

Cartographie des chantiers Ville de Québec

Projet #2 :
ModelBuilder et Arcpy

Présenté à:
Véronique Beaulieu et
Jonathan Gagnon

Dans le cadre du cours
GMT-4015
SIG et Analyse spatiale

Par
Équipe 5
Nicolas Massot (537 198 555)
Daniel Bélanger (997 076 988)
Benjamin Roy (111 174 022)

Remise : 28 avril 2024



UNIVERSITÉ
LAVAL

Faculté de foresterie, de géographie
et de géomatique

Table des matières

| | | |
|------|---|----|
| 1. | Introduction et mise en contexte..... | 1 |
| 2. | Présentation des données sources utilisées..... | 2 |
| 3. | Présentation et explication de l'interface..... | 3 |
| 3.1. | Outil 1 : Ajout d'un chantier..... | 3 |
| 3.2. | Outil 2 : Mise à jour des chantiers..... | 3 |
| 3.3. | Outil 3 : Extraire les chantiers d'un quartier dans un csv..... | 3 |
| 3.4. | Outil 4 : Cartographie des chantiers par quartier..... | 4 |
| 4. | Présentation et explication des modèles et des scripts..... | 4 |
| 4.1. | Outil 1..... | 4 |
| 4.2. | Outil 2..... | 5 |
| 4.3. | Outil 3..... | 7 |
| 4.4. | Outil 4..... | 8 |
| 5. | Rétrospective sur les outils, les modèles et les scripts..... | 9 |
| 6. | Annexes..... | 10 |
| 6.1. | Outil 1..... | 10 |
| 6.2. | Outil 2..... | 11 |
| 6.3. | Outil 3..... | 12 |
| 6.4. | Outil 4..... | 13 |

Tables des illustrations

| | | |
|-----------|---------------------------------------|----|
| Figure 1. | Présentation de la boîte à outil..... | 3 |
| Figure 2. | Formulaire outil 1..... | 5 |
| Figure 3. | Formulaire outil 2..... | 6 |
| Figure 4. | Formulaire outil 3..... | 7 |
| Figure 5. | Formulaire outil 4..... | 8 |
| Figure 6. | Modelbuilder de l'outil 1..... | 10 |
| Figure 7. | Script Python de l'outil 2..... | 11 |
| Figure 8. | Script Python de l'outil 3..... | 12 |
| Figure 9. | Script Python de l'outil 4..... | 13 |

1. Introduction et mise en contexte

Mandatés par la ville de Québec, qui souhaite disposer d'un système de gestion des chantiers à partir du logiciel ArcGIS Pro, nous avons développé des outils géomatiques permettant d'aider les administrateurs à optimiser la gestion des chantiers sur le territoire. Pour respecter ce mandat, nous avons créé quatre outils. Ceux-ci se démarquent par leur simplicité d'utilisation. En effet, nous avons considéré, conformément aux instructions reçues, que les utilisateurs du système ne sont pas des spécialistes en géomatique, de ce fait nous avons développé des outils simples d'utilisation. Le présent document fait aussi office de guide d'utilisateur, pour les outils suivants:

- Outil 1 : Outil d'ajout de chantier
- Outil 2 : Outil de mise à jour des chantiers
- Outil 3 : Outil de regroupement tabulaire des chantiers par quartier
- Outil 4 : Outil de cartographie des chantiers par quartier

Dans une ville telle que Québec, plusieurs types de chantiers peuvent affecter la circulation sur les voies publiques. Il est nécessaire de bien définir les types de chantiers, afin de mieux déterminer les restrictions qui en découlent. En s'inspirant [des travaux en cours de réalisation dans la ville de Québec](#), nous avons établi les catégories de chantiers suivantes:

- Réparation de chaussée (ex: remplir nid-de-poule);
- Travaux d'égouts/aqueduc;
- Élargissement/modification des voies
- Installation d'équipements (ex: signalisation)
- Travaux sur le système d'éclairage;
- Travaux sur les utilités publiques (Hydro, Telus, Bell)
- Construction/réfection des ponts et viaducs
- Autres travaux (ex: plantations)

Le premier outil pouvant être programmé en enchaînant une série de géotraitement existants, nous avons alors opté pour Model Builder puisqu'il s'agit de sa principale fonction.

Quant à eux, les outils 2 et 3 nécessitent d'aller faire des modifications dans les enregistrements des tables attributaires. C'est pour cette raison qu'ils ont été construits à l'aide de scripts Python. En effet, les curseurs sont des objets qui facilitent grandement la modification des tables par rapport aux Model Builder, apportant par le fait même une souplesse d'utilisation absente de ce dernier au niveau de la modification des enregistrements.

L'outil 4 a uniquement été effectué grâce au script python. Nous utilisons cette méthode pour faciliter la cartographie automatisée par ordinateur à l'aide d'ArcPy, en utilisant les attributs et méthodes du module mp.

2. Présentation des données sources utilisées

Notre mandataire étant la ville de Québec, il est naturel d'utiliser comme source le site Données Ouvertes Québec, qui est la source de donnée officielle du gouvernement du Québec. Afin de remplir notre mandat, nous avons utilisé les fichiers de forme correspondant aux voies de circulation ainsi que celle des quartiers de la ville. Il est alors possible de localiser et d'agréger les chantiers par quartiers.

Les données originales étant référencées en WGS 84, nous avons procédé à la transformation des couches dans le système de référence demandé, qui est le NAD83 CSRS MTM7. C'est dans ce dernier système que tous les outils présentés opèrent.

| # Donnée | Nom de la donnée | Produit/producteur | Système de coordonnées | Format | Hyperlien |
|----------|------------------|--------------------|------------------------|--------|----------------------|
| 1 | Quartiers | Ville de Québec | WGS 84 | shp | Lien |
| 2 | Routes | Ville de Québec | WGS 84 | shp | Lien |

Tableau 1. Source des données du projet.

3. Présentation et explication de l'interface

La boîte outil que nous avons développée apparaît ainsi dans le navigateur ArcGis Pro.

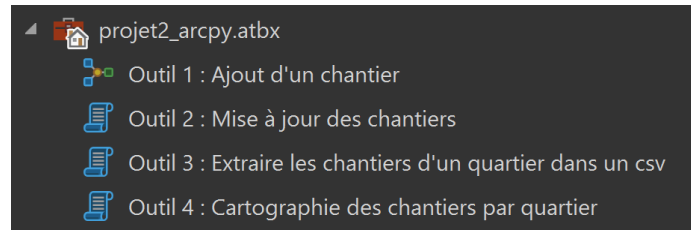


Figure 1. Présentation de la boîte à outils.

3.1.Outil 1 : Ajout d'un chantier

L'outil 1 sert principalement à géocoder de nouveaux chantiers en cours sur le territoire de la ville de Québec à l'aide de leurs adresses. Il permet également de leur assigner des informations descriptives telles que le type de travaux, les restrictions mises en place, ainsi que la date de début et de fin. Ces informations sont stockées dans la table nommée « chantier ». Il est ainsi possible de les visualiser dans la carte de ArcGis Pro.

3.2.Outil 2 : Mise à jour des chantiers

L'outil 2 a deux objectifs. Le premier est de mettre à jour le type de restriction, ainsi que de prolonger ou réduire la date de fin du chantier. Notons que nous sélectionnons un chantier avec l'OBJECTID. Les informations qui ne sont pas modifiées demeureront inchangées par rapport à l'entrée initiale du chantier. Le second objectif de cet outil est de vérifier s'il existe dans la base de données des chantiers qui sont terminés. Pour ce faire, l'outil parcourt tous les chantiers et vérifie si la date de fin est dépassée ou non. Si elle est dépassée, le chantier est supprimé.

3.3.Outil 3 : Extraire les chantiers d'un quartier dans un csv

L'outil 3 permet de lister les chantiers en cours dans un quartier donné. Ainsi, la sortie de cet algorithme est une table au format .csv, ce qui permet de rendre accessible les informations descriptives à tout un chacun, notamment à ceux qui ne sont pas familiarisés avec les SIG.

3.4.Outil 4 : Cartographie des chantiers par quartier

L'outil 4 sert à cartographier les travaux en cours dans les différents quartiers de la ville. Il permet de visualiser les chantiers d'un quartier donné. Pour faciliter l'utilisation, la cartographie engendrée peut être exportée sous forme d'un document PDF vectoriel.

Pour tous ces outils, plusieurs informations pertinentes sont aussi présentes dans les métadonnées de chacun des outils. Ces informations sont présentes sous forme de bulles d'informations.

4. Présentation et explication des modèles et des scripts

4.1.Outil 1

L'objectif de cet outil est de localiser et de lister les différents chantiers en cours dans la ville, tout en considérant le type de travail effectué, les restrictions à la circulation qui seront engendrées et les dates de début et de fin.

Pour faire fonctionner ce premier outil, l'utilisateur doit simplement accéder à la boîte de dialogue et entrer manuellement les informations demandées soit ;

- L'adresse du chantier (adresses civiques, le nom de la rue et ville)
- Les types de restriction
- Les types de travaux
- Les dates de début et de fin du chantier

Les types de restriction et de travaux sont déjà disponibles dans les listes de choix prédéfinis, tandis que les dates de début et de fin du chantier sont facilement sélectionnables avec l'utilisation du calendrier. Les bandes déroulantes ainsi que les calendriers faciliteront grandement l'utilisation de l'interface par un néophyte. De plus, pour répondre aux questions potentielles de l'utilisateur, des bulles d'informations sont disposées à chaque entrée de paramètre et permettent de visualiser ce qui est attendu.

Ainsi, l'interface de l'outil correspond à la figure 1 ci-dessous. Les informations munies d'un astérisque rouge doivent obligatoirement être inscrites pour pouvoir démarrer la recherche. Certaines autres informations complémentaires à l'outil sont présentées dans les métadonnées de l'outil.

The figure consists of two side-by-side screenshots of a mobile application interface titled 'Ajout d'un chantier' (Add a site) under the 'Geoprocessing' header. Both screenshots show a 'Parameters' tab with five input fields, each marked with a red asterisk (*). The left screenshot shows the form with empty fields. The right screenshot shows the form with the following data entered: 'Adresse_' is '17 Rue du Fort Québec', 'Travaux' is 'Élargissement/modification des voies', 'Restriction_' is 'Voies réduites', 'Date de début_' is '10/04/2024 01:16:09', and 'Date de fin_' is '26/04/2024 01:16:15'. Each date field has a clock icon for selection.

Figure 2. Formulaire outil 1.

Par exemple, si un chantier d'élargissement des voies prend place au 17 Rue du Fort à Québec cause une restriction de réductions des voies, débute le 10 mars et se termine le 26 mars, l'utilisateur devra inscrire ces informations dans le formulaire comme indiqué dans la figure 2 ci-dessus.

Une fois les informations entrées et exécutées, le procédé engendre une nouvelle entrée dans la table nommée « chantier ». Chaque entrée se voit attribuer un OBJECTID unique qui permettra d'identifier le chantier. De plus, grâce à l'adresse mentionnée lors de l'inscription des chantiers, l'exécution localise le lieu de cette entrée et enregistre les coordonnées X et Y du chantier dans des champs dédiés.

4.2.Outil 2

Ce deuxième outil fonctionne de pair avec le premier outil. Il s'agit ici de pouvoir modifier et supprimer les chantiers précédemment enregistrés avec l'outil 1. Pour ce faire, la boîte de dialogue de cet outil requiert quelques informations :

- « Chantier » : qui correspond au chemin d'accès jusqu'à la classe d'entités chantier créer à l'étape précédente.

- « OBJECTID » : qui correspond au OBJECTID des chantiers créés dans la table champs
- « Restriction » : permet de modifier la restriction du chantier sélectionné
- « Date_fin » qui permet de modifier la date de fin du chantier sélectionné
- « Supprimer les chantiers terminés » : l'option de supprimer les chantiers terminés permet de mettre à jour la table chantier dans le but de supprimer les chantiers périmés au moment de la mise à jour

Le champ restriction est toujours muni d'une bande déroulante permettant de sélectionner les types de restriction. La sélection de la date de fin est aussi simplifiée par la présence d'un calendrier virtuel. Dans ce formulaire, à l'exception du chemin d'accès de la table chantier, aucune information n'est obligatoire ce qui permet de modifier uniquement un élément à la fois. Il est aussi possible de modifier simultanément le type de restriction et la date de fin des travaux. Dans le but de supprimer uniquement les chantiers terminés lors de la mise à jour, sans modifier de chantier en cours, il suffit de seulement cocher la case « Supprimer les chantiers ». Pour faciliter l'utilisation de l'outil, chaque case est munie d'une bulle d'information qui rappelle à l'utilisateur ce qui est attendu d'inscrire à chaque ligne du formulaire. D'autres informations complémentaires sont aussi présentes dans les métadonnées de l'outil.

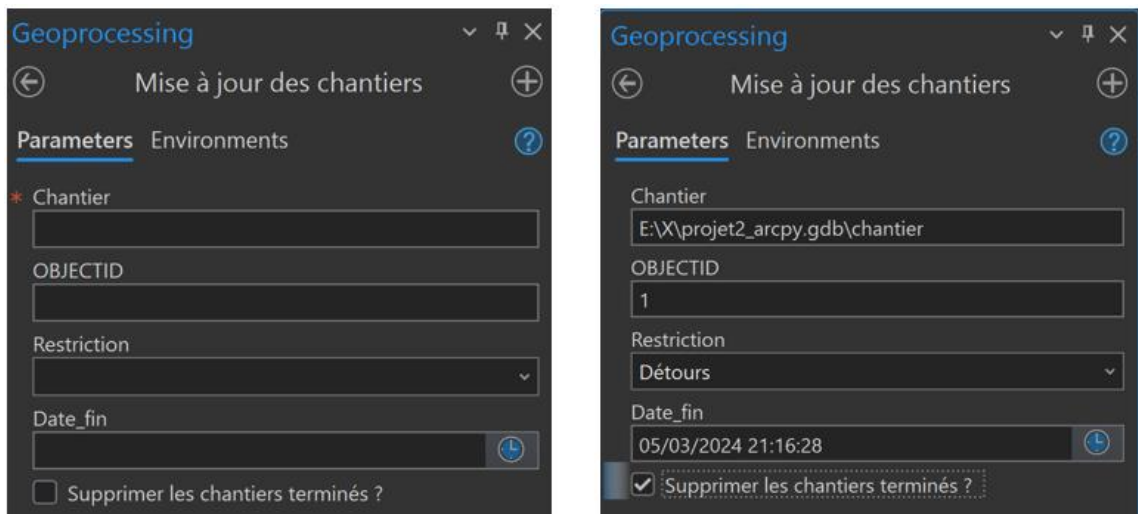


Figure 3. Formulaire outil 2.

Par exemple, pour notre table chantier située dans la gdb projet2_arcpy du dossier X de notre ordinateur. En plus de supprimer les chantiers terminés en cochant la case « Supprimer les chantiers ». Pour le chantier dont l'OBJECTID est 1, nous désirons modifier le type de restrictions pour « Détours » ainsi que la date de fin du chantier pour

le 5 mars 2024. L'utilisateur doit inscrire ces informations comme illustrées dans la figure 2 à droite.

La résultante de l'exécution de cet outil est la modification de la table chantier créer précédemment à l'outil 1. Ainsi, les chantiers expirés peuvent être retirés en sélectionnant la case « Supprimer les chantiers ». De plus, les restrictions ainsi que la date de fin de chaque chantier inscrit peuvent être modifiées.

4.3.Outil 3

Ce troisième outil a comme objectif d'engendrer un tableau regroupant les différents travaux en cours selon le quartier demandé. Pour atteindre ce résultat, l'utilisateur doit inscrire certaines informations ;

- Le chemin d'accès du fichier quartier sur son ordinateur
- Le chemin d'accès du fichier chantier sur son ordinateur
- Le nom du quartier pour lequel il souhaite obtenir la table regroupant les différents chantiers

Toutes ces informations sont obligatoires pour faire fonctionner l'outil.

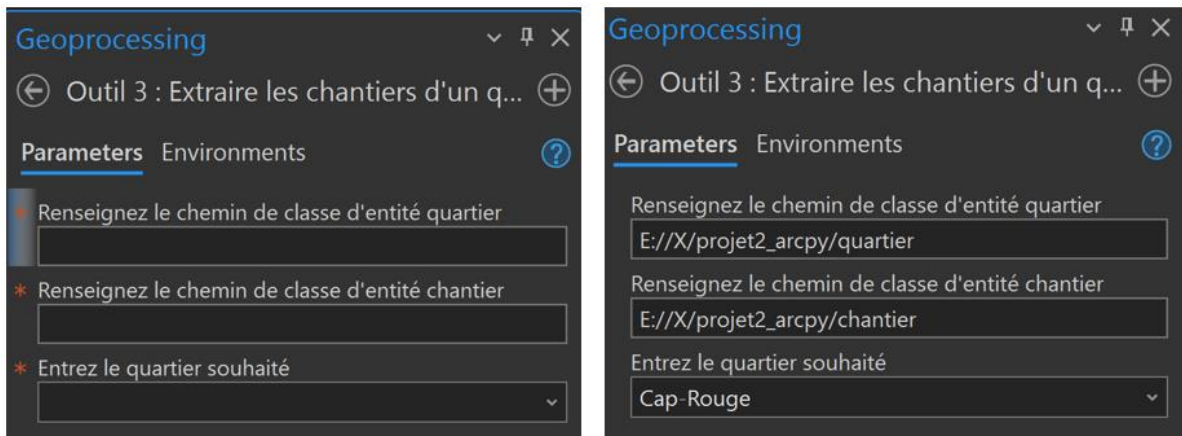


Figure 4. Formulaire outil 3.

Par exemple, pour notre table quartiers et chantier tous deux situés dans la gdb projet2_arcpy du dossier X de notre ordinateur. L'utilisateur doit sélectionner, dans le menu déroulant, le nom du quartier pour lequel il souhaite consulter la liste des chantiers en cours, dans notre exemple nous avons choisi le quartier Cap-Rouge.

L'exécution de cet outil engendre une table CSV. qui cumule tous les chantiers en cours dans le quartier demandé. La table est consultable directement dans ArcGIS Pro, mais

peut aussi être exportée dans n'importe quel format choisi par l'utilisateur. Lorsqu'utiliser sur le même poste de travail, les deux premiers renseignements à fournir demeureront toujours identiques. Il ne reste qu'à sélectionner à l'aide du menu déroulant le quartier qu'on souhaite visualiser. La table engendrée se nomme chantier_dans_quartier_selectionne.

4.4.Outil 4

Ce quatrième outil sert à cartographier rapidement les travaux en cours dans les différents quartiers de la ville. L'outil permet de visualiser les chantiers en cours pour un quartier à la fois. Pour faciliter l'utilisation, la cartographie engendrée est envoyée sous forme de document PDF. Pour ce faire, certaines informations doivent être inscrites dans les paramètres de l'outil par l'utilisateur soit ;

- Si Oui ou Non on souhaite exporter la visualisation des chantiers en PDF
- Le chemin d'accès vers le dossier de sortie de la carte PDF
- Le quartier à visualiser

L'option d'exporter la carte en format est très simple grâce à l'option de type booléen du formulaire. De plus, pour faciliter la sélection du quartier désiré, un menu déroulant permet à l'utilisateur de sélectionner les 35 différents quartiers de la ville de Québec.

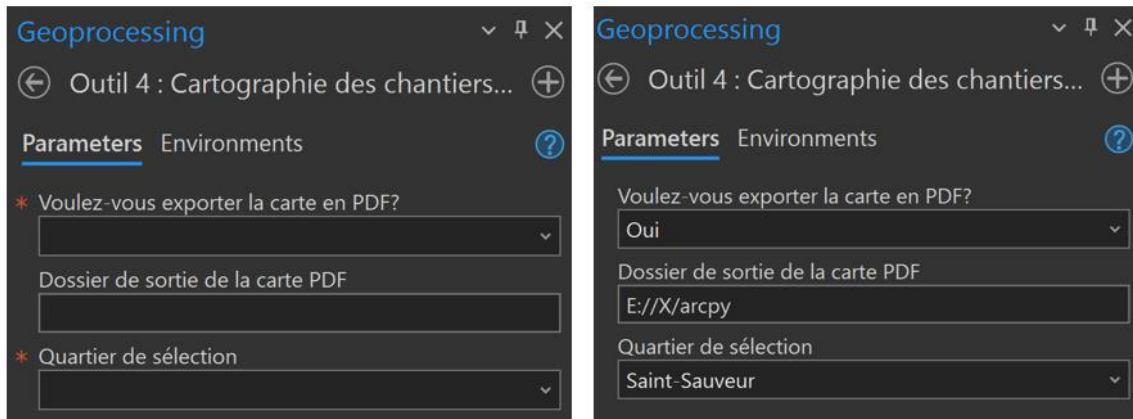


Figure 5. Formulaire outil 4.

Dans le cas illustré ci-dessus, l'utilisateur souhaite obtenir la carte PDF, dans le fichier arcpy de son dossier X, illustrant les chantiers en cours dans le quartier Saint-Sauveur.

Le résultat de l'exécution de cet outil est une carte PDF qui présente uniquement les chantiers en cours dans le quartier sélectionné.

5. Rétrospective sur les outils, les modèles et les scripts.

Sous certains aspects, le développement des outils prévus a été plus ardu qu'anticipé. En effet, l'utilisation de scripts python, qui permet une flexibilité et des possibilités non égalées par Model Builder, nécessite une connaissance approfondie des modules, objets et fonctions d'arcpy qui nous font un peu défaut. La résultante est que nos outils sont fonctionnels, mais, pourraient forcément être améliorés après le déploiement chez le client, soit par raffinement des fonctions existantes, soit par l'ajout de nouvelles fonctionnalités.

Par exemple, l'outil de géocodage des chantiers a dû être travaillé différemment dans le but de permettre la concaténation des différents champs. Cet appariement permet la localisation des chantiers par le géocodeur.

L'outil 2 pourrait être amélioré en ajoutant l'option de supprimer rapidement n'importe quel chantier, sans se soucier de sa date de fin prévue.

L'outil 3 pourrait être amélioré en ajoutant l'option de sélectionner les chantiers selon le type de travaux effectué. Cela permettrait par exemple de visualiser tous les chantiers en cours qui affectent le système d'éclairage de la ville. Aussi, nous souhaitons avoir un nom de table référent au nom du quartier sélectionné cependant, la concaténation du nom des quartiers comportant un tiret échouait. La solution a été de nommer toutes les tables résultantes.

Au terme de cet exercice, nous croyons donc avoir rencontré la plupart des spécifications demandées, mais pas toutes, et il va sans dire que même les outils qui sont fonctionnels gagneraient à être améliorés.

6. Annexes

6.1.Outil 1

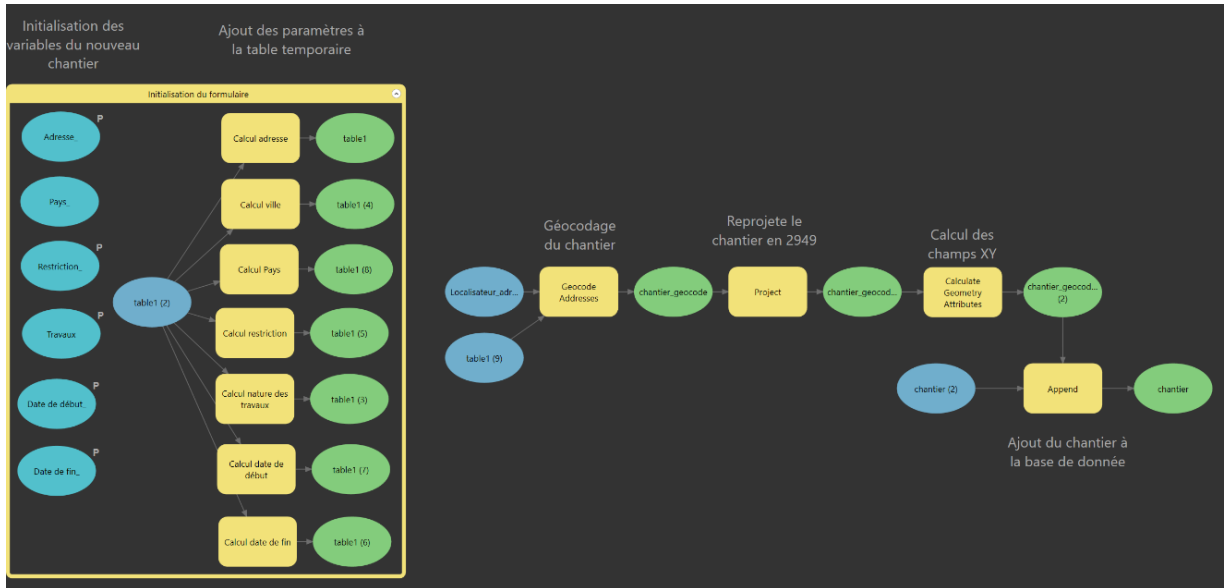


Figure 6. Modelbuilder de l'outil 1

6.2.Outil 2

```
1 # Importation du module arcpy pour la manipulation de données géospatiales
2 import arcpy
3 # Importation de la classe datetime du module datetime pour la gestion des dates
4 from datetime import datetime
5
6 # Définition de la fonction pour mettre à jour et supprimer des chantiers
7 def update_and_delete_chantiers(chantier, OBJECTID, Restriction=None, Date_fin=None, delete_expired=False):
8     """
9     Met à jour les champs 'Restriction' et 'Date_fin' pour un chantier spécifique identifié par son numéro,
10    permettant à l'utilisateur de laisser des valeurs vides pour ne pas les mettre à jour.
11    Supprime les travaux qui sont terminés si cette fonction est activée.
12
13    Args:
14        chantier (str): Chemin vers la classe d'entités 'chantiers'.
15        OBJECTID (str): Numéro du chantier à mettre à jour.
16        Restriction (str): Nouvelle valeur pour le champ 'Restriction', optionnel.
17        Date_fin (str): Nouvelle valeur pour le champ 'Date_fin', optionnel.
18        delete_expired(boolean): Permet de supprimer les travaux terminés après avoir comparé les dates de fin du travaux avec la date du jours.
19    """
20
21    # Convertir OBJECTID en entier si possible, sinon None
22    objectid_int = int(OBJECTID) if OBJECTID else None
23    # Liste des champs à manipuler
24    fields = ['OBJECTID', 'Restriction', 'Date_fin']
25    # Date du jour
26    today = datetime.now()
27    # Variable pour suivre si un chantier a été mis à jour
28    updated = False
29    # Compteur pour le nombre de chantiers supprimés
30    deleted_count = 0
31
32    # Ouvrir un curseur de mise à jour sur la couche chantier
33    with arcpy.da.UpdateCursor(chantier, fields) as cursor:
34        for row in cursor:
35            # Mise à jour du chantier spécifié par OBJECTID
36            if objectid_int and row[0] == objectid_int:
37                if Restriction not in (None, '', '#', ' '):
38                    row[1] = Restriction
39                if Date_fin not in (None, '', '#', ' '):
40                    row[2] = Date_fin
41                cursor.updateRow(row)
42                # Message de succès pour la mise à jour
43                arcpy.AddMessage(f"Chantier {OBJECTID} mis à jour avec succès.")
44                updated = True
45
46            # Suppression des chantiers dont la date de fin est dépassée
47            if delete_expired and row[2]:
48                # Convertir la date de fin en datetime si c'est une chaîne de caractères
49                date_fin_chantier = datetime.strptime(row[2], "%d/%m/%Y") if isinstance(row[2], str) else row[2]
50                if date_fin_chantier < today:
51                    cursor.deleteRow()
52                    deleted_count += 1
53
54    # Messages finaux
55    if updated:
56        arcpy.AddMessage(f"Chantier {OBJECTID} a été mis à jour.")
57    if delete_expired:
58        arcpy.AddMessage(f"{deleted_count} chantier(s) dont la date de fin est dépassée ont été supprimés.")
59
60 # Point d'entrée du script
61 if __name__ == "__main__":
62     # Récupération des paramètres en entrée
63     chantier = arcpy.GetParameterAsText(0)
64     OBJECTID = arcpy.GetParameterAsText(1) # Peut être vide si la mise à jour n'est pas nécessaire
65     Restriction = arcpy.GetParameterAsText(2)
66     Date_fin = arcpy.GetParameterAsText(3)
67     delete_expired = arcpy.GetParameterAsText(4) == 'true'
68
69     # Appel de la fonction avec les paramètres récupérés
70     update_and_delete_chantiers(chantier, OBJECTID, Restriction, Date_fin, delete_expired)
```

Figure 7. Script Python de l'outil 2

6.3.Outil 3

```
1 import arcpy
2
3 def outil3(quartier, chantier, quartier_selection) :
4     """
5     Cette fonction sert à sélectionner les chantiers qui se situent dans un quartier sélectionné, et de les exporter dans une table .csv
6
7     quartier = la classe d'entité qui comporte les quartiers
8     chantier = la classe d'entité qui comporte les chantiers
9     quartier_selection = quartier sélectionné par l'utilisateur parmi la liste des quartiers
10    """
11
12    arcpy.env.overwriteOutput = True
13
14    #Permet de concaténer la variable quartier_selection avec des éléments qui permettent de construire une requête SQL
15    concat = "g_nom =" + quartier_selection + ""
16
17    #Permet de sélectionner le quartier choisi avec notre variable concat
18    select_quartier = arcpy.management.SelectLayerByAttribute(quartier, 'NEW_SELECTION', concat)
19
20    #Permet de sélectionner les chantiers qui intersectent le quartier sélectionné
21    export = arcpy.management.SelectLayerByLocation(chantier, 'INTERSECT', select_quartier)
22
23    #Initialise le nom de mon fichier
24    #nom_sortie = "chantiers_dans_" + quartier_selection
25
26    #Permet d'exporter la table à l'endroit souhaité
27    arcpy.conversion.ExportTable(export, 'chantier_dans_quartier_selectionne', use_field_alias_as_name='USE_ALIAS')
28
29    #Définit les paramètres
30    if __name__ == "__main__":
31        quartier=arcpy.GetParameterAsText(0)
32        chantier=arcpy.GetParameterAsText(1)
33        quartier_selection=arcpy.GetParameterAsText(2)
34
35        #Appelle les fonctions
36        outil3(quartier, chantier, quartier_selection)
```

Figure 8. Script Python de l'outil 3.

6.4.Outil 4

```
1 import arcpy
2
3
4 def cartoautomatique(export, sortie, quartier_selectionne):
5     projet = arcpy.mp.ArcGISProject("CURRENT") # importe les variables d'environnement
6     map = projet.listMaps('Map')[0]
7     pages = projet.listLayouts('Layout')[0]
8
9     chantier = map.listLayers('chantier')[0] # importe les variables d'initialisation
10    quartier = map.listLayers('quartiers')[0]
11    chantier_quartier = map.listLayers('chantier_quartier')[0]
12
13
14    # Sélectionne le quartier désiré
15    quartier_temp = arcpy.management.SelectLayerByAttribute(quartier, 'NEW_SELECTION', "\"" + quartier_selectionne + "\"")
16
17    # Sélectionne les chantiers qui intersectent le quartier
18    couche_temp = arcpy.management.SelectLayerByLocation(chantier, 'INTERSECT', quartier_temp, selection_type='NEW_SELECTION')
19
20    # Ajoute les chantiers sélectionnés dans la couche 'chantier_quartier'
21    arcpy.management.Append(couche_temp, chantier_quartier, 'NO_TEST', update_geometry='UPDATE_GEOMETRY')
22
23    # Supprime la sélection des quartiers et des chantiers
24    arcpy.management.SelectLayerByAttribute(quartier, 'CLEAR_SELECTION')
25    arcpy.management.SelectLayerByAttribute(chantier, 'CLEAR_SELECTION')
26
27    #Tentative infructueuse d'améliorer le zoom sur les chantiers du quartier
28    #cam = projet.activeView.camera
29    #ext = arcpy.Describe(chantier_quartier).extent
30    #cam.setExtent(ext)
31
32    if export == 'Oui':
33        chemin_sortie = sortie + "/chantier dans " + quartier_selectionne + ".pdf"
34        pages.exportToPDF(chemin_sortie)
35
36    # Sélectionne les chantiers dans la table chantier_quartier, 'Canada' étant une sélection générique à tous les chantiers
37    #arcpy.management.SelectLayerByAttribute(chantier_quartier, 'NEW_SELECTION', "\"Pays\" = 'Canada'")
38    # Supprime les entités sélectionnées
39    #arcpy.management.DeleteFeatures(chantier_quartier)
40
41
42    # Définit les paramètres
43    if __name__ == "__main__":
44        export = arcpy.GetParameter(0)
45        sortie = arcpy.GetParameterAsText(1)
46        quartier_selectionne = arcpy.GetParameterAsText(2)
47
48        # Appelle la fonction
49        cartoautomatique(export, sortie, quartier_selectionne)
```

Figure 9. Script Python de l'outil 4.