par le CEN.

Métadonnées

Le transfert des métadonnées est aussi important dans STRADA-DB. Elles contiennent des informations telles que la qualité des données, les responsables de ces données, etc. . INTERLIS-DDL ne permet pas de

représenter le modèle des métadonnées indépendamment des données. On pourrait imaginer mettre ces métainformations dans une couche (TOPIC). Mais si l'on possède des données dans des couches différentes et si l'on met les métadonnées dans une autre, on ne peut pas dire à priori pour quelles couches ces métainformations sont valables. On peut admettre que ces métadonnées pourraient concerner toutes les couches. Mais la structure ne permet pas de la reconnaître comme un schéma conceptuel des métadonnées globales. Dans INTERLIS, toutes les couches doivent être indépendantes. Cette exigence n'est pas satisfaite, puisque toutes les couches dépendent de la couche qui contient les métadonnées. On peut se demander comment représenter des métadonnées qui ne concernent qu'un groupe de données, par exemple une couche. On pourrait le faire comme dans STRADA-DB, on ajoute toutes les métadonnées dans la table correspondantes. Néanmoins, on a l'inconvénient de répéter les métainformations pour chaque occurrence tout en sachant que ces métainformations ne changent pas fréquemment. La solution de GDF semble être la plus raisonnable. On a un groupe de métadonnées qui concerne tout le transfert et toutes les données à transférer (Global Data), dans une entête et pour chaque couche, on définit une section pour les métadonnées locales. Cette approche est à favoriser et on pourrait imaginer l'extension d'INTERLIS-DDL suivante:

**EXEMPLE**

```
TRANSFER Exemple;

DOMAIN
META !! Métadonnées globales
END META.

MODEL Ex1;
META !! Métadonnées valable pour tout le modèle
END META.

TOPIC Sub_Ex
META !! Métadonnées locales valable pour toute la couche
END META.
...
END Sub_ex.
END Ex1.
END Exemple.
```

On pourrait imaginer mettre les métadonnées seulement dans le format ITF. L'expertise sur INTERLIS incrémentiel envisage de mettre quelques métadonnées dans le format, notamment les informations nécessaires pour le transfert incrémentiel. Il est difficilement compréhensible de mettre des informations dans le format sans les ajouter aussi dans le schéma. Le transfert doit être aussi transparent que possible pour l'utilisateur. Ainsi, il devrait connaître les métadonnées et surtout le schéma des métadonnées au niveau conceptuel. Il faut que toutes les métadonnées soient précisées dans le fichier ILI. En plus, il existe souvent des métadonnées organisées en tables avec des relations comme la structure

des données au sens étroit. Par exemple, une table *Référence* qui contient le nom de l'ellipsoïde, la projection etc., qui est liée avec une table *Paramètre*, qui contient tous les paramètres nécessaires pour une transformation éventuelle. Cette 'faute' dans la documentation d'INTERLIS incrémentiel est due à l'intention de ne pas changer le formalisme INTERLIS-DDL à tout prix. Un changement dans le formalisme aurait engendré un changement de l'interface de la mensuration officielle (IMO) qui est ancrée dans une ordonnance approuvée. La modification d'une ordonnance produit une procédure juridique relativement longue, qui est à éviter. Une loi n'est pas censée être modifiée constamment. Il est redoutable que l'utilisation d'un formalisme et d'un schéma conceptuel (IMO) soit imposée aux professionnels au moyen d'une ordonnance. Le formalisme doit toujours être modifiable, puisqu'il doit être adapté aux besoins des utilisateurs. Comme l'informatique évolue constamment, les besoins en transfert se modifient avec la même rapidité. Une ordonnance, par contre, a un caractère statique et doit être durable.

Pour un échange standardisé, les bases géodésiques doivent être indiquées. Ces informations font partie des métainformations globales d'un échange. Bien que STRADA-DB n'utilise le système de coordonnées cartésiennes que pour des attributs optionnels, elle doit avoir des spécifications sur la référence. Pour les échanges à l'intérieur de la communauté d'information de STRADA-DB, c'est-à-dire les administrations fédérales et cantonales et les utilisateurs reconnus comme tels par les administrations, l'information de la référence n'est rarement utilisée. Mais si l'on veut diffuser les données de STRADA-DB à l'extérieur de la communauté, des transformations de la référence sont nécessaires. L'IMO n'exige pas d'information sur le système de référence car les informations de la mensuration officielle ne sont pas à diffuser hors de la Suisse et, de plus, tous les géomètres s'entendaient sur la référence géodésique à l'époque. Entre temps, l'introduction des nouvelles techniques de levés topographiques, notamment le GPS (Global Positioning System), a mis en question la référence géodésique reconnue dans la mensuration géodésique. La coexistence de deux systèmes de référence a engendré énormément de discussions entre les professionnels. Néanmoins, l'interface ne doit pas influencer la décision ou favoriser un système.

#### Structure hiérarchique

INTERLIS ne supporte que des relations 1:1 et 1:n. Il exige une structure hiérarchique. L'ordre des tables dans le schéma conceptuel est essentiel. Les entités indépendantes doivent figurer avant les tables dépendantes de celles-ci. On indique les tables mères avant les tables de détail. Cette ordre est logique, parce que pendant l'intégration d'un jeu de données dans une base de données mère les relations doivent être contrôlées. Soit on met les identifiants des tables mères en mémoire et en lisant les tables de détail, on contrôle directement l'existence de la table mère. Soit on met en mémoire toutes les tables de détail et en lisant toutes les tables mères on contrôle l'existence de toutes les tables de détail. Il est clair que la première possibilité est à favoriser parce qu'il y a beaucoup plus de tables de détail que des tables mères à mettre

en mémoire. Cette contrainte dans le schéma n'est pas essentielle pour STRADA-DB, puisqu'on a évité des relations n:n. L'avantage d'avoir une structure hiérarchique est de ne pas devoir indiquer le type de relation dans le sens inverse. Par exemple, un point de repère est toujours lié à exactement une axe. Il n'est pas nécessaire de mentionner la liaison que l'axe possède avec les points de repère.

#### Géométrie

INTERLIS permet de transférer la géométrie des objets spatiaux. Bien que STRADA-DB ne contienne pas des éléments descriptifs de forme d'un objet spatial, il est nécessaire de couvrir cet aspect comme il est décrit dans le chapitre 'Echange de données dans le cadre routier'. INTERLIS-DDL peut décrire des points, des lignes, des arcs, des polygones, des surfaces et des lots. Le point n'est pas défini comme point, mais comme paire de coordonnées, ou triplet de coordonnées. Les autres entités n'ayant que des coordonnées sont les centroïdes des lots. Dans le format, les points ne sont pas tout de suite reconnaissables. Il faut contrôler si la table a une géométrie surfacique ou ponctuelle dans le schéma, ou si la table est précédée par une table qui contient des polygones. Les occurrences d'une table doivent être de même type soit des points, des polygones, des lots ou des surfaces, sauf si l'on définit deux attributs géométriques qui décrivent deux types de géométries.

Dans le domaine routier, il est préférable de pouvoir représenter plusieurs géométries par objet. Les axes de STRADA-DB peuvent se composer de plusieurs segments. Dans INTERLIS, on peut facilement créer une table *Axe Segment* qui ne contient que les descriptions géométriques sans attributs alphanumériques. Elle est liée avec sa table mère *Axe*, qui contient les informations alphanumériques. Mais il existe dans la conception des routes le souhait de pouvoir séparer la planimétrie et l'altimétrie. La planimétrie est définie par des coordonnées cartésiennes X,Y et l'altimétrie par la distance sur la route et l'altitude. Il faut se poser la question si l'on crée deux attributs géométriques, un pour la planimétrie et l'autre pour l'altitude avec des domaines de valeurs différents ou si l'on transfère tous les points utilisés pour la définition de la planimétrie et de l'altimétrie en coordonnées cartésiennes comme seul attribut géométrique. En fait, on peut considérer la distance curviligne comme le résultat d'un traitement des coordonnées cartésiennes. Avec la deuxième méthode on diminue la quantité d'information à transférer et, en plus, on peut se baser sur des traducteurs existants sur le marché. Cependant il y a deux inconvénients. Le système source doit avoir une méthode de jointure des géométries et le système cible doit posséder l'algorithme de séparation des géométries.

#### Système de repérage

Pour STRADA-DB, la question posée ci-dessus n'est pas un problème, puisque dans STRADA-DB le repérage est fait en principe à l'aide des coordonnées curvilignes. Mais dès qu'on intègre la géométrie des axes, est-ce que cela nous dispense de transférer des distances curvilignes? Non, parce que toutes les données ayant une distance curviligne appartiennent à un axe. Par exemple, un point à la proximité d'un carrefour n'ayant que des coordonnées cartésiennes ne permet pas de trouver

l'axe auquel ce point appartient. Néanmoins, lorsqu'on a les coordonnées cartésiennes des points de repère et la géométrie des axes, est-ce qu'on veut encore transférer les distances curvilignes des entités spatiales, ou est-ce qu'on veut les transformer d'abord en coordonnées cartésiennes? Pour transférer les distances curvilignes, on a besoin de l'axe, du point de repère, de la distance et de l'écart à l'axe. Le transfert en coordonnées cartésiennes exige, de plus, l'axe, deux (ou trois) coordonnées et la méthode de transformation s'il existe un besoin. Ce besoin n'existe que pour l'intégration des données dans STRADA-DB. Avec le transfert en coordonnées cartésiennes des entités spatiales on peut utiliser des traducteurs déjà sur le marché. Le problème se pose au niveau pratique, les gens travaillant sur le terrain et levant des points sur une route, des distances curvilignes et des écarts, sont intéressés à pouvoir retrouver ces points plus tard. Actuellement, on a des géométries des axes digitalisés sur des plans 1:10000 à disposition. Ces axes diffèrent fortement de la géométrie de la réalité. Si l'on fait maintenant un calcul de la distance curviligne et de l'écart à partir de cette géométrie, on va jamais retrouver le bon point sur le terrain. Surtout l'écart à l'axe va être fortement faussé. Il est donc indispensable de garantir le transfert des coordonnées curvilignes, même si l'on avait la géométrie des axes. On peut imaginer transférer la géométrie de représentation comme des attributs géométriques et les coordonnées curvilignes comme des attributs alphanumériques. Cette solution fonctionne parce que les attributs contenant des coordonnées curvilignes ne sont pas tellement multivalués, puisque le nombre de ces attributs dépend uniquement de l'entité et pas de l'occurrence.

**Format**

Le format doit être dérivé directement du schéma conceptuel par un traducteur. Le format doit être simple et facile à créer. Pour un transfert complet d'un système à l'autre il faut avoir deux traducteurs. Un traducteur qui crée le format ITF à partir des données du système source et un autre qui transforme le format ITF dans le format propre au système cible. Les traducteurs doivent être simples à créer et les traductions doivent avoir une durée minimale. La durée de la traduction dépend fortement du nombre de données à transférer. C'est pour cela qu'une lecture multiple du fichier de source et du fichier ILI est à éviter. Le format ITF ne permet pas de lire les fichiers en une seule fois. L'utilisateur a un choix relativement large pour la création du format. A l'aide des *Views*, il peut lui-même créer l'ordre des données et la manière dont ces données sont à énumérer dans le fichier ITF. Malheureusement, ces données essentielles pour le format ITF se trouve à la fin du fichier ILI. Un traducteur doit lire au moins une fois presque tout le fichier pour savoir comment il doit créer ou lire le fichier ITF. Les *Views* ne favorisent pas l'introduction d'un traducteur unique pour tous les fichiers ITF. En plus, l'utilisateur peut redéfinir les signes par défaut (par exemple les espaces) comme la documentation d'INTERLIS le propose. Ces signes spéciaux figurent à la fin du fichier ITF. Il est incontournable de lire au moins une fois tout le fichier ITF pour savoir comment les signes sont encodés. Pour un fichier ITF de grande taille, la traduction du fichier à un format propre d'un système d'information peut avoir une durée

assez significative. Ces informations devraient figurer dans une entête du fichier définissant le format.

Dans le schéma conceptuel, l'utilisateur peut exprimer toutes les contraintes de n'importe quelle manière à l'aide des explications (//...//). Il est absolument impossible de pouvoir traduire toutes ces contraintes parce qu'elles ne sont pas indiquées avec un formalisme nécessaire.

#### La géométrie dans le ITF

L'attribut géométrique engendre une paire de coordonnées (ou un triplet de coordonnées) s'il s'agit d'un point, ou une table avec une liste de coordonnées s'il s'agit d'une polyligne, d'une surface ou d'un lots. Par contre, cette table précède la table principale s'il s'agit d'un lot et elle suit la table principale s'il s'agit d'une surface. Par conséquent, le traducteur, qui traduit le fichier ITF, doit lire jusqu'à l'attribut géométrique dans le fichier ILI pour savoir si la table qui suit dans le fichier ITF contient des informations alphanumériques ou géographiques. Ceci implique que le traducteur doit mettre en mémoire toutes les tables avec leurs types de géométrie pendant une lecture préalable. En plus, le traducteur doit comparer les coordonnées du centroïde avec celles des points d'appui des polygones pour former des lots. Ceci nécessite un algorithme relativement compliqué et pas facile à programmer. La définition des lots empêche de pouvoir produire un traducteur en quelques jours. On n'a pas vraiment d'intérêt à investir plus de temps pour faire ces traducteurs. On risque qu'INTERLIS ne soit pas utilisé et que les intervenants du transfert choisissent des formats associés aux logiciels de système d'information. L'avantage du concept des lots est de ne pas devoir indiquer toutes les lignes deux fois, c'est-à-dire une fois par entité. Pour le transfert des données routières, les lots ne seront pas utilisés, STRADA-DB, GDF et RADEF ne posent aucune contrainte topologique pour les éléments surfaciques.

INTERLIS permet de transférer des droites et des arcs. Les professionnels ont un intérêt à transférer, aussi des clothoïdes. INTERLIS envisage le transfert de cette forme particulière de ligne dans la version 2. Mais il faut tenir compte que cette géométrie est particulière aux professionnels routiers et la plupart des systèmes d'information ne peuvent gérer que des droites et des arcs. Pour le tracé des routes, il est important que les droites soient tangentes à l'arc qui l'entourent. Il existait beaucoup de discussions sur la définition de l'arc d'INTERLIS. Cet arc est défini par deux points aux extrémités de l'arc et un point intermédiaire. En indiquant les coordonnées du point intermédiaire avec autant de chiffres significatifs que celles des points aux extrémités, l'erreur d'arrondi engendre que les droites voisines ne sont plus tangentes à l'arc. Bien qu'il s'agisse d'une petite erreur, le transfert n'est pas sans perte. Si l'on tient compte de la précision des points, qui résulte d'une levé terrestre (pour la mensuration officiel l'ordre du centimètre), l'erreur d'arrondi semble ridicule. Mettant ce problème dans le cadre des systèmes d'information géographique. Les professionnels des systèmes d'information géographique prétendaient toujours qu'un des atouts de ces systèmes était de pouvoir afficher une zone à n'importe

quelle échelle. L'utilisateur doit décider si c'est raisonnable ou pas. Maintenant, il ne faut pas dire aux utilisateurs des systèmes d'information de ne pas agrandir la zone jusqu'à une échelle que l'erreur devienne visible. Heureusement, il existe la possibilité de mettre la tolérance à zéro. (WITHOUT OVERLAPS >0).

Transfert des méthodes

Bien qu'INTERLIS-DDL soit un langage de description des modèles conceptuels et non pas un langage de programmation, on se pose souvent la question si l'on ne veut pas ajouter quelques mots clés pour intégrer aussi les contrôles de cohérence. Pour STRADA-DB, un IF THEN ELSE et un FOR i:=1 TO n DO permettraient de représenter tous les contrôles nécessaires. Mais le schéma serait alourdi énormément si l'on voulait représenter toutes les méthodes. A notre avis, INTERLIS-DDL doit rester un schéma conceptuel seulement pour décrire la structure des données, non pas pour transférer des concepts de traitements, ni des contrôles. Par contre, on pourrait imaginer un autre fichier contenant ces méthodes et pouvant aussi être transférer, ensemble avec le fichier ILI et le fichier ITF.

Verrouillage

INTERLIS ne traite pas le problème de verrouillage. Ce problème est à résoudre plutôt au niveau organisationnel.

Requêteur

Pour STRADA-DB, un outil de traitement est déjà intégré dans la base de données, mais pas accessible par des tiers personnes. STRADA-DB n'a pas d'intérêt à fournir aux utilisateurs un outil pour qu'ils puissent sélectionner les données à transférer, comme le RADEF le propose. La mise à jour de tous les systèmes des utilisateurs, qui est réglée par l'administrateur, se révélerait très compliquée parce que l'administrateur ne saurait pas quel utilisateur mettre à jour et avec quelle fréquence.

## 8.0 DESCRIPTION DES PARTICULARITÉS DE STRADA-DB EN INTERLIS-DDL

### OID

Dans STRADA-DB, toutes les tables sont clairement identifiées par le BASE-ID et le numéro de version. Cette clé primaire est unique dans toute la Suisse. Il s'agit d'une clé univoque et stable selon l'expertise INTERLIS incrémentiel. Dans le schéma conceptuel d'INTERLIS le objet identifier (OID) n'est pas mentionné, parce qu'il s'agit d'un chiffre peu parlant donné par un système informatique. L'OID se trouve plutôt au niveau logique et pas au niveau conceptuel. Le compilateur INTERLIS qui produit un fichier descriptif du format, ajoute l'OID dans la liste des attributs. Bien que la version actuelle 1.0 d'INTERLIS n'exige qu'un OID univoque pour le transfert, les administrateurs de STRADA-DB ont intérêt à satisfaire les exigences pour le transfert incrémentiel, puisque l'effort à fournir est minimal.

Il existe trois solutions:

1. On ajoute le BASE-ID et le numéro de version comme attribut dans les tables. Par contre, on doit redéfinir l'OID, soit des OID's seulement valable pour le transfert (TID), soit on crée un Global Identifier (GID) en additionnant les deux clés partielles (par exemple: NE..00000000013212.1), ou on crée une opération mathématique qui transforme le BASE-ID et le numéro de version à une clé unique (par exemple: 1000 fois le numéro du BASE-ID plus le numéro de version). Cette dernière méthode porte un grand danger. Il faut absolument connaître le domaine de valeur que peut avoir le BASE-ID pour que l'opération mathématique ne donne pas un même résultat pour deux objets différents. L'inconvénient est le nombre d'attributs qui sert uniquement pour l'identification d'une entité: le BASE-ID, la version, les différentes parties de la clé conceptuelle et encore le GID.
2. On définit l'OID uniquement par le BASE-ID et on traite le numéro de version comme un attribut. Ce OID n'est pas global, parce qu'il peut avoir plusieurs versions d'un objet. On doit se contenter de ne pas transférer l'historique d'un objet.
3. On peut définir un OID composé de BASE-ID et le numéro de version avec chaque partie occupant un champ. L'inconvénient majeur serait de ne pas pouvoir utiliser les traducteurs sur le marché. INTERLIS-DDL lui-même ne précise aucune contrainte vis-à-vis la création d'un GID. La schéma serait dénormalisé parce qu'il existe des clés étrangères qui pointent que sur une partie de l'OID.

La deuxième méthode est préférable si l'on admet que les transferts de tout un historique des objets soient assez rares.

Bien que le schéma décrive les données au niveau conceptuel, il serait bien d'ajouter déjà la clé logique (OID) comme le premier attribut. Il serait présent à titre informatif et l'utilisateur ne serait pas surpris s'il s'insère soudainement un champ supplémentaire.

Listes de codes

La liste de code peut être reprise exactement comme telle dans le schéma conceptuel. On ajoute dans la section DOMAIN une énumération définissant toutes les possibilités de valeurs qu'on peut avoir pour un groupe d'attributs. INTERLIS-DDL n'exigerait pas de codage de toutes les valeurs. On pourrait bien définir une liste avec toutes les valeurs vrais, par exemple Langues (Allemand, Français, Italien). Mais comme STRADA-DB veut être indépendant de toutes les langues, la codification s'impose, par exemple Langues (D, F, I).

Catalogues de textes

Les catalogues de textes peuvent être traités comme les listes de codes, en principe, puisque les catalogues de textes peuvent être considérés comme des listes structurées.

On peut imaginer les scénarios suivants:

1. On ne transfère que les codes de tous les textes composés. Elles sont traitées exactement comme des listes de codes. Cette solution est pratiquement exclue, parce que la grandeur de ces listes de codes serait énorme.
2. On divise les attributs contenant des textes composés en plusieurs attributs. Les noms des colonnes agissent comme des noms des domaines dans lesquelles se trouvent les textes élémentaires, par exemple pour l'attribut *Fonction* de la table *Axe* on a le domaine *Type d'axe.Fonction* = (axe de maintenance, axe de projet, etc.).
3. On traite les textes composés comme des textes libres sans contrainte.
4. On crée des tables dans la même couche que les données, qui contiennent tous les textes composés utilisés avec leurs identifiants.

A notre avis, le transfert de tous les codes et de tous les textes composés pour chaque transfert alourdirait énormément le mécanisme de transfert. Par contre, on peut imaginer décrire tout le modèle de gestion des catalogues de texte avec INTERLIS-DDL dans une couche indépendante (TOPIC). Pour le premier transfert de STRADA-DB vers un nouveau système jamais mis en contact avec des données de STRADA-DB, on transfère au minimum les codes et les textes composés et on laisse toutes les autres tables dans cette couche vide. Si l'on veut conférer la gestion des catalogues de textes à un système externe, on utilise toutes les tables de la couche des catalogues de textes, tout en assurant que ce nouveau système est capable de gérer les catalogues de textes. Pour les transferts des données vraies, on ne doit que transmettre des codes et on ajoute dans une section des métadonnées les noms des catalogues de textes et leurs versions auxquels les codes se réfèrent.

Une autre possibilité est de transférer seulement les textes composés utilisés dans le jeu de données à transférer dans la même couche que toutes les autres données dans une table. Elle doit être mentionnée la première dans le schéma parce qu'elle doit être considérée comme une superentité de toutes les autres tables. Cette dernière solution est la seule qui tient compte de la dynamique des catalogues de textes. Le transfert préalable des catalogues de texte complets exige une mise à jour fréquente de ces catalogues, soit en remplaçant tout le paquet des catalogues de textes, soit en ne transmettant que les modifications.

La solution (2) aurait un intérêt si les catalogues de textes étaient plus statique. Bien que le schéma conceptuel créé pour le transfert ne doive pas changer le modèle conceptuel de la base de données, il faut réévaluer l'importance de la dynamique des catalogues de textes. Prenons l'exemple de la page 13, le catalogue du type de point de repère. Au niveau conceptuel, toutes les colonnes constituent un propre attribut: un attribut *Matériau de PR*, un attribut *Emplacement du PR*, un attribut *Fonction primaire du PR* et un attribut *Fonds dans la table PR*. Dans INTERLIS-DDL, ces attributs seraient représentés comme des attributs optionnels, puisqu'ils ne peuvent contenir aucune valeur. Par contre, l'ordre de ces attributs dans la table serait fixe. On ne peut plus changer l'ordre en fonction de la langue. On ne peut plus créer des phrases satisfaisant la grammaire de cette langue, par exemple 'plaque de plastique collée' ou 'geklebte Plastikscheibe'. On échange les colonnes *Matériaux* et *Type de fixation* selon la langue. En plus, on n'a plus la liberté de choisir plusieurs valeurs d'une même colonne, par exemple on ne pourrait plus prendre 'Voiture' et 'Piéton' de la colonne *Intervenant* du catalogue *Type d'accident*. Mais on peut bien contourner ce problème en ajoutant toutes les combinaisons possibles et réalistes dans le domaine de l'attribut *Intervenant de l'accident*.

#### Géométrie

L'intégration de la géométrie des axes et les différents systèmes de repérage dans un schéma conceptuel est suffisamment discutée dans le chapitre 'Critères d'évaluation/Exigences des bases de données routières'.



## 9.0 TEST D'INTERLIS

### 9.1 Généralités

On a choisi de tester l'utilisation d'INTERLIS pour un transfert de données de STRADA-DB à partir d'un fichier texte vers EXCEL. Bien qu'EXCEL ne soit pas une base de donnée, le transfert peut être quand même utile pour faire des calculs de la surface des couches par exemple. En plus, il peut aussi exister des utilisateurs qui n'ont pas une base de donnée à disposition et qui aimerait profiter de la diffusion des données de STRADA-DB. Pour ce test on a choisi EXCEL principalement parce qu'EXCEL ne possède pas vraiment un format propre. On peut insérer un fichier texte facilement et EXCEL va créer des colonnes lui-même selon le choix de séparateur (espaces, tabulations, etc.) des valeurs dans le fichier texte.

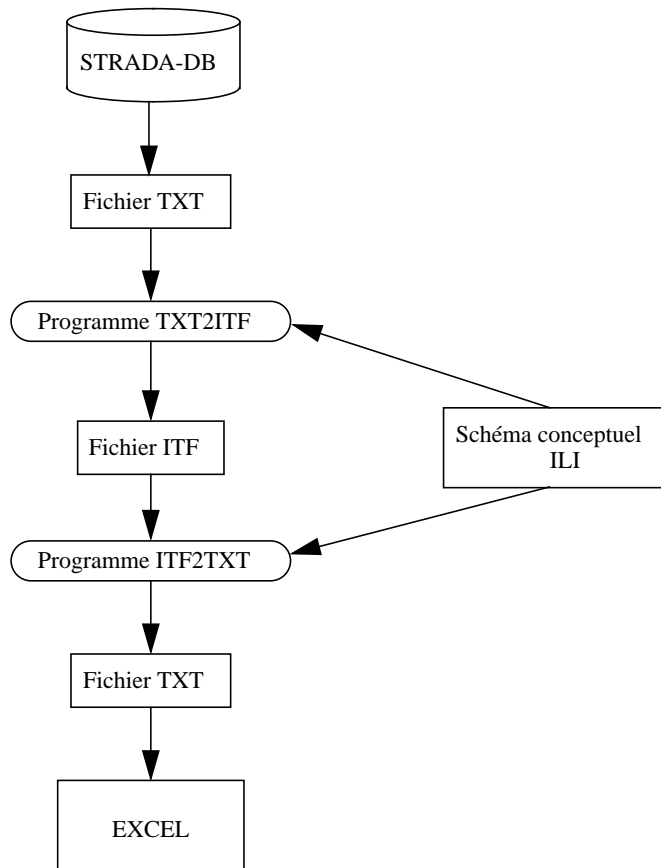


FIGURE 12.

Concept du test de INTERLIS

Avant d'écrire des traducteurs, il est nécessaire d'élaborer un schéma conceptuel pour le transfert. Etant donné que nous avons choisi d'avoir comme système cible un logiciel de calcul et non pas une base de données appuyé sur un autre modèle conceptuel, il s'agit uniquement de formaliser le modèle conceptuel de STRADA-DB avec INTERLIS-DDL. La démarche méthodologique se complique fortement s'il n'existe pas un modèle claire, représenté en diagramme Entité-Association. S'il existe, on crée directement des arbres hiérarchiques (Voir Annexe 1). Cette représentation sert à déterminer l'ordre des tables dans le fichier ILI. Elle permet d'attacher des niveaux de dépendances aux tables. Les tables indépendantes se trouvent au niveau zéro. Les tables qui dépendent directement d'une table indépendante sont classées au niveau un. Une table prend toujours la valeur supérieure du niveau de la table mère. Dans le fichier ILI, les tables de niveau inférieur sont présentés avant les autres. Pour avoir un groupement thématique dans le schéma, on parcourt une arborescence en descendant jusqu'à ce qu'on rencontre une table qui dépend d'une autre ne faisant pas partie de l'arbre. Et on commence à parcourir le nouvel arbre jusqu'à la table en question puis on continue par le premier arbre.

Théoriquement, on devrait extraire les données souhaitées de la base de données ORACLE directement dans un fichier txt, mais pour des raisons organisationnelles, nous avons eu des données STRADA-DB à disposition sur ACCESS. Nous avons sélectionné des données typiques de STRADA-DB, comme l'axe, les points de repère etc., pour restreindre la quantité de données à transférer. Dans le fichier txt, résultat de cette sélection, les occurrences sont arrangées par lignes (Voir ANNEXE 3). Bien qu'un fichier, qui possède un attribut par ligne, soit plus facile à traduire, ce fichier serait pénible à lire par l'éditeur. Dans ce fichier, les valeurs des différents attributs sont séparées par des tabulations. Chaque entité est introduite par les noms des attributs, énumérés en lignes. Le nom du premier attribut identifie l'entité.

Le travail consiste à faire un traducteur qui convertit ce fichier txt au format ITF d'INTERLIS en se référant au schéma conceptuel ILI. Ce schéma doit aussi être utilisé pour la lecture du fichier ITF et pour le transformer dans une forme telle que le fichier txt résultant puisse facilement être importé sur EXCEL. Cette transformation doit être réalisée par un deuxième traducteur ITF2XLS.

Pour faciliter la tâche, nous avons fixé quelques hypothèses:

- L'ordre des entités et l'ordre des attributs correspondent déjà à l'ordre fixé dans le schéma conceptuel.
- Les attributs correspondent aux attributs décrits dans le schéma.
- Aucun contrôle de cohérence n'est fait au niveau des traducteurs. Le schéma conceptuel transfère quelques contraintes qui sont à contrôler, mais ces contrôles peuvent bien être faits par de tierces applications.

Dans le fichier de sortie (Voir ANNEXE 5), chaque entité est identifiée avec les noms utilisés dans le schéma. L'identifiant est suivi par une ligne qui contient les noms des attributs. Ensuite, toutes les occurrences sont enregistrées en lignes. Les valeurs sont séparées par des espaces, puisque les espaces initiaux dans le fichier de départ sont représentés par des traits soulignés '\_' selon la configuration par défaut de ITF.

L'import final dans EXCEL se fait à partir de ce logiciel. Il faut tenir compte du fait que le nom du fichier porte l'extension 'txt' pour qu'EXCEL le reconnaisse comme un fichier texte.

Validation du concept

L'intérêt des professionnels des routes d'importer des données STRADA-DB en EXCEL est de pouvoir calculer des propriétés ne faisant pas partie du dictionnaire de STRADA-DB, par exemple le volume d'une couche de la superstructure de la chaussée.

## 9.2 Discussion

Les traducteurs se réalisent rapidement et le format ILI et ITF ne posent pas de problèmes graves. On constate que le transfert des données d'ACCESS à EXCEL peut être réalisé encore plus rapidement si l'on ne passe pas par le fichier ITF. Les mots clés qu'on ajoute dans le fichier ITF sont les mots qu'on retranche pour arriver au fichier txt final. Par contre, le fichier ILI peut être utile même à ce stade là, parce qu'il décrit d'une manière rigoureuse la structure des données. Les discussions avec des professionnels de la gestion des routes aboutissent à utiliser ce schéma conceptuel pour décrire un fichier txt qu'on peut insérer directement dans EXCEL. Dans EXCEL, les attributs ne sont pas vraiment gérés comme dans une base de données. L'appartenance d'une valeur à un attribut d'une table se fait visuellement. La ligne détermine à quel tuple cette valeur appartient et la colonne détermine à quel attribut elle appartient. Au niveau de la structure des données, on gagne relativement peu comparé avec un fichier txt. Dans le fichier txt, l'appartenance d'une valeur à un attribut est assuré par l'ordre dans la ligne. En introduisant des colonnes, cette relation est plus facilement mise en évidence.

La programmation a surtout montré qu'une lecture unique des fichiers n'était jamais suffisante. En plus, l'arrangement des données peut être amélioré dans le fichier ILI, parce que ce sont les dernières lignes dans le fichier ILI qui définissent le format (par exemple le séparateur des valeurs des attributs).

La programmation du traducteur ITF2XLS a soulevé la question de savoir si l'on veut faire quelques tests d'intégrité au niveau du traducteur ou si l'on veut les faire tous dans un autre processus. Il est clair que tous les tests ne sont pas faisables, puisque le nombre de contraintes formalisées dans le schéma ILI est assez restreint. Pour l'utilisateur il est préférable de savoir que tous les tests sont faits après un seul proces-

sus. La probabilité de faire des erreurs est beaucoup plus grande si on répartit les contrôles dans deux processus. En plus, que faire si l'on a transféré des données avec un autre format et qu'il y a quelques tests qui sont réalisés par le traducteur? Par contre, le programme faisant ces contrôles devrait accéder une fois de plus au schéma ILI.

Pour les traductions, nous n'avons pas pris les données des catalogues de texte, parce que le transfert serait très lourd. Mais la description de toutes les tables représentatives des catalogues de texte contenu dans l'annexe 2 en INTERLIS-DDL, permet de transférer les textes composés selon les règles d'INTERLIS.

### **9.3 Tests supplémentaires**

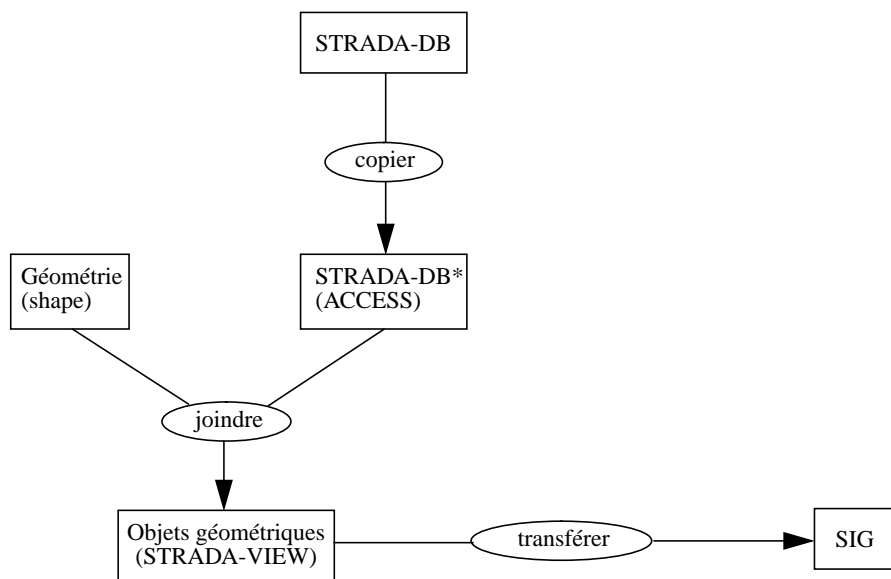
Il aurait été intéressant de transférer la géométrie des axes avec les données STRADA-DB. Dès que les STRADA-DB incluent les éléments géométriques, on les insère dans le même fichier d'entrée txt. Mais pour le moment, une jointure de ces deux types d'éléments semblerait assez artificielle. En plus, le transfert des éléments géométriques n'a de sens que si le système cible est un système d'information géographique. Un jeu de coordonnées dans une base de données n'est pas très informatif. Les axes peuvent être représentés par des polygones selon la définition d'INTERLIS. La définition des lignes prévue par STRADA-DB selon laquelle on définit une droite par le point de départ, l'azimut et la distance comporte quelques inconvénients importants. La continuité d'un segment d'axe n'est pas garantie. Comment peut-on savoir si le point final d'une droite correspond au point de départ de la droite suivante? L'avantage est de pouvoir représenter tous les segments des axes dans la table *Axe* sans créer une table supplémentaire *Segments des axes*. Mais le risque de commettre une faute topologique nous semble trop grand. Une autre possibilité est de définir la première droite avec le point de départ, l'azimut et la distance et les droites suivantes uniquement avec l'azimut et la distance. Dès qu'il y a une discontinuité, on remet le point de départ. L'inconvénient principal est d'avoir plusieurs définitions pour une droite. En évaluant toutes les possibilités, la définition d'INTERLIS est la meilleure.

Il est clair que la représentation des routes comme lignes dans un système d'information est relativement maigre. Il est nécessaire d'introduire des largeurs des routes. Bien qu'on ait des largeurs dans le catalogues de données de STRADA-DB, appelé profil géométrique, il est préférable de décaler la géométrie de l'axes des deux côtés d'une constante. Cette constante dépend uniquement de l'importance des routes (Autoroutes, routes cantonales). Il est important de considérer la géométrie des axes plutôt comme une représentation graphique. Elle ne coïncide qu'approximativement avec la réalité. Seules les données pures de STRADA-DB sont à utiliser pour se repérer dans le monde réel. Il est donc beaucoup plus facile d'avoir une largeur fixes pour la

représentation graphique et les informations ayant une référence curviligne doivent s'adapter à la représentation tel que les proportions correspondent plus ou moins à la réalité.

Le transfert des concepts de représentation, comme l'axe tendu, et ses paramètres ne font pas partie du schéma ILI, parce que cette est une méthode de représentation de données et non pas des données nouvelles. Par contre, on pourrait transférer la sémiologie graphique utilisée pour la représentation graphique dans une couche pour avoir un certain standard dans la visualisation.

On peut imaginer aussi le scénario suivant: Au lieu de transférer des données de STRADA-DB vers un autre système d'information, on peut aussi transférer des données à partir de STRADA-VIEW vers un autre système. L'avantage de ce scénario est d'avoir la géométrie en forme de SHAPE d'ARCVIEW et les informations alphanumériques qui sont liées aux entités spatiales. Par contre, on n'a pas d'entités alphanumériques et il n'est pas possible de transférer l'historique des entités, puisque STRADA-VIEW ne gère pas l'historique. En plus, on devrait régulièrement mettre à jour l'extrait des entités spatiales auquel STRADA-VIEW se réfère. Dans ce scénario, on n'aurait pas besoin de toucher STRADA-DB pour chaque transfert.



---

FIGURE 13.

Le scénario de jointure des éléments géométriques avec les objets et le transfert à partir de STRADA-VIEW.



## 10.0 EVOLUTION DANS LE DOMAINE DES BASES DE DONNÉES

Bases de données orientées  
objets

Une alternative aux bases de données relationnelles sont les bases de données dites orientées objets. On a constaté que le modèle relationnel avait quelques faiblesses. Il est trop simple pour représenter les objets dans le monde réel. En plus, le modèle relationnel est incompatible avec les langages de programmation. C'est la raison pour laquelle on a inventé les bases de données orientées objet. Ce type de base de données permet de représenter des structures plus complexes, les données et les traitements ne sont plus séparés. En plus, tous les objets possèdent des identifiants uniques. La clé composite des modèles relationnelles pose un problème si l'on veut mettre à jour cette clé. Par contre, dans les modèles orientés objets, l'identifiant reste stable pendant toute la vie de l'objet. Les objets sont liés par des liens de composition ('un objet est composé de tels objets') ou des liens de généralisation / spécialisation ('un objet est un sous-groupe d'un autre objet'). Dans ces derniers liens, des méthodes peuvent être héritées. Une méthode valable pour une superentité est transmise aux sous-entités.

### EXEMPLE

```

CLASS Etudiant
TYPE TUPLE (num: INT,
            nom: STRING,
            prénoms: SET STRING,
            adresse: TUPLE (nr: INT,
                           rue: STRING,
                           ville: STRING,
                           pays: STRING)
            cours suivis: SET TUPLE (nom-cours: STRING,
                                     NOTE: REAL))

```

On s'aperçoit que la démarche décrite dans le chapitre 'Test de INTERLIS' pour mettre en évidence les dépendances ressemble fortement à la démarche pour l'élaboration du schéma conceptuel orienté objet, mais cette fois au niveau de la base de données et non pas au niveau du transfert. Comme la base de données de STRADA-DB est faite sur ORACLE, typiquement relationnelle, la méthode orientée objets n'est pas à reprendre comme telle, mais d'étendre la base de données avec des améliorations proposées par des méthodes orientées objets. STRADA-DB et INTERLIS, par exemple, ont repris la notion de objet identifier (OID).

Avec INTERLIS-DDL, la structure d'une base de données orientée objets est difficilement représentable, parce que INTERLIS-DDL ne possède pas le concept de super-entité/ sous-entités. En plus, le mécanisme d'INTERLIS ne permet pas de transférer des méthodes.

La plupart des bases de données actuelles sont relationnelles à cause de la plus grande simplicité.

#### Open GIS

Au niveau de la standardisation du transfert des données spatiales, une autre approche a été choisie par un consortium appelé Open GIS Consortium, qui a été créé en 1994. Ce consortium ne se contentait pas de standardiser des données, comme les formats, ni de standardiser le formalisme de description des modèles conceptuels, comme le CEN veut le faire, mais de trouver un standard au niveau informatique appelé Open Geodata Interoperability Specification (OGIS). OGIS a le but de fournir le cadre pour les concepteurs des logiciels qui permet à leurs utilisateurs d'accéder et de traiter des données géographiques provenant des différentes sources à l'aide d'une interface générique qui respecte les principes de la technologie d'information ouverte. Par une technologie ouverte on comprend que cette interface doit aussi être accessible par des utilisateurs des systèmes d'information non géographiques. Pourquoi faut-il avoir un standard au niveau informatique? Admettons un monde hétérogène de systèmes d'information géographiques, le transfert de données spatiales est garanti par des traducteurs de données. Une perte d'information est inévitable, puisque les logiciels de système d'information gère les informations géographiques et non géographiques d'une autre manière. OGIS essaie de résoudre ce problème d'interopérabilité en définissant un standard pour les logiciels.

Pour citer un exemple, OGIS définit la méthode de traitement de la géométrie:

1. par un jeu de points ayant des coordonnées d'un même système de référence.
2. par une collection de géométries décrit dans un même système de référence. Par géométrie, on entend soit un point, une courbe, une surface ou un solide.
3. par un algorithme qui construit avec le point et les géométries une entité spatiale.
4. par un système de référence spatial et temporel qui donne à l'entité une représentation dans le monde réel.

Pour les administrations des routes, qui utilisent STRADA-DB et qui envisagent de transformer cette base de données en un système d'information géographique, les décisions prises par le Open GIS Consortium peuvent être très importantes, surtout pour la description des éléments spatiaux. Si par exemple OGIS décide qu'une surface ne peut pas contenir plus que cinq îlots, il ne faut pas créer un concept qui considère une commune comme une surface et tous les parcelles comme des îlots. Ceci parce que les développeurs, membre du OGC, ne supporteraient pas ce concept.

## 11.0 SYNTHÈSE

Rentabilité des transferts basés sur les schémas

Le choix final d'un mécanisme d'échange dépend fortement de la rentabilité. A quel niveau est-ce que les transferts basés sur des schémas sont rentables?

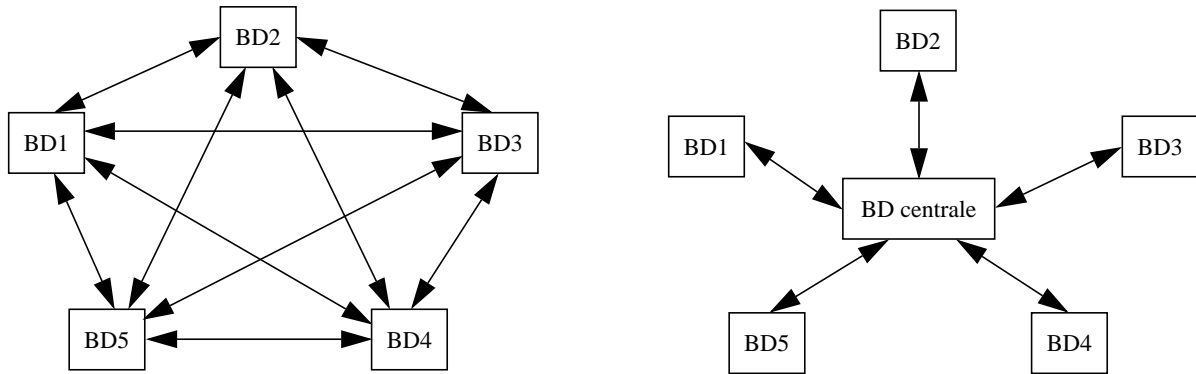


FIGURE 14.

### Différence entre système fédéré et système centralisé

Prenons le cas des formats propriétaires et le scénario décrit à la figure 14 à gauche. Admettons que pour un transfert quelconque, on utilise le format engendré par la base de données de numéro inférieur, par exemple pour un transfert des données entre la BD2 et la BD4, on utilise le format de la BD2. Si l'on admet que toutes les bases de données communiquent entre elles uniformément, on a besoin de:

$$\sum_{k=1}^n 2 \times (k-1)$$

traducteurs. Si l'on utilise un échange basé sur des schémas, on a besoin de deux traducteurs par base de données, un traducteur pour la création du format intermédiaire et un traducteur pour la lecture de ce format. On a besoin de  $2 \times n$  traducteurs. Pour que l'utilisation des mécanismes comme INTERLIS soit rentable, il faut avoir au moins quatre bases de données différentes (différentes au niveau informatique et pas au niveau concept).

Si l'on compare cette configuration avec un système centralisé dans laquelle tous les transferts se déroulent via la base de données centrale, l'utilisation n'est pas rentable. Pour le transfert propre au système central, les  $n$  systèmes périphériques ont besoin de deux traducteurs, pour la lecture et pour la création du format. Le système central n'a pas besoin d'un traducteur, puisqu'il s'agit de son propre format. En résumé, on doit écrire  $2n$  traducteurs. Si l'on utilise le mécanisme d'INTERLIS, on doit créer deux traducteur de plus ( $2n+2$ ), parce cette

fois le système central doit écrire les données dans le format ITF et de lire des données de ce format. INTERLIS nécessiterait plus de traducteurs que les transferts avec des formats propres.

Le système de l'administration des routes est d'une part un système fédéré parce que toutes les administrations cantonales sont souveraines dans la conception de leurs bases de données. A première vue, l'administration peut être un système centralisé, puisque toutes les administrations cantonales sont censées d'utiliser STRADA-DB. Mais ce point de vue est faux, parce que les administrations cantonales ont la liberté de coupler les données de STRADA-DB avec n'importe quel système d'information. Vu la grande diversité de système d'information des routes, par exemple STRADA-Arc, STRADA-Adalin, etc., l'utilisation du mécanisme INTERLIS est utile.

Les transferts basés sur les formats

L'évaluation des formats standards de description des routes, comme GDF et RADEF, a démontré clairement leurs limites. D'une part, ces formats ont été créés pour satisfaire le besoin en données routières dans un domaine d'activité particulier, comme GDF pour la navigation automobile, et d'autre part ces formats sont le fruit de discussions entre divers partenaires n'ayant pas forcément la même opinion. RADEF représente relativement peu de données, parce que les administrations nationales n'étaient pas prêtes à standardiser plus. Le schéma RADEF pourrait être bien représenté par le schéma conceptuel de INTERLIS au lieu d'avoir une liste des entités et des attributs définis dans le dictionnaire de RADEF. Par contre, GDF représente un modèle de métadonnées complet et rigoureux, qui a révélé un défaut de INTERLIS-DDL. Dans le schéma ILI, il n'est pas possible de représenter un modèle de métadonnées globales pour toutes les couches d'informations.

Schéma conceptuel

INTERLIS fournit un formalisme (INTERLIS-DDL) qui permet de décrire un modèle conceptuel de données d'une manière relativement simple, mais suffisamment précis. Ce schéma peut être élaboré aussi pour STRADA-DB. Il s'agit donc aux partenaires de STRADA-DB de s'entendre sur un schéma, puisqu'il est possible de créer plusieurs schémas pour représenter un seul modèle conceptuel. L'introduction d'un schéma unique dans la communauté d'information exige de prendre des décisions et de fixer des priorités. La communauté doit s'accorder sur une identification unique, sur un transfert des catalogues de textes, sur le transfert de l'historique des entités et sur le transfert du système de repérage. Le travail a mis en évidence toutes les possibilités de décrire les concepts de STRADA-DB en INTERLIS. Le choix final de description dépend d'une pondération des critères et sur cette pondération un consensus est absolument nécessaire à l'intérieur de cette communauté. En plus, les éléments de base pour la description de la géométrie des axes doivent être clairement définis, même avant l'introduction de la géométrie dans STRADA-DB. Le transfert des données géométriques et des données alphanumériques peut être réalisé avec INTERLIS bien que la source de ces deux types d'informations soit différente.

Le schéma en INTERLIS-DDL (fichier ILI) peut être modifié sans que les traducteurs changent. Mais le but est de créer un schéma statique, qui satisfait la plupart des partenaires et intervenants de STRADA-DB. Ce schéma statique doit décrire presque complètement le modèle conceptuel de STRADA-DB parce que STRADA-DB représente le noyau commun de toutes les bases de données routières des administrations.

### Format de INTERLIS

Contrairement à l'utilité du schéma ILI, l'évaluation du mécanisme de INTERLIS a démontré quelques faiblesses du format ITF engendré par le schéma, comme l'arrangement des données dans le fichier ITF. Ces inconvénients n'empêchent pas le transfert, mais diminuent la performance du transfert. Ils allongent le temps de transfert et compliquent l'écriture des traducteurs. Les défauts du format ont été démontrés surtout par le test, qui a consisté à transférer un jeu de données de STRADA-DB au tableur EXCEL. Le test a démontré aussi que la traduction des données en format ITF était une étape inutile, parce que les mots clés de ITF, que nous avons ajoutés, ont été effacés pour l'import des données dans EXCEL. On pourrait imaginer le scénario où le fichier ILI décrit directement un fichier texte. Pour le transfert de données dans un outil informatique, qui n'est pas de type base de données, cette démarche est tout à fait satisfaisante. Mais si le système cible est une base de données, l'utilisation du format ITF est essentielle pour ne pas perdre l'indépendance aux outils informatiques.

A notre avis, le fichier ILI ne doit pas contenir des méthodes, parce qu'une intégration des méthodes compliquerait le schéma conceptuel. Ce schéma doit conserver son but initial: la description de la structure de données. Mais on peut imaginer de transférer non seulement le fichier ILI, mais aussi un fichier qui contient les méthodes, un fichier 'MET'. On pourrait ajouter des contraintes formalisées dans le schéma ILI et la traduction des contrôles dans un langage de programmation se décrirait dans le fichier MET. Il est clair qu'au niveau de la communication, on devrait se contenter d'utiliser un seul langage de programmation. Dans le fichier ILI, on devrait mentionner dans la section des métadonnées l'identifiant et la version du fichier MET à laquelle toutes les formulations des contraintes se réfèrent.

### Transfert entre systèmes d'information géographiques

INTERLIS a été fait surtout pour le transfert de données entre différents systèmes d'information géographiques. L'apparition de traducteurs puissants qui maîtrisent les traductions entre les logiciels de système d'information géographiques les plus répandus, a remis en question l'utilité d'INTERLIS. FME, un de ces traducteurs, passe aussi par un format intermédiaire (SAIF). Il traduit toutes les données d'abord du système source en format SAIF et à partir du format SAIF il crée le format utilisé par le système cible. FME agit comme une boîte noire, puisque l'utilisateur peut préciser le système source, le système cible et le fichier qui contient les données et la traduction se fait directement sans que l'utilisateur puisse savoir ce qui se passe à l'intérieur. Le fichier de départ doit contenir des données sans défauts de structure. La traduction avec FME ne peut pas garantir cette cohérence. Au niveau économique,

le traducteur FME coûte Fr. 1500.-. Si l'on veut avoir le FME avec le traducteur INTERLIS inclu, cela coûte environ Fr. 4500.-. La seule profession qui doit utiliser INTERLIS est celle des géomètres, parce que la forme des données de la mensuration officiel (INTERLIS) est déclarée dans une ordonnance et cette loi a un caractère normatif.

Par contre, la version 2 de INTERLIS offrirait la possibilité de ne transmettre que les modifications par rapport à un état d'une base de données. Ce point n'est pas couvert par les traducteurs, comme FME.

INTERLIS peut être très bien utilisé si les données proviennent de différentes sources. On a plusieurs fichiers d'entrée et on crée un seul fichier ITF. Dès que STRADA-DB est intégré dans un système d'information géographique, l'utilité de INTERLIS est à réévaluer. Il est aussi très important de ne pas négliger les décisions prises par le consortium OGIS, parce que tous les essais d'introduire un standard dans une communauté d'information ne valent rien, s'ils ne correspondent pas à la méthode de gestion des données spatiales des logiciels des systèmes d'information.

## 12.0 CONCLUSION

---

L'objectif de ce travail était de montrer l'adéquation d'INTERLIS pour l'échange de données pour la gestion de l'entretien routier. STRADA-DB est l'outil principal pour la gestion de données routières de la Confédération et des cantons. A partir de STRADA-DB le besoin en échange de données a été exprimé. Ensuite, une étude sur les divers mécanismes d'échange a été réalisée.

Nous avons pu montrer que le mécanisme d'INTERLIS peut très bien être utilisé pour la création d'une interface standardisée pour la gestion de l'entretien routier. Avec le langage INTERLIS-DDL, à priori toutes les entités et tous leurs attributs de STRADA-DB peuvent être représentés dans le schéma conceptuel. Par conséquent, le transfert des occurrences qui correspondent à la description dans le schéma conceptuel peut être réalisé. Pour les transferts basés sur des formats, comme RADEF et GDF, il n'est pas possible de transférer toutes les données de STRADA-DB, parce que ces formats ont été créés pour des objectifs différents. Ces mécanismes sont peu flexibles et très statiques. INTERLIS, par contre, ne définit pas le modèle conceptuel d'échange, mais il définit le mécanisme d'échange et le formalisme à utiliser pour créer le schéma conceptuel.

Il existe des traducteurs sur le marché qui permettent de transférer des données d'un logiciel de système d'information géographique à un autre. Ces outils sont très puissants, puisqu'ils peuvent faire de traductions à partir de nombreux logiciels de systèmes d'information. Par contre, il n'existe que peu de traducteurs suffisamment puissants qui utilisent le mécanisme d'échange d'INTERLIS. L'écriture de traducteurs capables de dériver le format à partir de n'importe quel schéma conceptuel INTERLIS est difficile parce que les utilisateurs ont beaucoup de liberté pour s'exprimer dans le schéma (à l'extrême les contraintes d'intégrité sont décrites en langage courant).

Le mécanisme d'échange d'INTERLIS passe par le schéma conceptuel. Avant le transfert de données, un schéma doit être créé. Dans INTERLIS, on prend l'hypothèse que l'utilisateur qui gère le système source et l'utilisateur du système cible sont des personnes différentes. Avec le transfert du schéma la communication nécessaire de la structure des données est garantie. Par contre, si un utilisateur change uniquement le logiciel de système d'information géographique, il va intégrer toutes les données dans un système d'information vierge. Il n'a pas besoin de schéma parce qu'il connaît le modèle conceptuel de la base de donnée source. Le gestionnaire utiliserait un mécanisme qui traduit directement les données pour éviter d'écrire un schéma uniquement pour ce transfert. Par contre, le schéma conceptuel reste un moyen de décrire la structure de données une fois pour toutes.

Les défauts dans le format d'INTERLIS sont essentiellement dûs aux définitions des attributs géométriques du schéma conceptuel. Les lots, qui exigent de préciser les éléments géométriques après la table principale dans le format (contrairement aux surfaces et aux polygones), compliquent l'écriture des traducteurs. Pour STRADA-DB, il n'est pas nécessaire d'utiliser des lots, parce qu'elle n'exige pas d'absence de recouvrement pour les éléments surfaciques.

Pour la description des données de STRADA-DB en INTERLIS-DDL, nous proposons de créer trois couches (TOPIC). La première couche est la couche des métadonnées, qui doit contenir tous les paramètres nécessaires pour la définition de système de projection et l'ellipsoïde. La deuxième couche contient tous les éléments de STRADA-DB, c'est-à-dire les éléments constitutifs du système de repérage de base et les réseaux d'exploitation. Si l'on voulait séparer le système de repérage de base et les réseaux d'exploitation dans deux couches différentes, on devrait répéter la géométrie des axes et la position des points de repère parce qu'INTERLIS exige que les couches soient indépendantes. La troisième couche contient tous les éléments nécessaires pour définir les catalogues de textes. Les listes de codes sont définies dans la deuxième couche comme énumération dans la section du domaine.

Nous proposons d'utiliser les polygones pour décrire la géométrie des axes, un attribut géométrique pour décrire la planimétrie et un attribut géométrique pour décrire l'altimétrie avec la distance longitudinale et l'altitude. Les distances curvilignes et les écarts des objets ponctuels, linéaires ou surfaciques sont transférés comme attributs alphanumériques. C'est au traducteur qui crée le format propre du système cible, de donner à ces éléments de repérage une signification dans l'espace. On peut faire cela parce que ces descriptions géométriques ne sont pas vraiment multivaluées. Un point est repéré par quatre attributs: l'axe, le point de repère, la distance longitudinale et l'écart. L'élément linéaire est défini par deux points et un élément surfacique est défini par quatre points (trapèze).

INTERLIS est capable de décrire la structure de STRADA-DB. Si l'on choisit d'extraire des données directement de STRADA-DB ou de transférer les données à partir de STRADA-VIEW, on doit passer par un fichier intermédiaire, soit le format ITF ou un autre format. Le schéma conceptuel est un moyen de transférer le modèle conceptuel de la base de données de 'communication' et de décrire un format. Le schéma n'est pas un outil pour la conception d'une base de données, puisqu'il existe des méthodes plus efficaces, comme le diagramme entité-association.

A la suite de ce travail, il serait particulièrement intéressant de comparer Open GIS par rapport à INTERLIS et évaluer l'importance des éléments manquant d'INTERLIS pour l'échange des données de STRADA-DB. De plus, l'intégration des éléments géométriques dans STRADA-DB et dans le transfert est à étudier. Bien que ce travail ait

---

## Conclusion

---

fourni quelques réflexions au niveau du transfert des éléments géométriques, il est nécessaire d'étudier les différents scénarios proposés. Finalement on pourra prendre les décisions nécessaires pour établir un échange standardisé dans le cadre de la gestion de l'entretien routier.

---

## Conclusion

---

## 13.0 BIBLIOGRAPHIE

---

Aebischer, M. and Sturdik, P. (1996): STRADA-DB: Konsistenzbedingungen Version 2.10.

ARIS AG (1995): STRADA-V0, Technischer Bericht.

Beilage 5.: EXPRESS Sprachelemente, Uebersicht une Vergleich mit INTERLIS.

Brochure RADEF: ... a system to provide access to European highway information.

Buehler Kurt and McKee Lance (1996): The Open GIS Guide, Introduction to Interoperable Geoprocessing, [<http://opengis.org/techno/specs.htm>].

CEN TC 287 (1996): Geographic Information - Data Description - Spatial Schema, prénorme européenne, final draft, prENV12160.

CEN TC 287 (1996): Geographic Information - Reference Model, prénorme européenne, final draft, prENV12009.

CEN TC 287 (1996): Geographic Information - Data Description - Transfer, norme européenne, draft, prEN287-010.

CEN TC 287 (1997): Geographic Data Files, Version 3.0, [<http://www.intergraph.com/ehq/gdf/gdfdoc/GDFDOC.stm>].

Claussen H. and Histenmann V. (1995): GDF works! The Validation of the Digital Map Standard, [<http://www.intergraph.com/ehq/gdf/gdfpub/GDFPUB.stm>].

Claussen, H. and Lichtner, W. (1989): GDF, a proposed standard for digital road maps to be used in car navigation systems, [<http://www.intergraph.com/ehq/gdf/gdfpub/GDFPUB.stm>].

Département fédéral de la justice et police (1994): Ordonnance technique sur la mensuration officiel (OTEMO).

Département fédéral des transports et de l'énergie, Office fédérale des routes (1994): STRADA-DB: Banque de données routières pour le SGE, Guide d'introduction et d'exploitation.

Direction du projet REMO (1991): INTERLIS mécanisme d'échange de données pour systèmes d'informations de territoire.

Direction fédéral de la mensuration cadastrale, Direction du projet REMO (1992): 'Commentaires sur la description du catalogue des données de la mensuration officielle en INTERLIS'.

Eidgenössisches Verkehrs- und Energiedepartement, Bundesamt für Strassen (1994): STRADA-DB: Strassendatenbank für das MGE, Leitfaden für die Einführung und den Betrieb.

Germann, M.; Dorfschmid, J. (1997): Inkrementelle Nachlieferung mit INTERLIS

Gnägi, H.R. (1995): Datenaustausch zwischen GIS , In: Vermessung, Photogrammetrie, Kulturtechnik 4/95, p. 183-187.

Gnägi, H.R. : INTERLIS DDL, The Swiss Conceptual Schema Language.

Gnägi, H.R. : Requirements for Conceptual Schema Language.

Gnägi, H.R. und Keller, S. (1997): 'Glossar für INTERLIS und AVS', Eidgenössische Vermessungsdirektion, Bern.

Gnägi, H.R., Golay, F., Keller, S., Sievers, B., Wicki, F. (1997): 'INTERLIS - eine Standortbestimmung', In: Vermessung, Photogrammetrie, Kulturtechnik 7/97, p. 472-481.

Heres, Luc (1991): Standards for drive information systems, [<http://www.intergraph.com/ehq/gdf/gdfpub/GDFPUB.stm>]

Heres, Luc and Wood, Tim (1992): GDF, a lingua franca for geographic information, [<http://www.intergraph.com/ehq/gdf/gdfpub/GDF-PUB.stm>].

ISO TC 211 WG1-WI3 (1996): Conceptuel Schema Language.

ISO TC 211, Berre, A.J. (1995): Workshop on Conceptual Modelling Language, SINTEF.

ISO TC 211: Conceptual Schema Language Evaluation, INTERLIS Candidature, Presentation and Overview of Submitted Documents.

Jufer-Meier, R., Etienne, M., Morgenthaler, M. (1996): STRADA-DB, STRADA-DB Logical Data Model Version 2.10.

Keller, S. (1997): A representation model and a mapping language for INTERLIS.

Keller, Stefan: INTERLIS und DXF im Vergleich: Geo-Datenaustausch im Strukturwandel, [<http://www.geo.unizh.ch/~keller/ILI-VSE.html>].

Modèle conceptuel de la mensuration officielle

Projektleitung RAV (1991): INTERLIS ein Daten-Austausch-mechanismus für Land-Informationen-Systeme.

Schweizer, Michael (1997): STRADA-INTERFACE, Konzept, Import, Export, Dokumentation, Datenschema, Entwurf.

Soor, Tal (1997): Project RADEF: Data Model and Data Dictionary, Release 03.

Spaccapietra, Stefano (1997): SGBD orienté objet, En: Base de données Volume 2.

Thomas, Jonathan (1997): Project RADEF: Data Dictionary and Tool Design.

Vereinigung Schweizerischer Strassenleute VSS (1993) Norme SN 640940, Catalogue des données routières.

Vereinigung Schweizerischer Strassenleute VSS (1993), Norme SN 640941, Catalogue des données routières, repérage dans l'espace.