



Retours sur l'utilisation du pack de 40 heures

Département des Côtes d'Armor

Application Geotrek-Admin

Octobre 2021



Paris

Nantes

Toulouse

Bruxelles

Makina Corpus

Siège social
11 rue du Marchix
44000 Nantes

Établissement de Toulouse
52 Rue Jacques Babinet
31100 Toulouse



+33 (0)9 70 33 21 50



Table des matières

1	Contexte.....	3
2	Vocabulaire.....	3
3	Fonctionnement en place historiquement dans Geotrek-Admin	3
3.1	Topologie et pathAggregation.....	3
3.2	Calcul d'ordre.....	4
4	Énoncé du problème du Département.....	4
5	1ère analyse et tentative de solution : ajouter des points de passage automatique lors de l'édition d'une topologie.....	4
6	Explications du problème de l'ordre des pathAggregations.....	6
6.1	D'un point de vue affichage et fonctionnel.....	6
6.2	Focus sur les triggers SQL.....	7
7	Recherche de solutions.....	11
7.1	Avec les triggers existants.....	11
7.2	Ajouter un nouveau trigger.....	13
7.3	Remplacer les triggers SQL par du code Python.....	13
7.4	Conclusion autour des triggers SQL.....	13
8	Conclusion des développements et des tests.....	14
9	Suite de l'étude.....	14

1 Contexte

Le Département des Côtes d'Armor a relevé une anomalie concernant des modifications non souhaitées d'itinéraires (<https://github.com/GeotrekCE/Geotrek-admin/issues/2515>).

Dans le cadre de la consultation et de la commande « 2020-DSI7-MB » portant sur « des prestations d'expertise de paramétrage et développement d'évolutions sur Geotrek-Admin », le Département a fait l'acquisition d'un pack de 40 heures (soit 5 jours) auprès de Makina Corpus.

Le présent document explique comment a été utilisé ce premier pack d'heures dans l'investigation et la compréhension du problème soulevé.

2 Vocabulaire

- **PathAggregations** : partie d'une topologie, qui permet de rattacher, par rapport à un tronçon, l'emplacement d'un objet.
- **Topologie** : concerne tous les objets* s'appuyant sur la segmentation dynamique pour se rattacher automatiquement à un tronçon.

Par exemple, les contenus touristiques ne sont pas des topologies car ils sont rattachés via une zone tampon (buffer).

*POIs, Itinéraires, Services, Sentiers, Infrastructures, Signalétiques, Statuts Foncier, Statuts physique, Tronçon de compétence, Tronçon de gestion de travaux, Tronçon de gestion signalétique.

3 Fonctionnement en place historiquement dans Geotrek-Admin

Une topologie est stockée sous la forme d'une liste d'objets, nommés pathAggregations dans Geotrek-Admin, qui correspond à une séquence ordonnée de morceaux de tronçons.

3.1 Topologie et pathAggregation

Un tronçon A a une topologie, qui est elle-même composée d'un pathAggregation.

Lors de la découpe du tronçon A, on se retrouve avec deux tronçons :

- le tronçon A qui a été découpé, nommé désormais A' ;
- un nouveau tronçon, nommé tronçon B, qui est la seconde partie du tronçon A.

On obtient alors 2 tronçons A' et B.

La topologie qui était sur le tronçon A est alors incomplète.

Le pathAggregation initial ne couvre alors que le tronçon A', mais il manque le second pathAggregation qui couvre le tronçon B et qui va devoir être créé.

Geotrek-Admin recalcule alors automatiquement le pathAggregation du tronçon B.

Lors de ce recalcul du pathAggregation, Geotrek-Admin recalcule le point d'arrivée sur le tronçon B à partir de l'ancien point d'arrivée du tronçon A.

Un pathAggregation est alors recréé sur le tronçon B.

3.2 Calcul d'ordre

Lors de la création d'une topologie (exemple d'un itinéraire), Geotrek-Admin sauvegarde l'ordre de passage par différents pathAggregations.

En utilisant, cet ordre de passage, Geotrek-Admin est capable de reconstituer la géométrie d'un itinéraire, ce qui permet de l'afficher.

Or la création d'un nouveau pathAggregation invalide l'ordre de passage précédemment enregistré.

Il est important de préciser qu'un point de passage est stocké en tant que pathAggregation.

4 Énoncé du problème du Département

Le Département sollicite Makina Corpus pour le problème suivant : « quand on insère un tronçon, Geotrek-Admin recalcule le chemin le plus court sur un itinéraire ».

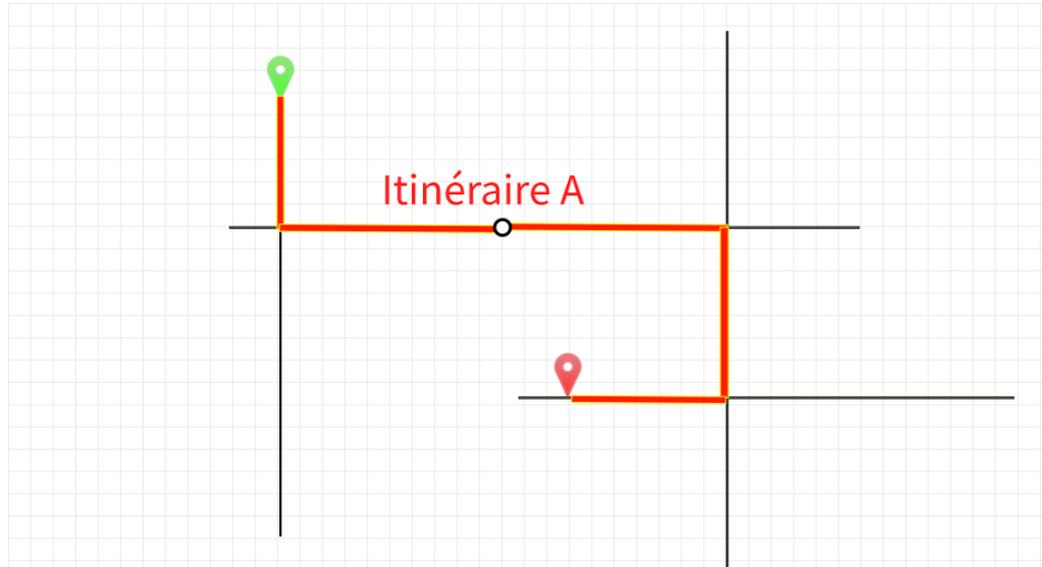
Le Département met alors en doute la qualité de la donnée.

5 1ère analyse et tentative de solution : ajouter des points de passage automatique lors de l'édition d'une topologie

Lors de notre première compréhension du problème et pour apporter une solution, nous avons cherché à ajouter un moyen de conserver l'ordre de passage lors de la sauvegarde des topologies, en ajoutant des points de passage lorsqu'il y avait une différence avec la géométrie d'origine.

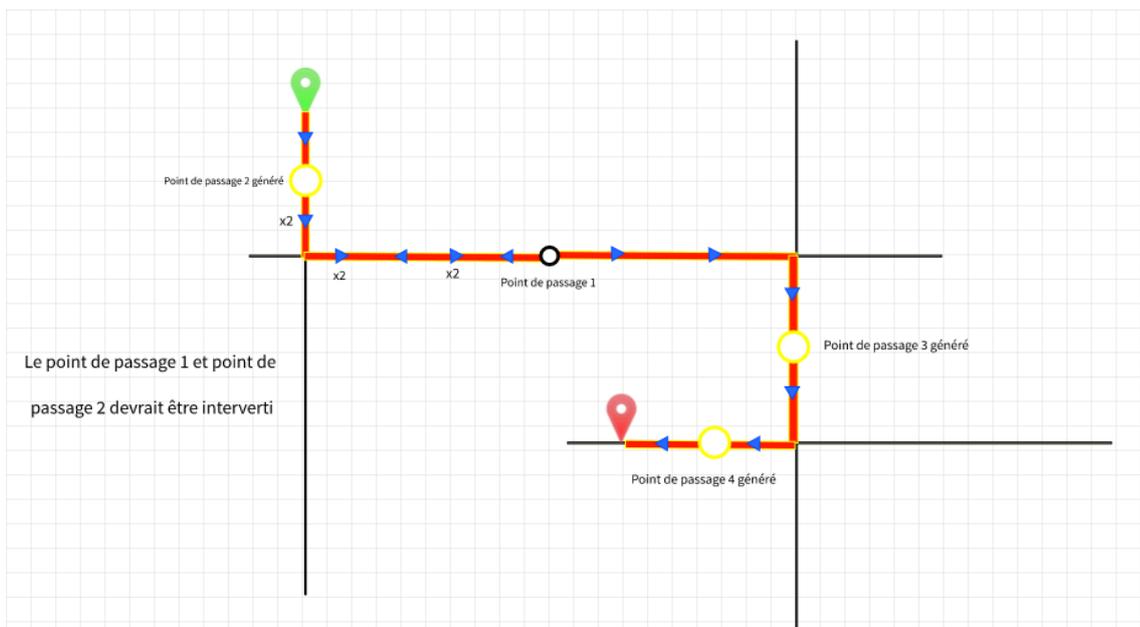
(Temps passé = 2 Journées)

Exemple de mise en place de points de passage, qui devait permettre une conservation de la géométrie :



Cette première tentative d'implémentation de solution a révélé l'existence de problèmes d'ordre de passage : sur certains itinéraires, après l'ajout des points de passage, on constate que la topologie passe plusieurs fois par le même endroit.

Les points de passages sont calculés en fonction de l'ordre du pathAggregation.



Après la première tentative d'implémentation d'un correctif, nous avons observé que l'ajout de point de passage engendrait alors un autre problème.

Le problème est alors plus profond et plus complexe à analyser et à résoudre : il s'agit finalement de l'ordre stocké dans les pathAggregations et de leur gestion et non pas que des points de passage et du recalcul de l'itinéraire.

6 Explications du problème de l'ordre des pathAggregations

6.1 D'un point de vue affichage et fonctionnel

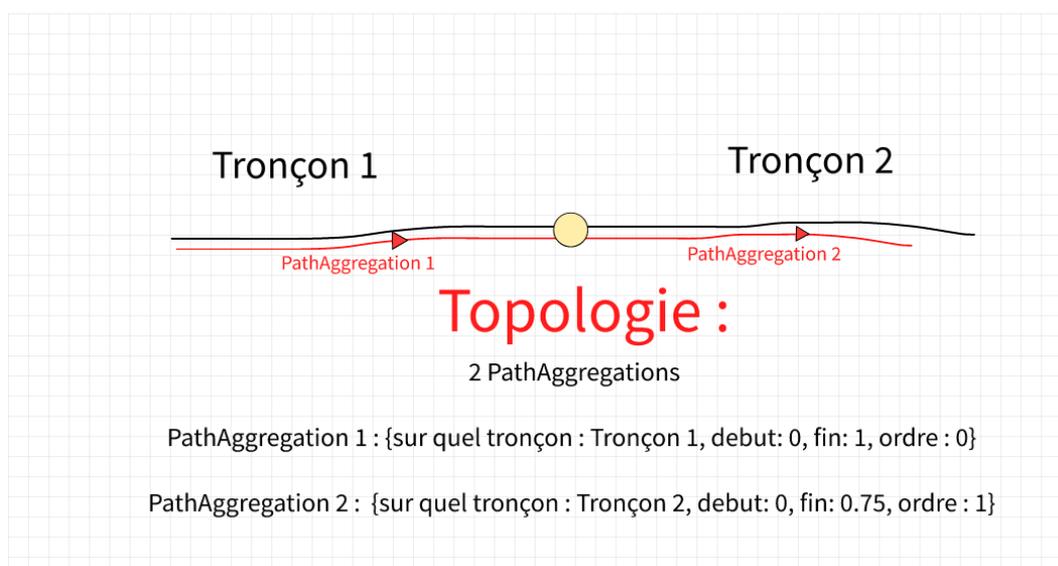
Le Département a tracé des itinéraires, basés sur un réseau de tronçons, en mettant différents points de passage à divers endroits.

Les itinéraires ont été sauvegardés et sont alors « propres ».

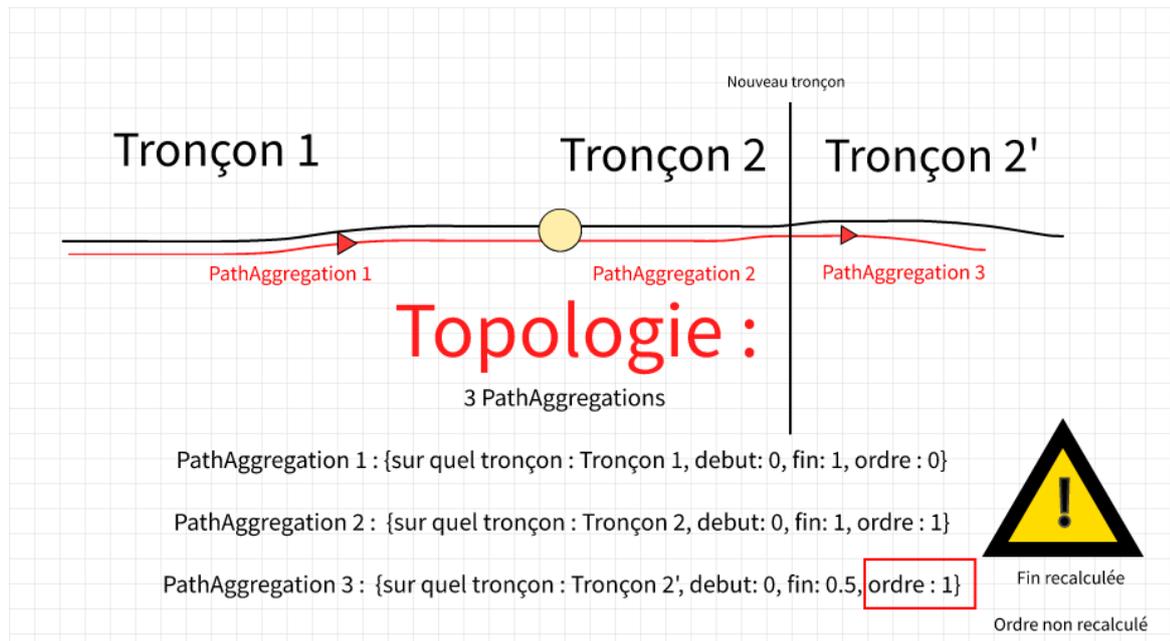
Plus tard, le Département a créé de nouveaux tronçons, qui sont alors venus recalculer les pathAggregations en place et en recréer de nouveaux dans le cas décrit ci-dessus.

Lors de la création d'un nouveau tronçon, l'application recrée automatiquement un nouveau pathAggregation. L'ordre des points de passage du nouveau tronçon n'est alors plus conservé.

Tronçons d'origine et ordre de passage



Tronçons découpés et et ordres de passage recalculés



Lors de la re-création de la topologie à partir des nouveaux pathAggregations et comme certains ordres sont dupliques, Geotrek-Admin va aléatoirement choisir un ordre plus qu'un autre, ce qui va alors soit conserver la topologie dans sa bonne forme, soit totalement la corrompre.

Lors de la corruption, la géométrie n'est alors plus une ligne mais devient une polyligne.

Cet aspect aléatoire a alors rendu plus complexe son analyse.

6.2 Focus sur les triggers SQL

La gestion des PathAggregations est faite de plusieurs triggers SQL (code déclenché automatiquement dans la base de données lors de la modification des géométries).

L'étude des triggers SQL concernés a permis de repérer un premier problème dans un premier trigger SQL, nommé *post_50_paths_split*, chargé de la mise à jour des pathAggregations après la découpe des tronçons.

Extrait du code du traitement d'un nouveau pathAggregation lors de la découpe d'un tronçon

```
-- Copy topologies overlapping start/end
INSERT INTO core_pathaggregation (path_id, topo_object_id, start_position, end_position,
"order")
  SELECT
    tid_clone,
    et.topo_object_id,
    CASE WHEN start_position <= end_position THEN
      (greatest(a, start_position) - a) / (b - a)
    ELSE
      (least(b, start_position) - a) / (b - a)
    END,
    CASE WHEN start_position <= end_position THEN
      (least(b, end_position) - a) / (b - a)
    ELSE
      (greatest(a, end_position) - a) / (b - a)
    END,
    et."order"
  FROM core_pathaggregation et,
       core_topology e
  WHERE et.topo_object_id = e.id
        AND et.path_id = path.id
        AND ((least(start_position, end_position) < b AND greatest(start_position,
end_position) > a) OR -- Overlapping
            (start_position = end_position AND start_position = a AND "offset" = 0)); -- Point
```

Ce code montre que l'ordre du pathAggregation d'une topologie est conservé lorsqu'on coupe un tronçon en deux :

PathAggregation : [Tronçon | début (entre 0 et 1) | fin (entre 0 et 1) | ordre]

Légende des 2 figures ci-dessous :

Topologie 1

Topologie 2

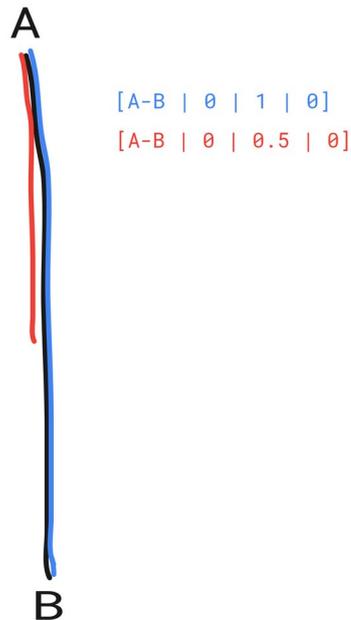


Figure 1: Tronçon initial sur lequel passe 2 topologies

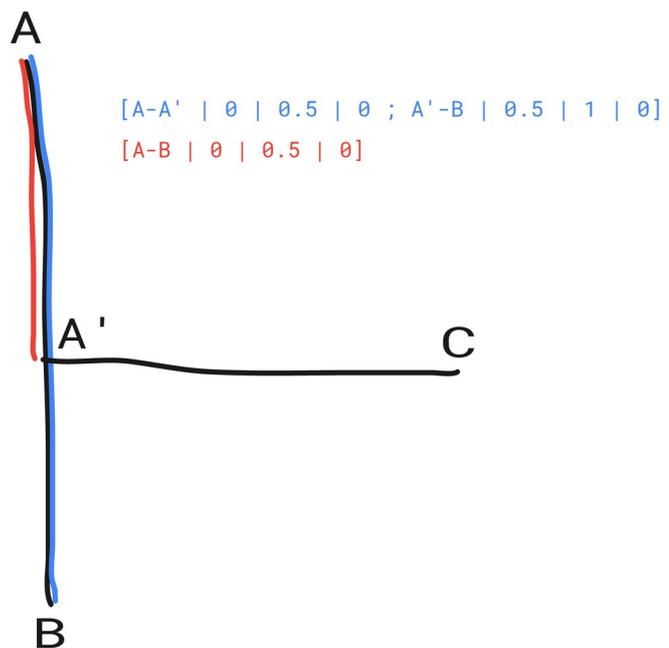


Figure 2: Ordre des pathAggregations dupliqué suite à la découpe d'un tronçon

L'ajout d'un tronçon A'C déclenche le trigger SQL ci-dessus, qui provoque l'ajout d'un pathAggregation pour toutes les topologies sur le tronçon découpé (AB) mais en conservant l'ordre. Comme vu précédemment, si l'ordre est dupliqué, la géométrie de la topologie peut alors être corrompue.

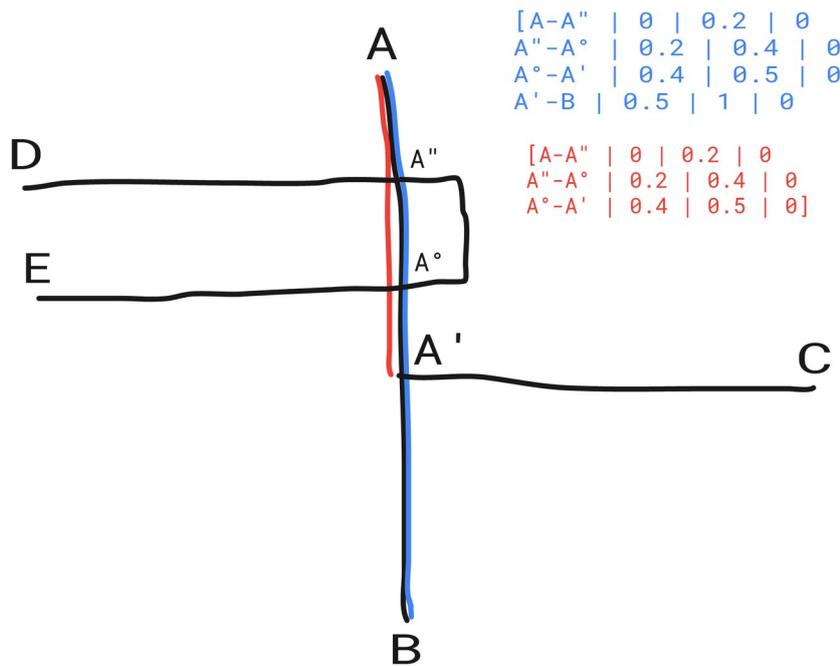


Figure 3: Exemple plus complexe d'un nouveau pathAggregations dupliqué suite à la découpe d'un même tronçon en deux points

Après la découpe du tronçon, un second trigger nommé *post_20_topologies*, vient mettre à jour les topologies liées à ces tronçons.

Lorsqu'un utilisateur souhaite afficher un objet correspondant à une topologie et que celle-ci est linéaire, Geotrek-Admin recrée entièrement la géométrie de la topologie, conséquence du découpage des tronçons et des pathAggregations.

Extrait du code du fichier chargé de la régénération de la topologie

```
-- Regular case: the topology describe a line
-- NOTE: LineMerge and Line_Substring work on X and Y only. If two
-- points in the line have the same X/Y but a different Z, these
-- functions will see only on point. --> No problem in mountain path management.
FOR t_offset, t_geom, t_geom_3d IN SELECT e."offset", ST_SmartLineSubstring(t.geom,
et.start_position, et.end_position),
                                ST_SmartLineSubstring(t.geom_3d, et.start_position,
et.end_position)
FROM core_topology e, core_pathaggregation et, core_path t
WHERE e.id = topology_id AND et.topo_object_id = e.id AND et.path_id = t.id
AND GeometryType(ST_SmartLineSubstring(t.geom, et.start_position,
et.end_position)) != 'POINT'
ORDER BY et."order", et.id -- /\ We suppose that path aggregations were
created in the right order
```

Le trigger chargé de la régénération des topologies utilise l'ordre des pathAggregations.

(Temps passé = 1 Journée)

7 Recherche de solutions

7.1 Avec les triggers existants

Solution A : Trouver le bon ordre lorsqu'on insère un tronçon

Option A1 : En ajoutant « 1 » à l'ordre de tous les pathAggregations situés après la découpe

Option difficile à mettre en place, car dans les triggers, il y a souvent des effets de bord qui peuvent créer des topologies invalides, notamment à cause des points de retours et de passage.

(-> option testée par Makina Corpus)

Option A2 : En rendant l'ordre décimal

A l'insertion de tronçon, mettre la valeur entre l'ordre de celui qu'on découpe et le tronçon suivant (exemple : entre l'ordre 1 et l'ordre 2, le nouveau morceau sera le 1,5).

Il y a beaucoup trop de cas particuliers : points de retour, points de passage, découpe sur points de retour, découpe sur points de passage. (Le point de retour a également un ordre).

Cette option est une bonne idée mais difficile à faire fonctionner dans 100 % des cas (tronçons découpant plusieurs fois le même tronçon).

(-> option testée par Makina Corpus)

Solution B : Récupérer le bon ordre lors de la récupération de la topologie

Option B1 : En retrouvant l'ordre à partir de l'ancienne géométrie

Option difficile à mettre en place si, par exemple, le tronçon est allongé pour découper un autre tronçon sur lequel l'itinéraire passe.



Il faudrait pouvoir inverser le système historiquement en place, en recalculant les pathAggregations à partir de la géométrie, sur le réseau pour retrouver son ordre.

Cette option sera longue à mettre en place et risque d'avoir un impact important sur les performances.

Option B2 : En recréant l'ordre à partir des points de départ et d'arrivée des pathAggregations ayant le même ordre

Le raisonnement serait alors de ce type : si A-fin touche B-début alors on sait que A est avant B , etc. Cette option ne pourrait pas fonctionner pas avec la possibilité de découper un tronçon plusieurs fois avec le même tronçon, ainsi que les topologies qui font un aller-retour. (cf.figure 3).

Toutes ces options sont complexes à développer, cependant si on fait le choix d'empêcher la découpe des tronçons plusieurs fois, la majorité des cas particuliers pourront être gérés.

7.2 Ajouter un nouveau trigger

Une des piste pourrait être d'ajouter un nouveau trigger. Cela pourrait permettre de recalculer l'ordre.

Nous supposons que cela pose le même problème que lorsqu'on récupère la topologie (cf. solutions B).

7.3 Remplacer les triggers SQL par du code Python

Cette option aurait pour objectif de redévelopper tous les calculs effectués relatifs à la base de données lorsqu'on effectue des éditions sur les objets.

Cette option sera de fait très longue et très coûteuse.

De plus, cela engendre un problème dans le cas des ajouts effectués directement dans la base de données. En effet Geotrek-Admin permet de manipuler certaines données en se connectant directement à la base de données : c'est le cas des tronçons. Lorsque l'on ajoute un tronçon dans la base de données, les triggers SQL sont déclenchés. Si on remplaçait les triggers par du code python, ce dernier ne pourrait pas être déclenché lors d'une modification directe de la base de données. Cela engendrerait alors un problème d'intégrité des données entre celles gérées par l'interface et la base de données.

7.4 Conclusion autour des triggers SQL

Les diverses imbrications des triggers SQL, provenant d'un choix d'architecture inhérent à la conception initiale de Geotrek-Admin (pour permettre d'intervenir directement sur la base de données sans passer par l'application), ont engendré de fait une certaine difficulté d'analyse lors du commencement du traitement du problème soulevé par le Département.

Tout au long des phases de test, les solutions envisagées ont nécessité de théoriser le fonctionnement des déclenchements des triggers SQL préalablement à leur mise en place. Il a également fallu dans ce laps de temps implémenter des jeux de tests unitaires pour reproduire les divers scénarii explicités ci-avant.

(Temps passé = 2 Journées)

8 Conclusion des développements et des tests

Lors des phases d'analyses, de développements et de tests, l'ensemble des solutions mentionnées dans les parties 6 et 7 ont été envisagées.

Cependant aucune n'a été jugée suffisamment pertinente, opérationnelle et viable au regard des attentes fonctionnelles de la communauté Geotrek, pour être implémentée dans le code source de Geotrek-Admin.

9 Suite de l'étude

Face à l'inadéquation de toutes les solutions évoquées ci-dessus, il devient nécessaire de chercher à limiter la quantité de cas particuliers existants, ce qui permettrait alors d'améliorer le taux de succès des solutions.

Nous proposons de poursuivre l'étude via l'approche suivante :

1. Faire le bilan des actions réalisables afin de réduire le nombre de cas particuliers, pour créer un contexte plus favorable à la résolution du problème ;
2. Dans ce contexte plus favorable, rejouer tous les tests des solutions évoquées.

Pour réduire les cas particuliers, il conviendra de créer une nouvelle validation du formulaire d'édition de tronçons, qui empêcherait l'utilisateur de découper plusieurs fois un même tronçon. En effet, à ce jour, il semble que la découpe multiple d'un tronçon cause la grande majorité des cas de corruption.

Deux journées d'analyse sont financées par le Parc national des Cévennes pour pousser l'investigation plus loin et tenter d'apporter une ou des solutions plus probables.

Dates de planification des deux journées d'analyse : 9 et 10 novembre 2021.