

# Projet VBA ArcGIS

## Développement d'un comparateur de services de géocodage

# 1 - Rôle du plugin et contexte d'utilisation.

## 1.1 Comparer la qualité de différents services de géocodage.

L'objectif du plugin développé était la comparaison de services de géocodage, basés sur des bases de données de géolocalisation différentes : la BD ADRESSE® de l'IGN, considéré comme base de référence dans notre projet, et le service de géocodage fourni par ArcGIS Online, utilisant les données Tele Atlas pour l'Europe. Pour permettre cette comparaison, et apporter un indicateur de qualité de géocodage, il fallait que l'utilisateur du plugin ait un retour graphique des coordonnées calculées. Le projet nécessitait pour cela d'afficher, dans la fenêtre principale d'ArcMap, la localisation communiquée par chacun des services sur un fond de carte permettant un repérage le plus précis possible afin de pouvoir contrôler la justesse des deux bases de données.

## 1.2 Données fournies.

Si le service web de géolocalisation d'ArcGIS Online ne demande aucun traitement préalable à son utilisation, ce n'est pas le cas de la BD ADRESSE®. Celle-ci, publiée en 2004 dans sa première version, résulte de la fusion des deux anciennes bases d'adressage de l'IGN, POINT ADRESSE® et ROUTE ADRESSE®. La version 2, révisée en Janvier 2009, que nous utilisons dans le projet, est dans la même lignée, se présentant sous la forme de plusieurs fichiers de forme, dont les suivants ont été utilisés :

- Un shapefile ADRESSE, issu de la BD Point Adresse, de type point, associant à chaque adresse connue de l'IGN un point fixe en Lambert 93.
- Un shapefile ROUTE\_ADRESSE\_NOMEE, issu de la BD Route Adresse, de type polyligne, comportant des tronçons de voie dont on connaît les numéros de départ et de fin à gauche et à droite.
- Un shapefile ROUTE\_ADRESSE, reprenant le précédent en y ajoutant des voies n'ayant pas nécessairement de nom, ou pas de numéros.
- Des shapefiles COMMUNES et ARRONDISSEMENTS, contenant les multipolygones correspondants aux limites administratives.






























## 1.3 Traitement des données.

Dans la gamme ArcGIS (Desktop ou Serveur), un service de géolocalisation s'appuie normalement sur une unique classe d'entité, à l'exception importante des services « composites », lesquels permettent d'assembler plusieurs services uniques de géolocalisation en leur attribuant une hiérarchie. C'est la méthode utilisée par le géocodeur d'ArcGIS Online, et celle que nous avons souhaité mettre en place pour la BD Adresse. En effet, cette dernière étant composée de plusieurs fichiers de différents types, donc non concaténables, il fallait pour l'exploiter au mieux créer a minima un service de géocodage par fichier de forme.

Autre problème, la BD Adresse résultant de deux anciennes BD, les contenus ne sont pas harmonisés, et ont donc rendu obligatoire d'importants traitements. Ainsi, si certains points de la BD Point Adresse sont localisés précisément sur le lieu reflétant l'adresse réelle, la majorité est interpolée à partir de différentes sources d'informations. Par exemple, dans certains cas, Point Adresse va fournir un point correspondant au centroïde d'une voie comme adresse précise de l'un des numéros contenus dans cette voie. Pour cette même adresse, Route Adresse, basée sur la segmentation des voies et l'extrapolation des numéros entre un numéro de départ et un d'arrivée, permettra de placer l'adresse géolocalisée de manière bien plus précise. On obtiendra donc un meilleur résultat avec la bd Route Adresse qu'avec Point Adresse, ce qui ne permet pas de hiérarchiser ces deux bases de données. Suivant des conseils et scripts réalisés par [arcOpole](#), la chaîne de traitement suivante a donc été appliquée :

- Fractionnement de Point Adresse selon la source (Champ ORIGIN\_ADR) de collecte du point (Adresse projetée à la plaque, à la parcelle, extrapolée sur une voie etc.)  
En résultent les classes d'entités se terminant par « \_PA »
- Dans Route Adresse, dédoublement des entités n'ayant pas le même nom de voie à droite et à gauche, et de celles n'ayant pas la même commune à droite et à gauche, puis attribution à chacune d'une seule voie et commune. (on dédouble donc les entités contenant deux informations différentes en dotant chacune des résultantes d'une partie de l'information)
- A partir de cela, extraction des entités (dont certaines doublées donc) ayant un numéro de départ et de fin, à gauche comme à droite, différent de 0.  
Ces entités sont assemblées dans la nouvelle classe Adresse Extrapolée.
- Fusion de ROUTE\_ADRESSE et ROUTE\_ADRESSE\_NOMEE, agrégation des entités partageant même nom, même commune et s'intersectant, puis suppression des doublons.
- Fusion des fichiers Communes et Arrondissements afin de considérer les arrondissements parisiens comme des communes.
- A l'aide d'une table de correspondance trouvée sur arcOpole, ajout d'un champ Code Postal, joint selon le code INSEE, à chacune des couches résultantes.

A partir des fichiers de forme obtenus, on a ensuite créé un localisateur simple pour chacun, puis rassemblé tous ceux-ci dans un localisateur composite. Voilà la liste de l'ensemble des fichiers et localisateurs utilisés au final pour constituer notre service de géocodage à partir de la BD Adresse :

Name	Type
 BD_Adresse_CP	Locator
 BD_Adresse_INSEE	Locator
 Niv1_1_Proj_Plaque_PA_CP	Locator
 Niv1_1_Proj_Plaque_PA_INSEE	Locator
 Niv1_2_Proj_centroide_PA_CP	Locator
 Niv1_2_Proj_centroide_PA_INSEE	Locator
 Niv1_3_Troncon_route_PA_CP	Locator
 Niv1_3_Troncon_route_PA_INSEE	Locator
 Niv1_4_Adr_extrapolée_RA_CP	Locator
 Niv1_4_Adr_extrapolée_RA_INSEE	Locator
 Niv2_1_Voie_PA_CP	Locator
 Niv2_1_Voie_PA_INSEE	Locator
 Niv2_2_Voie_RA_CP	Locator
 Niv2_2_Voie_RA_INSEE	Locator
 Niv3_1_Zone_adressage_PA_CP	Locator
 Niv3_1_Zone_adressage_PA_INSEE	Locator
 Niv4_1_Commune_PA_CP	Locator
 Niv4_1_Commune_PA_INSEE	Locator
 Niv4_2_Commune_RA_CP	Locator
 Niv4_2_Commune_RA_INSEE	Locator
 Niv1_1_Proj_plaque_PA	Personal Geodatabase Feature Class
 Niv1_2_Proj_centroide_PA	Personal Geodatabase Feature Class
 Niv1_3_Troncon_route_PA	Personal Geodatabase Feature Class
 Niv1_4_Adr_extrapolée_RA	Personal Geodatabase Feature Class
 Niv2_1_Voie_PA	Personal Geodatabase Feature Class
 Niv2_2_Voie_RA	Personal Geodatabase Feature Class
 Niv3_1_Zone_adressage_PA	Personal Geodatabase Feature Class
 Niv4_1_Commune_PA	Personal Geodatabase Feature Class
 Niv4_2_Commune_RA	Personal Geodatabase Feature Class

Après transfert de ces fichiers sur le serveur intranet de l'ENSG nommé Zakaria, on a créé des services ArcGIS Server de localisation se basant sur les locateurs BD\_Adresse\_CP et BD\_Adresse\_INSEE.

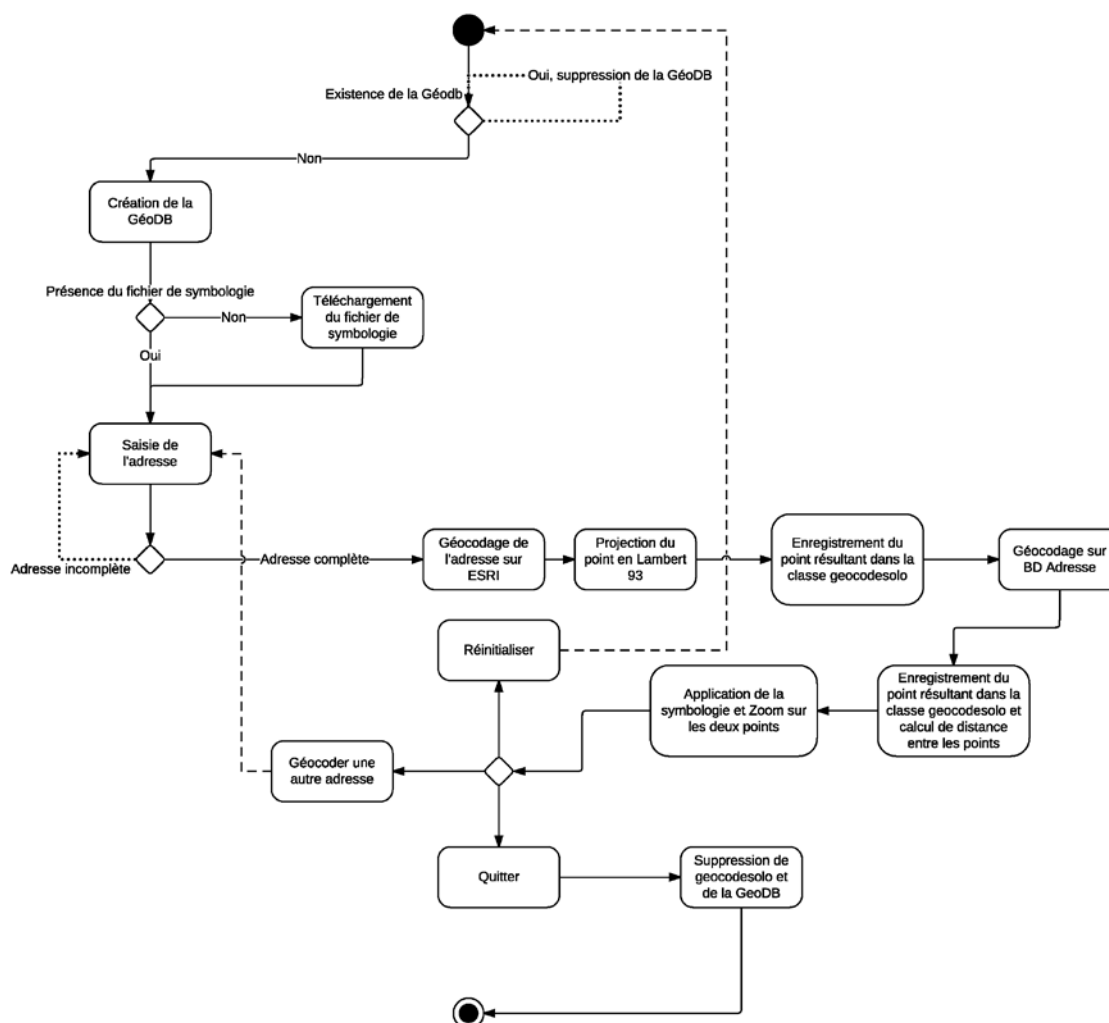
## 1.4 Contexte d'utilisation.

Les données fournies ne portant que sur l'Île-de-France, le service de géocodage ne fonctionnera que dans ces conditions, puisque si le service ArcGIS Online fournira des résultats cohérents, le service BD Adresse ne trouvera aucune correspondance. Tout le plugin a été développé de manière à être le plus autonome possible, ne requérant aucun fichier de départ, aucune condition d'utilisation particulière si ce n'est d'avoir une connexion à internet pour la récupération de la symbologie et l'utilisation de ArcGIS Online, d'être dans l'enceinte de l'ENSG pour avoir accès au serveur Zakaria, et enfin, d'exécuter le fichier mxd depuis une partition où l'utilisateur possède les droits en écriture pour la création de la géodatabase stockant les résultats du géocodage.

## 2 - Logique de fonctionnement et présentation de l'Interface Homme Machine.

### 2.1 Modèle de fonctionnement.

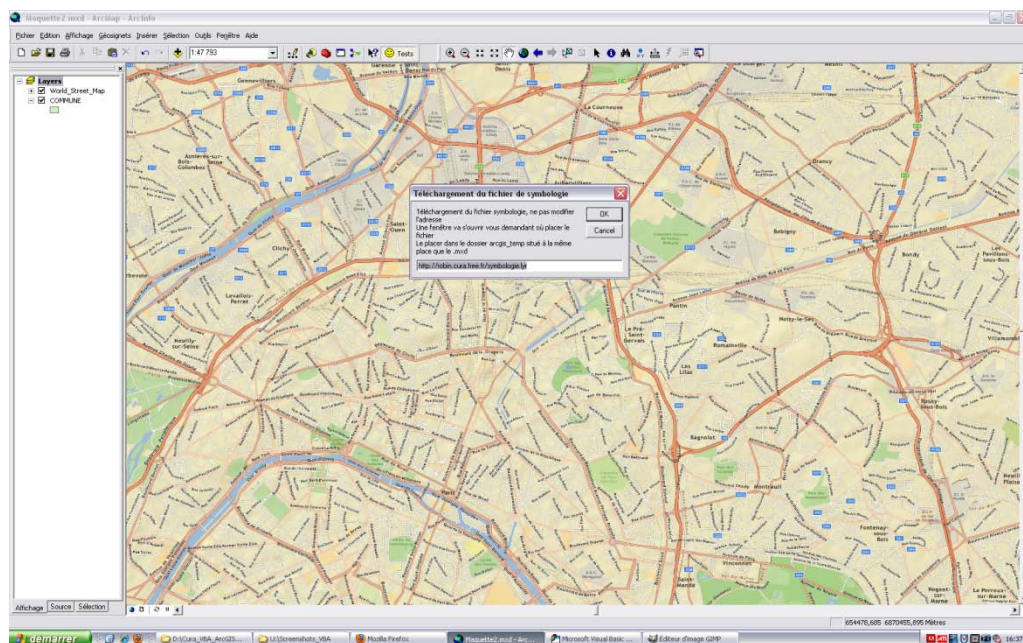
#### Diagramme UML d'activité



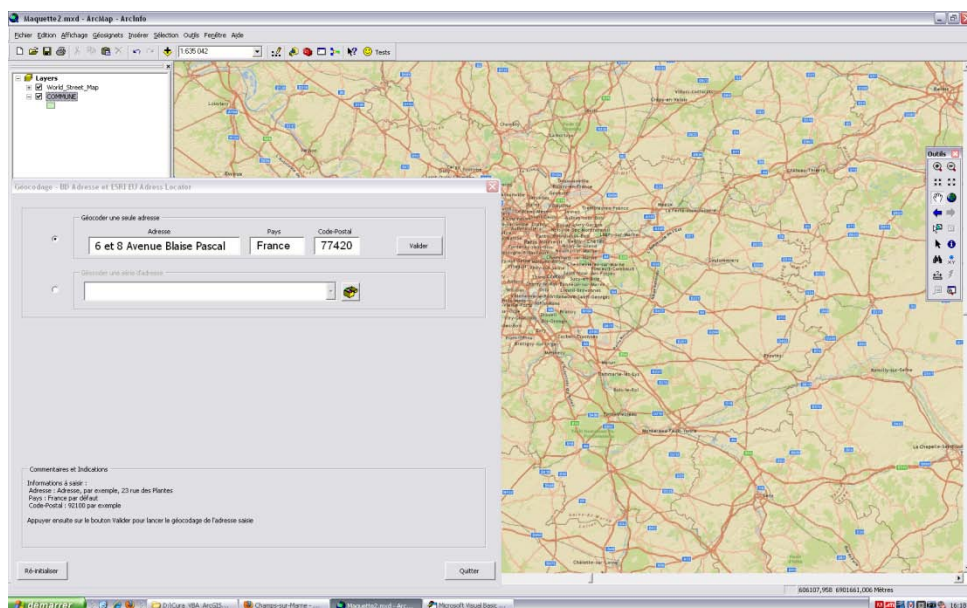
## 2.2 Description et présentation de l'interface du plugin

En premier lieu, il convient de noter que l'interface, réalisée sous Visual Basic Editor, a été créée avant le travail principal de développement, et contient donc une partie non achevée, laquelle devait permettre, avec plus de temps, d'offrir à l'utilisateur la possibilité de géocoder et afficher une table d'adresse. On y reviendra ultérieurement. Notons aussi que l'on se place ici dans l'optique d'un premier lancement du plugin, donc juste après avoir ouvert le fichier mxd sous ArcMap.

En lançant le plugin depuis le bouton « Tests » (dont la personnalisation graphique et textuelle a été oubliée dans la précipitation du rendu), il est d'abord proposé à l'utilisateur de télécharger, depuis un répertoire ftp, le fichier symbologie.lyr qui servira à l'affichage des points.

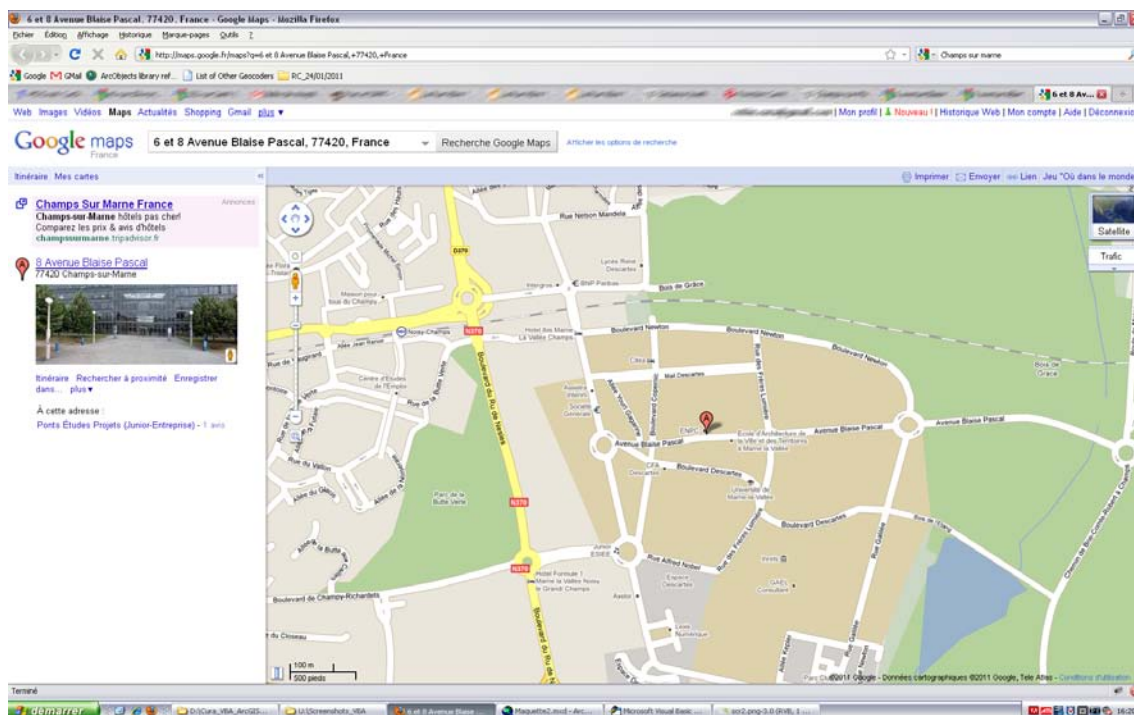


Le formulaire s'affiche ensuite, et en sélectionnant le bouton radio supérieur, l'utilisateur est invité à entrer l'adresse à géocoder.

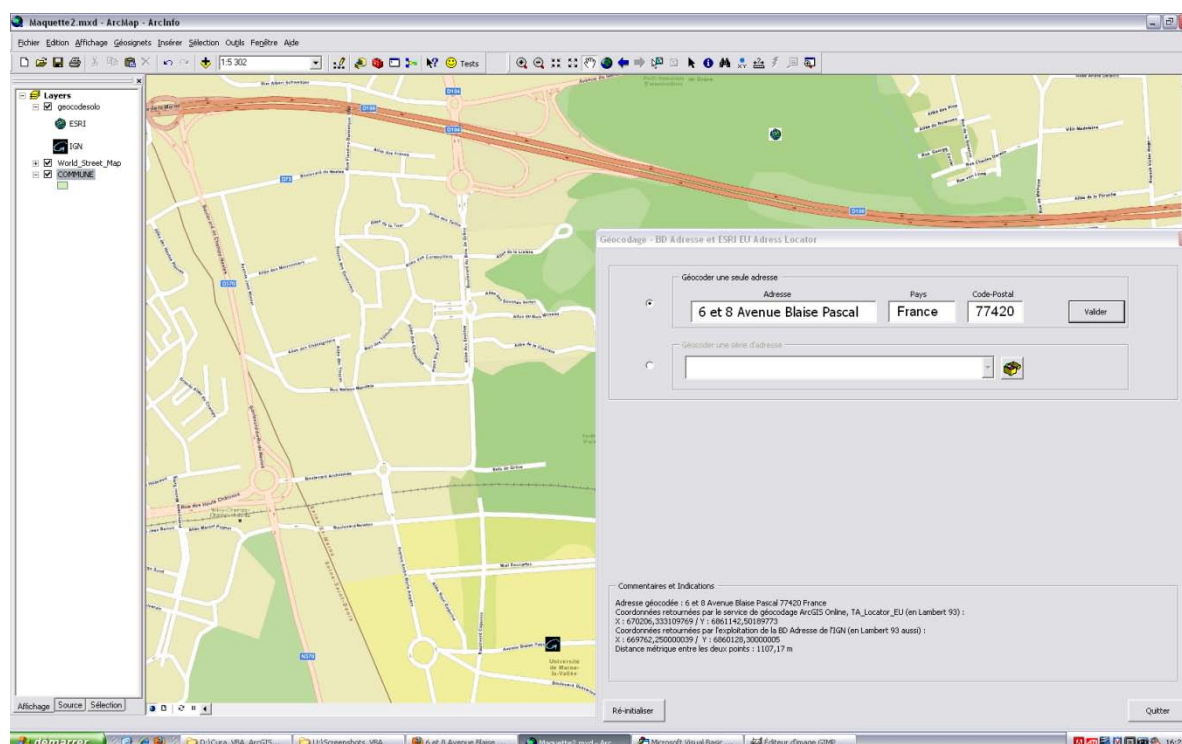




Après saisie d'une adresse et validation (bouton « Lancer »), le plugin tourne quelques secondes avant de proposer à l'utilisateur d'afficher l'adresse dans Google Maps. Si l'utilisateur fait ce choix, un nouvel onglet sera ouvert dans le navigateur par défaut, contenant la page Google Maps correspondant à l'adresse saisie.



Quelque soit le choix, dans ArcMap, une nouvelle couche geocodesolo apparaîtra dans la TOC, et la carte sera zoomée sur l'extension des points géocodés, lesquels se verront appliquer la symbologie décrite dans le fichier symbologie.lyr. La zone de commentaire affichera les informations principales sur chaque point, ainsi que la distance euclidienne les séparant.



## 3 - Présentation du code.

### 3.1 Grandes étapes du programme

Voilà la procédure principale, correspondant au clic sur le bouton Valider, commentée plus précisément que dans le fichier rendu.

```
Private Sub CmdSoloCode_Click()  
' On vérifie que les champs sont bien remplis  
If txtAdresse.Value <> "" And txtCP.Value <> "" And txtPays.Value <> "" Then  
    '##### GEOCODAGE SUR ESRI (TASKS.ARCGIS.COM) #####  
    Call geocodage_esri(txtAdresse.Value, txtCP.Value, txtPays.Value, pFCSolo)  
    '##### GEOCODAGE SUR IGN (ZAKARIA) #####  
    Call geocodage_ign(txtAdresse.Value, txtCP.Value, pFCSolo)  
  
    'On récupère la couche créée et on l'ajout dans la TOC  
    Set pLayer = New FeatureLayer  
    Set pLayer.FeatureClass = pFCSolo  
    pLayer.Name = pFCSolo.AliasName  
    'MsgBox pLayer.Name  
    pMxDoc.AddLayer pLayer  
    'On y applique la symbologie  
    Call Fonctions.applysymb  
Else  
    MsgBox "Vous devez remplir les champs Adresse, Pays et Code-Postal"  
End If  
  
'On zoom sur l'extent de la couche  
Dim pActiveView As IActiveView  
Set pActiveView = pMap  
If pMap.Layer(0) Is Nothing Then Exit Sub  
  
pActiveView.Extent = pMap.Layer(0).AreaOfInterest  
'Si l'échelle est < 1 :1000, on la fixe à 1 :1000  
If pMap.MapScale < 1000 Then  
    pMap.MapScale = 1000  
Else  
    'Sinon, on la fixe à échelle + 1:1000 pour que les points ne soient pas aux bordures de  
    la carte.  
    pMap.MapScale = pMap.MapScale + 1000  
End If  
  
pActiveView.Refresh  
pMxDoc.ActiveView.Refresh  
pMxDoc.UpdateContents  
  
'On propose l'ouverture de Google Maps  
If MsgBox("Voulez-vous afficher l'adresse recherchée, dans une page du navigateur, sur Google  
Maps ?", vbYesNo, "Recherche de l'adresse sur Google Maps") = vbYes Then  
    Call OpenURL.ouverturegmaps(txtAdresse.Value, txtCP.Value, txtPays.Value)  
End If  
  
'On met à jour le champ de commentaire avec les infos sur les points.  
Distance_points = Sqr((pPointIGN.X - pPointESRI.X) ^ 2 + (pPointIGN.Y - pPointESRI.Y) ^ 2)  
lblComments.Caption = "Adresse géocodée : " & txtAdresse.Value & " " & txtCP.Value & " " &  
txtPays.Value & vbCrLf _  
& "Coordonnées retournées par le service de géocodage ArcGIS Online, TA_Locator_EU (en Lambert  
93) : " & vbCrLf _  
& "X : " & pPointESRI.X & " / Y : " & pPointESRI.Y & vbCrLf _  
& "Coordonnées retournées par l'exploitation de la BD Adresse de l'IGN (en Lambert 93 aussi) :"  
& vbCrLf _  
& "X : " & pPointIGN.X & " / Y : " & pPointIGN.Y & vbCrLf _  
& "Distance métrique entre les deux points : " & Format(Distance_points, "0.00") & " m"  
End Sub
```

On fait donc appel à des procédures externes qui vont, dans l'ordre d'appel, créer le point via le service de géolocalisation ArcGIS Online, faire de même pour le service BD Adresse, appliquer la symbologie et enfin, ouvrir la page Google Maps.



Nous ne présenterons ici que la géolocalisation via ArcGIS Online, plus complexe que sur Zakaria. La création de la Géodatabase suit le code donné dans les manuels et les fonctions secondaires (téléchargement du fichier .lyr et ouverture de Google Maps) sont suffisamment simples et courtes pour être compréhensibles à la lecture du code.

La première étape est la connexion au serveur ArcGIS fournissant le service de géocodage. Pour cela, on passe d'abord par la création d'un objet PropertySet (iPropertySet) dans lequel on enregistre les propriétés de la connexion :

```
Dim connectionProperties As IPropertySet
Set connectionProperties = New PropertySet
connectionProperties.SetProperty "Url", "http://tasks.arcgisonline.com/arcgis/services"
```

On va ensuite créer la connexion, l'objet AGSServerConnectionFactory construisant une connexion (AGSServerConnection), à laquelle on appliquera le PropertySet défini avant :

```
Dim AGSServerConnectionFactory As IAGSServerConnectionFactory
Dim AGSServerConnection As IAGSServerConnection
Set AGSServerConnectionFactory = New AGSServerConnectionFactory
Set AGSServerConnection = AGSServerConnectionFactory.Open(connectionProperties, 0)
```

On peut maintenant se connecter au serveur et récupérer le service de localisation dans un objet Locator (lequel est l'objet de base utilisé pour le géocodage, tant en locateur fixe (fichier local) qu'un service web. Ici, on lui attribue la valeur du locateur ArcGIS Online pour l'Europe.

```
Dim locatorManager As ILocatorManager2
Set locatorManager = New locatorManager
Dim locatorWorkspace As ILocatorWorkspace
Set locatorWorkspace = locatorManager.GetAGSLocatorWorkspace(agsServerConnectionName)
Dim AGOLLocator As ILocator
Set AGOLLocator = locatorWorkspace.GetLocator("Locators/TA_Address_EU")
```

Le géocodage d'adresse à proprement parler se fait via la méthode MatchAddress de l'objet AddressGeocoding, dans laquelle on va faire correspondre les champs en entrée (champs attendus par le service de géocodage) aux champs de sortie (champs renseignés par l'utilisateur). Le reste du code est commenté (pas dans le mxd) pour une compréhension détaillée :

```
'Les adresses attendues par le locateur
Dim addressInputs As IAddressInputs
Set addressInputs = AGOLLocator
'Adresses en entrée
Dim addressFields As IFields
Set addressFields = addressInputs.addressFields

Dim fieldCount As Integer
fieldCount = addressFields.fieldCount
'Tableau contenant l'ensemble de l'adresse à géocoder
Dim addressFieldNames() As String
ReDim addressFieldNames(fieldCount)
'On remplit ce tableau
For i = 0 To fieldCount - 1
    addressFieldNames(i) = addressFields.Field(i).Name
Next
'On crée un PropertySet contenant notre tableau
Set addressProperties = New PropertySet
    addressProperties.SetProperty addressFieldNames(0), esri_Adresse
    addressProperties.SetProperty addressFieldNames(1), ""
    addressProperties.SetProperty addressFieldNames(2), esri_CP
    addressProperties.SetProperty addressFieldNames(3), esri_Pays

'On géocode notre adresse en entrée
Set pAddressGeocoding = AGOLLocator 'Notre locateur
Set pMatchProperties = pAddressGeocoding.MatchAddress(addressProperties)
```

```

'Création du système de projection Lambert 93

Dim pFactory As ISpatialReferenceFactory3
Set pFactory = New SpatialReferenceEnvironment
Dim pL93 As ISpatialReference3
Set pL93 = pFactory.CreateProjectedCoordinateSystem(esriSRProjCS_RGF1993_Lambert93)

'On crée la feature dans notre table de sortie
Dim pFeatureESRI As IFeature
Set pFeatureESRI = geodb_sortie.CreateFeature

'On parcourt maintenant les champs retournés par le géocodage
Set pMatchFields = pAddressGeocoding.MatchFields

For i_esri = 0 To pMatchFields.fieldCount - 2
    Set pMatchField = pMatchFields.Field(i_esri)

    'L'un de ces champs contient un iPoint, que l'on ajoute à notre GeoDB après projection en
    Lambert 93 (le locateur fournit du WGS 84)
    If pMatchField.Type = esriFieldTypeGeometry Then
        Set UserForm1.pPointESRI = pMatchProperties.GetProperty(pMatchField.Name)
        UserForm1.pPointESRI.Project pL93

        If Not UserForm1.pPointESRI.IsEmpty Then
            Set pFeatureESRI.Shape = UserForm1.pPointESRI
        End If
    Else
        ' On récupère maintenant les informations contenues dans les autres champs nous
        intéressant
        If pMatchField.Name = "Score" Then
            pFeatureESRI.Value(2) = pMatchProperties.GetProperty(pMatchField.Name)
        End If
        If pMatchField.Name = "Match_addr" Then
            pFeatureESRI.Value(3) = pMatchProperties.GetProperty(pMatchField.Name)
        End If
    End If
Next i_esri

'Dans un champ créé dans la couche, on ajoute l'information ESRI qui sera utilisée comme
critère pour la symbologie
pFeatureESRI.Value(1) = "ESRI"
'On crée un identifiant unique permettant l'association des points (devait être utilisé en cas
de demande de géocodage unique d'une nouvelle adresse, mais non mis en oeuvre).
pFeatureESRI.Value(5) = esri_Adresse & esri_CP

'On enregistre notre point et sa table attributaire dans geocodesolo.
pFeatureESRI.Store

```

## 3.2 Commentaire et critique du code

Le géocodage d'une adresse passe donc, comme on peut le constater, par de multiples étapes et des successions d'objets. On pourrait optimiser ce code assez fortement, déjà, en supprimant notre boucle for de parcours des champs renvoyés par le service, puisque l'ordre est toujours le même et défini dans les informations REST du service. De même, le géocodage via Zakaria reprend dans l'ensemble la même procédure, et l'on aurait pu éviter de recréer la connexion en changeant simplement les paramètres (connectionProperties). Toutefois, la connexion a un service distant prends du temps (de latence), et ce programme ne peut donc s'exécuter instantanément, d'autant que l'écriture dans une classe d'entité (en particulier la copie du IPoint) prend aussi plusieurs secondes dans notre cas. En raison de contraintes logistiques (travail à l'ENSG et à domicile, où Zakaria n'est pas accessible), il nous fallait

pourtant que les différentes briques du plugin soient le plus indépendantes possibles, afin de pouvoir selon les conditions les désactiver une à une.

De manière plus générale, le plugin n'est pas suffisamment robuste, en grande partie en raison de son développement inachevé : lors de la réalisation du modèle de développement à suivre, d'autres fonctions étaient envisagées (géocodage multiple, ajout de plusieurs adresses successives dans le géocodage simple) et le début du programme était pensé pour faciliter la mise en place de ces fonctionnalités. Le manque de temps a ainsi eu le double effet de ne pas pouvoir développer ces fonctionnalités mais aussi de fragiliser le reste du programme.

Certains malfonctionnements sont véritablement problématiques, par exemple la difficulté à lancer le programme plusieurs fois sans avoir supprimé tous les composants qu'il crée : Si l'utilisateur ne clique pas sur le bouton quitter ou réinitialiser en fin de programme, la couche geocodesolo reste active, bloquant la fermeture de la géodatabase, et empêchant ainsi sa suppression et donc une nouvelle exécution du plugin. Dans l'ensemble, je ne pense pas que l'on puisse dire que le plugin est utilisable en tant que tel par un utilisateur autre que moi, la routine d'utilisation n'étant pas claire. Pour autant, la fermeture du mxd (donc de ArcMap) après chaque géocodage enlèverait tous ces risques d'erreur, mais n'est pas très agréable pour l'utilisateur.

## 4 - Après le projet : Améliorations apportables et ressenti personnel.

### 4.1 Les améliorations simplement implémentables

La grosse difficulté du projet consiste dans le géocodage sur un service distant. Ce bloc étant fonctionnel, la mise en place des fonctionnalités basiques comme le géocodage de table aurait été assez rapide. De même pour le géocodage en série d'adresses uniques, s'ajoutant à la couche geocodesolo, avec un zoom sur la dernière adresse géocodée. Reprenons ici ces ajouts :

- **Géocodage multiple** : Prévu dans l'interface de départ, il devait permettre à l'utilisateur de sélectionner une table soit dans la TOC, soit en explorant Windows à la recherche d'un fichier dbf (ces deux fonctions sont déjà développées et fonctionnelles dans le programme), de faire correspondre les champs de la table aux champs attendus, puis de résulter en la création d'un shapefile situé au même endroit que la table. Ce serait en quelque-sortes l'équivalent de la fonction de création de points à partir d'une table XY.
- **Succession de géocodages simples** : Avec des tests de contenu de la couche geocodesolo, on aurait pu ajouter chaque adresse géocodée à cette couche, avec un zoom s'appliquant sur les deux dernières entités (via la création d'une extent à partir des XY de ces points), et des statistiques simples basées sur la comparaison des XY utilisant l'identifiant Adresse+CP (Adresse\_origine), comme la distance moyenne, la distance max et la distance min.
- **Évolution de l'aspect du formulaire** : Si le formulaire d'entrée doit être complet, on peut imaginer qu'après exécution, il prenne un aspect plus compact pour que l'utilisateur ait plus de visibilité sur l'affichage de la carte
- **Améliorations de fonctionnement** : Forcer le téléchargement du fichier .lyr dans le dossier contenant le mxd, avant de quitter le plugin, proposer l'enregistrement des couches résultantes (geocodesolo et geocodemulti) au format shape. Quitter le programme supprimerait alors tous les composants créés sauf éventuellement le fichier .lyr.
- **Suppression de l'appel au fichier .lyr** : Si l'utilisation de l'outil de geoprocessing ApplySymbologyFromLayer est très pratique car faisant appel à une fonction existante, c'est toutefois une contrainte dans le sens où cela requiert de passer par un fichier externe et le téléchargement qui va avec. En dehors de l'aspect de confort, on peut imaginer que la lecture de la symbologie depuis un fichier est plus lourde que la création et l'application d'une symbologie native dans le programme. A voir les exemples de code de création et de cartographie de symbologie du manuel ou de mes camarades, j'ai choisi de faire au plus rapide, mais au détriment certain de l'optimisation du code.

Une autre amélioration m'a intéressé dès la conception du modèle de mon programme, et permettrait de faire de ce plugin un véritable outil de comparaison de qualité de géocodage : la mise en place d'appel à des services extérieurs de géocodage, basés sur des API, tels que Google Maps, qui utilise ses propres données, ou Yahoo, basé sur les données Navteq. On aurait alors une comparaison entre la totalité des fournisseurs de données adresse sur le territoire européen. Cependant, bien que facilitée par des API web, l'appel à ces services au sein d'un projet VBA se révélait un challenge trop important dans le

temps imparti, et j'ai abandonné cette idée après avoir vu la difficulté de géocodage via un service ArcGIS Server. Si on avait utilisé ici un locator en dur (un fichier) pour les fonctions demandées, rendant possible une utilisation du geoprocessing, il y aurait sûrement eu le temps de mettre en place au moins l'appel à l'API Google Maps, assez simple. Ne pouvant me résigner à abandonner complètement cette idée, j'ai mis en place l'ouverture de page internet contenant l'adresse Google Maps, mais ce n'est qu'un pis aller de réconfort.

## 4.2 Impressions personnelles

Pour une première expérience de programmation réelle, ce projet s'est révélé extrêmement instructif et agréable. En effet, si nous avions, avec Sébastien Rey puis avec vous, réalisé des TP, c'était la première application réelle que nous avions à produire, étant autonomes quant aux choix techniques et méthodologiques à appliquer. J'ai beaucoup apprécié toute cette démarche de recherche de solutions à des problèmes assez complexes, d'autant qu'in fine, il n'y a que très peu de code redondant et simple, celui-là même qui par l'imbrication de boucles, de conditions, peut facilement embrumer mon esprit. Sans vérification, il me semble que la très grande majorité du code ici produit fait appel aux ArcObjects, et que ce projet m'a aidé à bien mieux comprendre leur fonctionnement et la logique à suivre dans leur utilisation.

L'autonomie que vous encouragez m'a été très agréable, et je pense qu'elle m'a bien préparé à un éventuel stage en programmation SIG. Je suis toutefois plus que conscient de l'avantage que d'avoir une personne ressource « à disposition », ne serait-ce que pour apporter un regard extérieur quand un blocage se fait sentir.

Ce sujet en particulier représentait un défi et une source d'apprentissage pour moi, en ce que, bien qu'ayant beaucoup pratiqué ArcMap, je n'avais encore jamais utilisé un seul des outils de géocodage, lesquels sont une composante pourtant importante des SIG dans un usage professionnel. Combiner ces fonctions avec la découverte du « monde » d'ArcGIS Server me paraît véritablement professionnalisant, dans le contexte actuel de décentralisation des usages informatiques.

J'aimerais ajouter que ce rapport comporte une importante partie sur les données et leur traitement, pour une raison simple : si cela peut sembler annexe dans le cadre de ce projet tel que défini par le sujet, les traitements préalables à l'utilisation d'un service de géocodage basé sur la BD Adresse ont occupé plus d'un tiers du temps alloué au projet. Les scripts fournis par arcOpole sont très précieux pour automatiser cette tâche, mais ils ne fonctionnaient pas sur les ordinateurs de l'ENSG ou sur mon ordinateur, par manque de puissance (dépassement de mémoire après plus de 6h de traitement pour le cas le plus long). Il m'a donc fallu un temps non négligeable pour comprendre le code python exécuté par ces scripts, et bien plus ensuite pour le mettre en place manuellement : avec un peu de rigueur, ce n'est pas très compliqué de réaliser ces opérations, mais chacune des instructions, les plus bénignes y compris, prend plusieurs minutes en raison de la quantité de données. Ainsi, l'ajout d'un champ et son calcul dans le shapefile ROUTE ADRESSE mettait plus de 3 minutes par exemple. Il ne faudrait donc pas négliger le temps pris par cette étape, et j'ai donc choisi d'y consacrer une part importante dans ce rapport.

Pour conclure, je dirais simplement que je suis ravi de ce projet, qui, en me poussant à l'autonomie et à la confrontation à des problèmes trop complexe pour moi de prime abord, m'a profondément motivé et engagé sur une voie nouvelle (pour moi) des SIG : la programmation.