

BAILLY Éric
Université Nice Sophia-Antipolis
Centre de la Méditerranée Moderne et
Contemporaine (EA 1193)

VIGNAL Matthieu
Université Nice Sophia-Antipolis

BALME Thibaut
Université Nice Sophia-Antipolis

Les itérateurs du ModelBuilder dans Arcgis 10

Arcgis propose trois méthodes pour automatiser des opérations :

La première consiste à utiliser la fonction « par lot ». Elle est présente pour chaque élément de l'arc toolbox. L'utilisateur paramètre l'outil afin que l'action soit réalisée sur plusieurs couches à la suite. Cependant, cette opération est longue et fastidieuse puisqu'il faut configurer manuellement les processus un à un.

La deuxième méthode utilise la programmation. En effet Arcgis est scriptable à l'aide du langage Python. Néanmoins, pour employer cette fonctionnalité, l'utilisateur doit maîtriser ce dernier.

La troisième méthode fournie par Arcgis est le ModelBuilder. Cet outil, à travers une interface iconographique, permet de réaliser des séries d'opérations. Toutefois, le reproche principal que l'on pouvait lui faire jusqu'à présent résidait dans la difficulté à réitérer des opérations automatiquement. La version 10 d'Arcgis apporte cette fonctionnalité. Ainsi, le jeu d'outils Itérateur comporte douze opérateurs qui permettent de répéter un ou plusieurs processus sur un ensemble d'entrées.

Cet article à caractère pédagogique classe ces nouvelles fonctionnalités du ModelBuilder en trois parties et les illustre par des exemples simples. L'ensemble des opérations est détaillé dans la documentation officielle d'Arcgis :

<http://help.arcgis.com/fr/arcgisdesktop/10.0/help/index.html#/na/004000000000n0000000>

1 Les itérateurs classiques de programmation

Deux itérateurs proposés par ModelBuilder sont similaires à ce que l'on trouve dans l'élémentaire de tout langage de programmation. Le "Pour" et le "Tant que" correspondent respectivement aux boucles "for" et "while" et fonctionnent identiquement.

L'utilisateur fournit trois arguments à l'itérateur "Pour" : "Valeur début", "Valeur fin" et "Par valeur". L'itérateur "Pour" incrémente une variable nommée "Valeur", en débutant de l'argument "Valeur début", jusqu'à l'argument "Valeur fin" par pas de "Par valeur". Il est ainsi

possible d'effectuer plusieurs fois une ou des opérations. Le nombre de réitérations doit être connu au départ.

En Python (le langage de programmation d'Arcgis), il correspond à la séquence suivante :

" For Valeur in Range (Valeur_début, Valeur_fin, Par_valeur) : ".

Voici un exemple d'utilisation de cet itérateur. Nous allons créer des zones tampons autour de voies routières, dans le but de déterminer différents impacts à proximité de ces routes.

L'itérateur "Pour" est paramétré afin qu'il répète l'action de création de zones tampons 5 fois, sur des distances variant de 10 (valeur de début) à 50 (valeur de fin) suivant un pas de 10. L'unité est le mètre. Nous obtenons donc 5 couches représentant un buffer de valeurs 10, 20, 30, 40 et 50 mètres.

Le nom des couches récupérées en sortie peut aussi être automatisé, en utilisant l'expression « % Valeur% ». La taille du tampon apparaît ainsi dans chaque nom de couche créée.

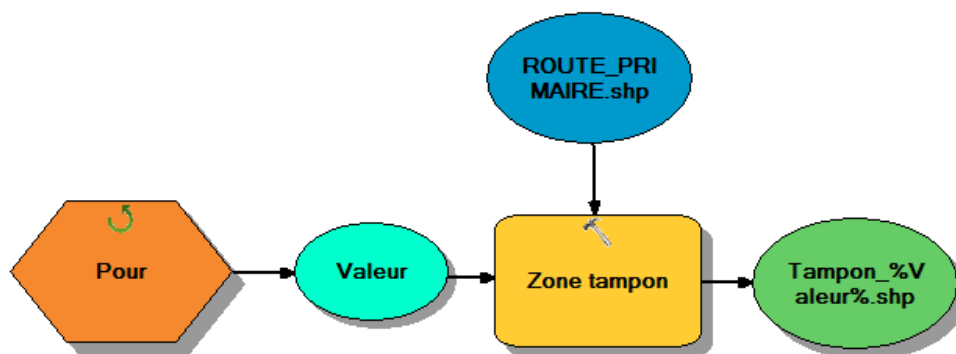


Figure 1 : Processus de l'itérateur "Pour"

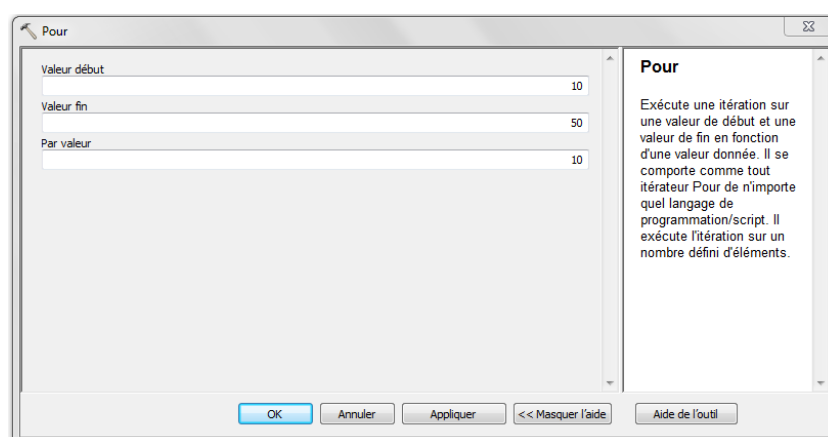


Figure 2 : Boîte de dialogue d'entrée des variables de l'itérateur "Pour"

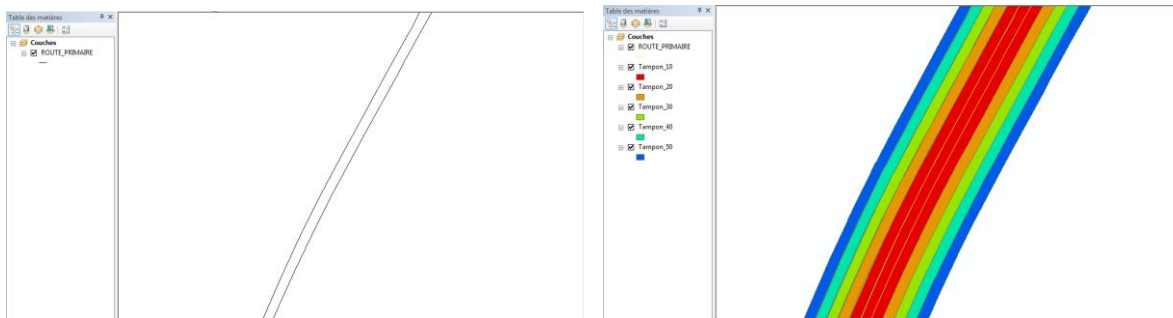


Figure 3 : le résultat des 'tampons'

2 les itérateurs d'objets dans une couche

Cette deuxième catégorie d'itérateurs travaille avec les éléments constitutifs d'une couche. Ils sont au nombre de 4 :

- "itérer la sélection" utilise des entités appartenant à une classe,
- "itérer la sélection de ligne" accède aux lignes d'une table,
- "itérer les valeurs de champs" exploite les valeurs d'un champ,
- "itérer les valeurs multiples" manipule une liste de valeur.

Notre démonstration porte sur l'itérateur « Itérer la sélection d'entité ». Il réitère une ou plusieurs opérations pour chaque entité d'une classe d'entités. L'exemple qui suit crée un ensemble de couches d'après une sélection au sein d'une classe d'entités. À partir de la couche "communes.shp" contenue dans la base de données GEOFLA de l'IGN¹, comprenant les 22 régions françaises, une couche distincte pour chaque région est créée. L'itérateur passe en revue le champ "CODE_REG" de la couche et renvoie son contenu (variable "Valeur") et l'entité associée. Nous obtenons alors 22 couches distinctes pour les 22 régions de France métropolitaine.

Ce type d'automatisation est intéressant lorsqu'il est nécessaire de posséder une couche différente pour chaque valeur. Les gains de temps et de productivité sont considérables par rapport à l'exportation par sélection, procédure basique dans Arcgis.

Cet itérateur génère donc deux sorties : l'entité extraite et la valeur du champ. Pour obtenir une couche contenant l'entité extraite, il convient d'utiliser l'outil « générer une couche ». La valeur du champ peut être utilisée pour nommer cette couche comme nous l'avons présenté dans le premier exemple, par le biais de l'opération "copier des entités".

¹ <http://professionnels.ign.fr/geofla>, fichier GEOFLA_1-1_SHP_LAMB93_FR-ED131

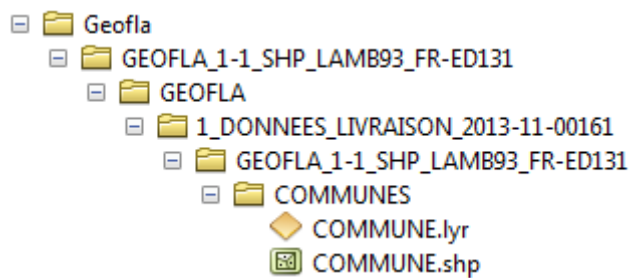




Figure 4 : Organisation de la base GEOFLA

Table



COMMUNE

FID	Shape	ID GEOFLA	CODE COMM	INSEE COM	NOM COMM	CODE REG	NOM REGION
0	Polygone	1 013	65013	ANSOST	73	MIDI-PYRENEES	
1	Polygone	2 152	30152	LES IMAGES	91	LANGUEDOC-ROUSSILLON	
2	Polygone	3 294	61294	MORTREE	25	BASSE-NORMANDIE	
3	Polygone	4 044	47044	CAHUZAC	72	AQUITAINE	
4	Polygone	5 272	47272	SAINT-QUENTIN-DU-DROPT	72	AQUITAINE	
5	Polygone	6 373	24373	SAINT-AUBIN-DE-CADELECH	72	AQUITAINE	
6	Polygone	7 086	48086	LUC	91	LANGUEDOC-ROUSSILLON	
7	Polygone	8 167	32167	LAAS	73	MIDI-PYRENEES	
8	Polygone	9 165	88165	ETVAL-CLAIREFONTAINE	41	LORRAINE	
9	Polygone	10 131	67131	ESCHAU	42	ALSACE	
10	Polygone	11 130	32130	FAGET-ABBATIAL	73	MIDI-PYRENEES	
11	Polygone	12 411	67411	ROSHEN	42	ALSACE	
12	Polygone	13 236	40236	POYARTIN	72	AQUITAINE	
13	Polygone	14 480	64480	SAINT-GLADIE-ARRIVE-MUNEN	72	AQUITAINE	
14	Polygone	15 305	34305	LE SOULIE	91	LANGUEDOC-ROUSSILLON	
15	Polygone	16 271	40271	SAINTE-MARIE-DE-GOSSE	72	AQUITAINE	
16	Polygone	17 226	38226	MENS	82	RHONE-ALPES	
17	Polygone	18 096	73096	CRUET	82	RHONE-ALPES	
18	Polygone	19 398	71398	SAINT-CHRISTOPHE-EN-BRESSE	26	BOURGOGNE	
19	Polygone	20 302	38302	LE PERIER	82	RHONE-ALPES	
20	Polygone	21 462	32462	VIC-FEZENSAC	73	MIDI-PYRENEES	
21	Polygone	22 199	39199	DOMBLANS	43	FRANCHE-COMTE	
22	Polygone	23 245	32245	MAUMUSSON-LAGUIAN	73	MIDI-PYRENEES	
23	Polygone	24 394	33394	SAINT-EMILION	72	AQUITAINE	
24	Polygone	25 198	26198	MONTELMAR	82	RHONE-ALPES	
25	Polygone	26 254	11254	MONTREAL	91	LANGUEDOC-ROUSSILLON	
26	Polygone	27 253	40253	SAINT-CRICO-CHALOSSE	72	AQUITAINE	
27	Polygone	28 138	46138	LABASTIDE-MURAT	73	MIDI-PYRENEES	
28	Polygone	29 182	46182	LUZEC	73	MIDI-PYRENEES	
29	Polygone	30 408	55408	POUILLY-SUR-MEUSE	41	LORRAINE	
30	Polygone	31 410	17410	SAINT-THOMAS-DE-CONAC	54	POITOU-CHARENTES	
31	Polygone	32 140	15140	NAUCELLES	83	AUVERGNE	



1 (0 sur 36594 sélectionnés)

COMMUNE

Figure 5 : La table attributaire de GEOFLA

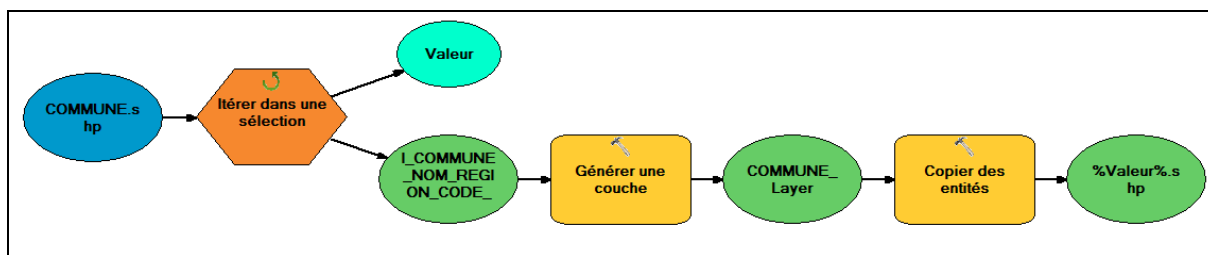


Figure 6 : Processus de l'itérateur « Itérer la sélection d'entité ».

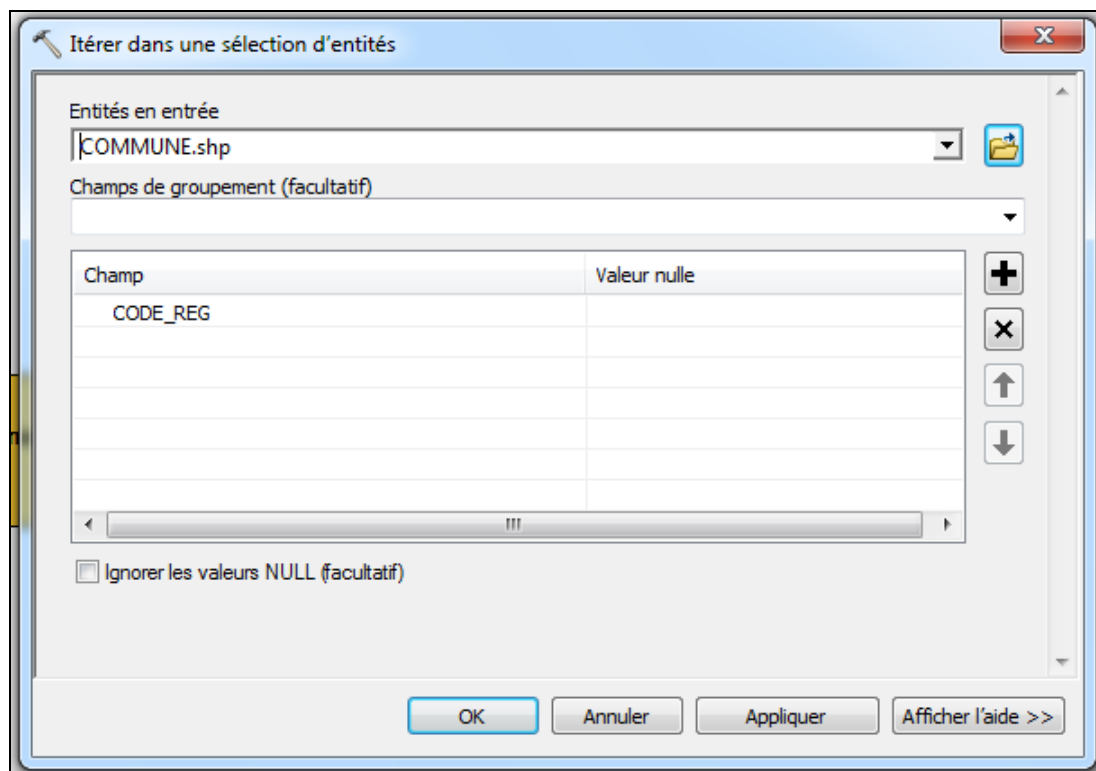


Figure 7 : boîte de sélection pour l'itérateur « Itérer la sélection d'entité ».

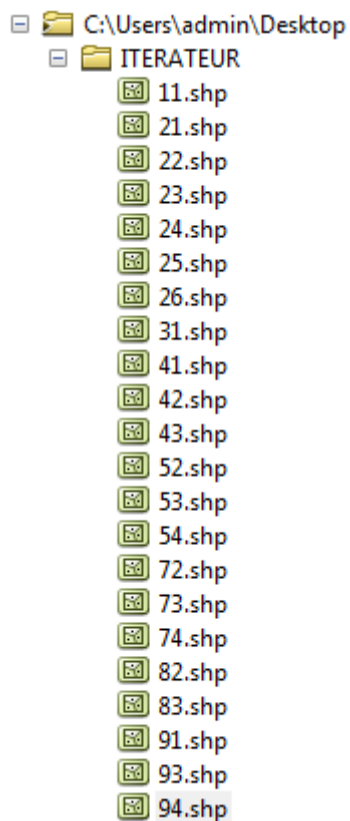


Figure 4 : Le résultat avec un shape par région

3 Les itérateurs liés aux espaces de travail et dossiers

Dans cette troisième catégorie, l'utilisateur accède aux informations contenues dans les espaces de travail sous la forme de dossiers informatiques. On trouve 6 itérateurs correspondants aux jeux de données, classes d'entité, fichiers, rasters, tables et espaces de travail. Il est ainsi possible d'automatiser de nombreuses tâches.

L'itérateur « Itérer les rasters » exécute une opération sur l'ensemble des fichiers rasters contenus dans un dossier ou un catalogue d'images². Comme dans les exemples précédents, il sélectionne une à une les données (ici les fichiers rasters), permettant ensuite d'effectuer toutes les opérations voulues individuellement.

Définissons par exemple une projection à tous les fichiers rasters contenus dans notre dossier d'exercices ("Test_Raster"). L'itérateur génère en sortie un fichier raster et son nom. La définition de la projection (ici le LAMBERT 93) est indiquée grâce à l'outil « Définir Projection ». Cette opération est répétée pour l'ensemble des classes d'entités tout en gardant le nom de départ de la couche raster.

L'outil permet aussi de ne sélectionner que les fichiers rasters d'un certain type (option format de raster, ASCII dans notre exemple).

Si l'option "Récursif" est cochée, les éléments contenus dans les sous-dossiers de "Test_Raster" seront analysés.

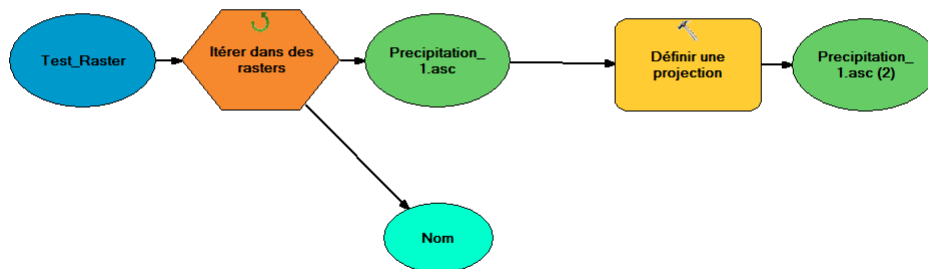


Figure 5 : Processus

² "Un catalogue d'images est un ensemble de jeux de données rasters défini sous un format de table dans lequel chaque enregistrement représente un jeu de données rasters du catalogue. Un catalogue peut contenir des milliers d'images." <http://resources.arcgis.com/fr/help/main/10.1/index.html#//009t0000000n000000>

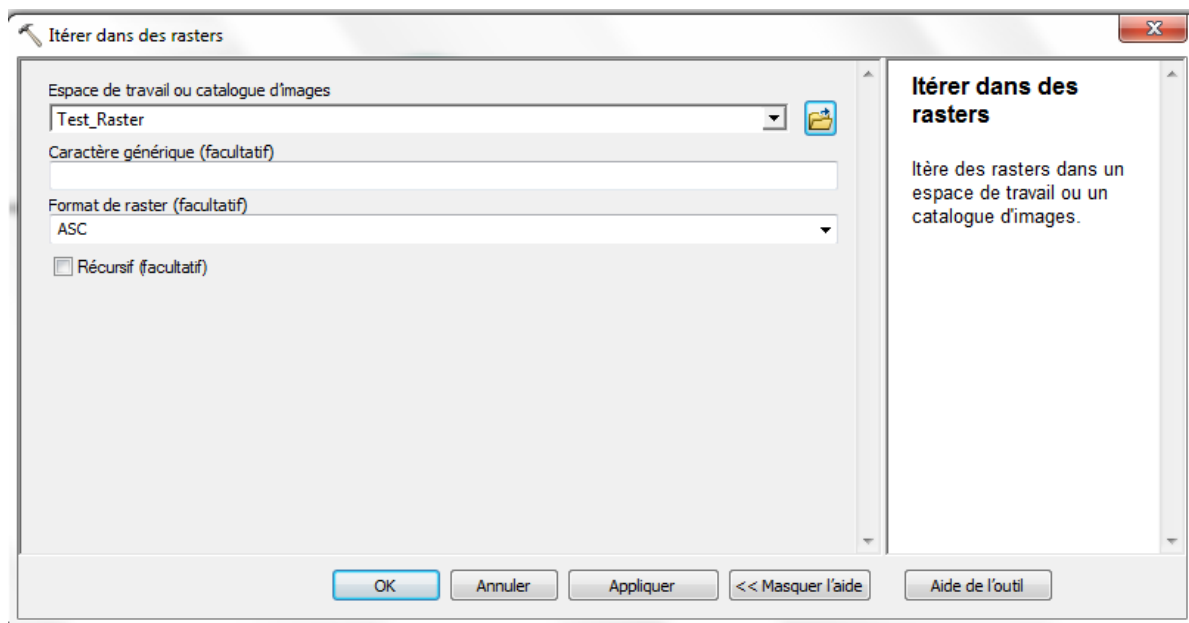


Figure 6 : La boîte de dialogue Itérer dans des rasters

Dans le même ordre d'idée, il est tout à fait possible par le biais de l'itérateur « Itérer les classes d'entités », de convertir des points issus d'un GPS (format GPX) afin de les intégrer dans une couche au format point.

L'exemple qui suit utilise un dossier contenant un ensemble de fichiers de coordonnées au format "texte". Pour chaque fichier, une couche est créée. Les coordonnées sont sélectionnées une à une et matérialisées sous la forme d'un point importé dans la couche. On obtient ainsi autant de couches cartographiant tous les points que de fichier. Chaque couche porte le nom du fichier³. L'option permettant de travailler dans les sous-dossiers existants est également présente.

```
<bounds minlat="43.99" minlon="7.03" maxlat="44.03" maxlon="7.09"/>
<trk>
  <name>track</name>
  <trkseg>
    <trkpt lat="44.025755000" lon="7.068247000">
      <ele>1439.000000</ele>
      <time>2013-05-14T03:56:49Z</time>
    </trkpt>
    <trkpt lat="44.025565000" lon="7.068369000">
      <ele>1438.000000</ele>
      <time>2013-05-14T03:57:04Z</time>
    </trkpt>
    <trkpt lat="44.025418000" lon="7.068317000">
      <ele>1438.000000</ele>
      <time>2013-05-14T03:57:21Z</time>
    </trkpt>
    <trkpt lat="44.025377000" lon="7.068270000">
      <ele>1438.000000</ele>
      <time>2013-05-14T03:57:37Z</time>
    </trkpt>
    <trkpt lat="44.025261000" lon="7.068302000">
      <ele>1438.000000</ele>
      <time>2013-05-14T03:57:52Z</time>
    </trkpt>
  </trkseg>
</trk>
```

Figure 7 : Un fichier avec des coordonnées GPX

³ L'utilisateur veillera à ne pas nommer les fichiers avec des espaces, des accents et des caractères spéciaux.




Nom	Modifié le	Type	Taille
 6 SOMMETS DEPUIS LE COL DE LA SINNE	11/12/2014 23:38	Fichier GPX	225 Ko
 AQUEDUC DE LA SIAGNE	11/12/2014 23:39	Fichier GPX	214 Ko
 AQUEDUC DU LOUP	11/12/2014 23:39	Fichier GPX	149 Ko

Figure 8 : Trois fichiers dans notre dossier

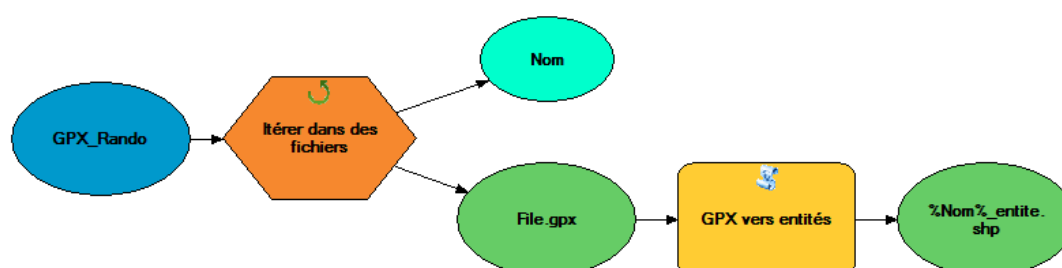


Figure 9 : Le processus

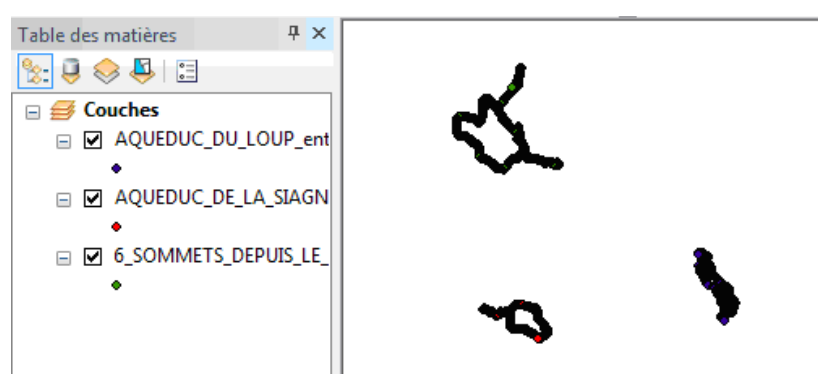


Figure 10 : Le résultat de l'intégration dans Arcgis

Conclusion

L'outil ModelBuilder permet de former des regroupements d'opérations. Ainsi le gain de temps est effectif, car une fois le modèle réalisé, il suffit de faire varier ses paramètres sans avoir à recommencer toutes les opérations.

Cette évolution corrige toutefois une lacune essentielle. Il devient maintenant possible et de manière très simple de réitérer n'importe quel ensemble d'opérations. L'outil dépasse ainsi le stade d'une simple iconographie de séquences d'instructions, pour prendre une tout autre envergure. Investir en programmation Python devient donc réservé à des problématiques différentes bien plus complexes.

Remarquons enfin que ces itérateurs sont prévus uniquement pour une utilisation dans ModelBuilder et non dans des scripts Python. En effet, leur exportation en langage Python n'est pas utilisable. Toutefois, mis à part un gain de temps, cette impossibilité n'est pas embarrassante pour un programmeur expérimenté.