



UNITE MIXTE DE RECHERCHE N° 6575
Archéologie et territoires

ECOLE THEMATIQUE
Système d'Information à Référence Spatiale et Archéologie
Tours - 8/13 septembre 2003

Atelier

Utilisation des langages macro-commande

par T. Lorho

Archéologie et territoires (UMR 6575)

UNIVERSITE DE TOURS – 3, PLACE ANATOLE FRANCE – 37000 TOURS – FRANCE

TELEPHONE 02 47 36 81 12 – TELECOPIE 02 47 36 81 04

Atelier "Macrolangages"

Le langage Avenue sur Arcview3.2

Thierry LORHO

Objectif: Le but de cet atelier est de vous familiariser avec le langage de développement Avenue sur Arcview 3.2, en vous montrant les possibilités qu'offre ce langage à travers l'écriture de scripts et la réalisation de boîtes de dialogue pour la mise en place de procédures ou d'extensions répondant à des besoins précis et directement accessibles à partir de l'interface d'Arcview.

Exercice: Pour mettre en pratique ces notions, nous allons réaliser une procédure permettant de matérialiser par des traits les relations qui unissent 2 à 2 les entités qui constituent un fichier de points.

Chaque trait créé sera caractérisé par une orientation, une longueur, les numéros ou identifiants des 2 points qui ont permis de le construire, ainsi que de deux descripteurs permettant de préciser la nature des 2 points.

Ceci peut permettre, par exemple, d'analyser une répartition de mobilier archéologique relevé en x,y,z en représentant sous forme de traits les connexions éventuelles.

Développement:

1. Création d'une boîte de dialogue:

- 1.Lancer Arcview3.2.
- 2.Ouvrir un nouveau projet.
- 3.Charger l'extension Dialog designer dans arcview.

Cette extension permet la création de boîte de dialogue.

- 4.Dans la fenêtre projet, activer l'icône boîtes de dialogues et cliquer sur nouveau.

Une boîte de dialogue vide apparaît, il va s'agir de la remplir.

Dans le menu *éditeur de boîte de dialogue*, cliquer sur *propriétés*.

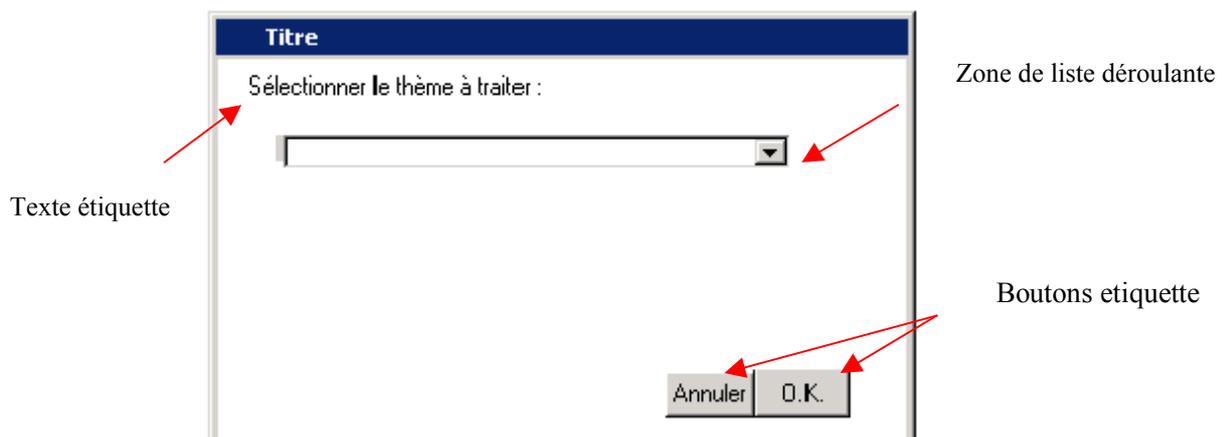
Saisissez « liste_themes» dans la rubrique nom, puis cliquer sur **OK**.

- 5.Placer dans la boîte de dialogue les contrôles :

Deux boutons étiquette (labelbouton)

Une zone de liste déroutante (combo box)

Un texte étiquette (textlabel)



Normalement, vous devez avoir une boîte de dialogue de couleur grise avec des contrôles activables mais qui sont sans effet. Ceci est logique puisqu'il n'y a aucun script rattaché à l'activation de ces contrôles.

C'est ce que nous allons nous attacher à faire maintenant.

Sauvegarder votre développement en appuyant sur l' « icône disquette », fermer la boîte de dialogue grise et la fenêtre list_themes

2. Développement des scripts de la boîte de dialogue :

Il y a trois scripts à développer correspondant aux trois contrôles « activables » de la boîte de dialogue, afin qu'une action puisse se produire en retour de l'activation (clic sur un bouton).

Le Bouton annuler :

1. Dans la fenêtre projet, cliquer sur l'icône script et nouveau
Une nouvelle fenêtre vide d'édition de script s'ouvre.

Dans le menu script, cliquer sur propriété afin de donner un nom .

Dans la rubrique nom tapez : **astage_bddlisttheme_cancel**

2. Cliquer dans la fenêtre d'édition et taper le code suivant :

```
self.getdialog.close
```

3. cliquer sur compiler.

4. sauvegarder.

5. fermer la fenêtre d'édition de script.

6. ouvrir la boîte de dialogue

7. double cliquer sur le bouton annuler pour faire apparaître la boîte de propriété

8. double cliquer dans la rubrique click et défiler la liste de scripts jusqu'à sélectionner celui que vous venez d'écrire.

9. Compiler et exécuter la boîte de dialogue.

10. Cliquer sur le bouton annuler et si tout se passe bien la boîte de dialogue doit disparaître.

Fermer la boîte d'édition de la boîte de dialogue et sauvegarder

La zone de liste :

1. Ouvrir une nouvelle fenêtre de script et donner lui le nom : **astage_bddlisttheme_listtheme.update**

2. taper le code suivant :

```
lavue=av.getactivedoc
```

```
listtheme= List.Make
```

```
listtheme={}
```

```
for each thm in lavue.GetThemes
```

```
if (thm.Is(FTheme)) then
```

```
thmFtab = thm.GetFtab
```

```
theShapeField = thmFtab.FindField("Shape")
```

```
shapetype = theShapeField.GetType
```

```
if((shapetype=#field_shapepoint)or(shapetype=#FIELD_shapemultipoint)) then
```

```
listtheme.Add(thm)
```

```
end
```

```
end
```

```
end
```

```

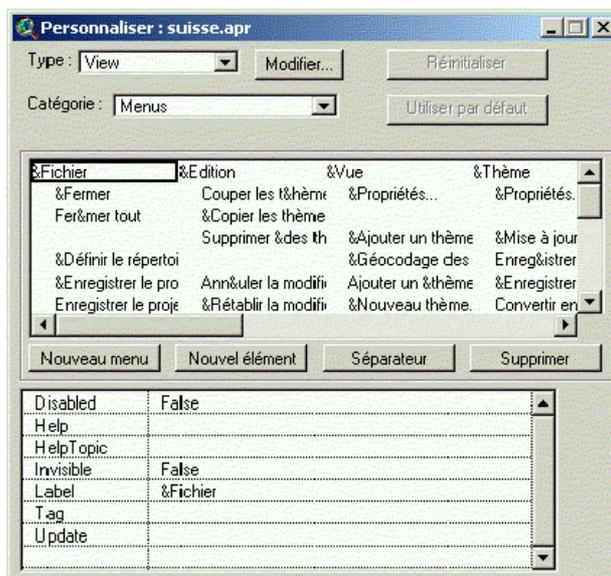
self.definefromlist(listtheme)
if(listtheme.count=0) then
    msgbox.info("Il n'y a aucun theme de géométrie ponctuelle dans cette vue,"+nl+"la
réalisation des vecteurs est impossible", "")
    exit
end

```

3. Compiler le script, fermer la fenêtre d'édition et sauvegarder.
 4. Ouvrir la boîte de dialogue et associer le nouveau script à la zone de liste dans sa rubrique update.
 5. Compiler la boîte de dialogue et lancer l'exécution. Vous pouvez constater que rien ne s'affiche dans la zone de liste.
- Ceci est du au fait que la boîte de dialogue est hors contexte, elle est conçue pour être lancée dans une vue arcview.

Mise en contexte de la boîte de dialogue list_theme :

1. Ouvrir une nouvelle fenêtre d'édition de script et lui donner le nom :
astage_bddlisttheme.open
 2. Taper le code suivant :
self.findbyname(« ltheme »).getdialog.update
 3. Compiler et fermer la fenêtre de script.
 4. Associer ce script à la boîte de dialogue dans sa rubrique open.
 5. Compiler la boîte de dialogue, sauvegarder et fermer la fenêtre d'édition de boîte de dialogue.
 6. Ouvrir une nouvelle fenêtre de script et lui donner le nom :
astage_bddlisttheme.menu.open
 7. Tapez le code suivant :
av.getproject.fnddialog("list_themes").open
 8. Compiler, sauvegarder et fermer la fenêtre d'édition de script.
 9. Cliquer sur l'icône vue dans la fenêtre de projet et cliquer sur nouvelle
 10. Ajouter à cette nouvelle vue les thèmes contenus dans le dossier astage dont le chemin d'accès est **C:\astage**.
 11. Double-cliquer dans la barre de menu de la fenêtre vue, à un endroit où il n'y a pas de bouton.
- Ceci provoque l'ouverture de cette boîte de contrôle :



12. Cliquer sur le bouton « nouveau menu ».

Un nouveau menu intitulé menu est apparu dans la fenêtre vue.

13. Cliquer dans la rubrique label et tapez **Stage**.

Le nouveau menu s'intitule maintenant stage.

14. Cliquer sur le bouton nouvel élément puis sur la rubrique label et taper **Lancer la procédure**.

15. Dans la rubrique click, associer le script **astage_bddlisttheme.menu.open**

16. Fermer la fenêtre de personnalisation du menu et sélectionner le menu « stage » en cliquant sur « Lancer la procédure ».

Normalement, la boîte de dialogue intitulée Liste des thèmes ponctuels à la vue s'affiche. Si vous cliquez sur la flèche de la zone de liste vous pouvez lire la liste de tous les thèmes ponctuels présents dans la vue.

Cependant si vous en sélectionnez un et que vous cliquez sur le bouton OK rien ne se passe.

C'était prévisible car nous n'avons rattaché aucun script à ce contrôle.

C'est l'objet de la partie suivante.

3. Développement du script associé au bouton O.K..

Ce script va permettre de tracer des traits entre toutes les entités ponctuelles du thème sélectionné à l'aide de la boîte de dialogue.

Il devra donc créer un nouveau thème de géométrie linéaire avec sa table attributaire.

Cette table possèdera les champs suivants :

1) orientation

2) longueur

Ces deux attributs, qui permettent de caractériser le segment, auront été calculés avant d'être inscrits dans la table

3) numéro ou identifiant du point de départ

4) numéro ou identifiant du point d'arrivée

5) nature du point de départ

6) nature du point d'arrivée

7) identifiant du segment = produit de la concaténation des champs 3 et 4.

Ce champ servira d'identifiant aux segments du thème.

Les valeurs des champs 3 à 6 auront été établies à partir de champs appartenant à la table du thème ponctuel. Le choix de ces champs devra pouvoir être effectué par l'utilisateur.

Remarque : il est nécessaire que la table du thème ponctuel sélectionné possède 2 champs X et Y indiquant les coordonnées de chaque point.

Si ce n'est pas le cas, il faut les créer.

Mettez la table attributaire en mode mise à jour, créez 2 champs numériques (leur nom n'est pas important) et calculez leur valeur.

Pour le champ X, tapez dans la fenêtre calcul de champ : **[Shape].GetX**

Pour le champ Y, tapez dans la fenêtre calcul de champ : **[Shape].GetY**

Fermez la mise à jour

1. Ouvrir une nouvelle fenêtre d'édition de script.

2. Nommer le script **astage_bddlisttheme_ok**

3. Tapez le code suivant :

lavue= av.getactivedoc 'Définition la vue ouverte comme le document actif
Tabd= letheme.getFtab 'Récupération la table attributaire du thème ponctuel
Tabd.seteditable(true)
listchamp= Ttabd.getfields 'Création de la liste des champs de la table attributaire

‘Choix du champ indiquant la coordonnées X des points

```
fx=msgbox.choice(listchamp,"Choisissez le champs correspondant à la coordonnées x du point","Coordonnées x")
```

```
cx=""
```

```
If (fx<>nil) then
```

```
cx=fx.getalias
```

```
depx=tabd.findfield(cx)
```

```
else
```

```
xd=field.make("X",#FIELD_DECIMAL,10,2)
```

```
tabd.addfields({xd})
```

```
depx=tabd.findfield("X")
```

```
cx="x"
```

```
For each rec in tabd
```

```
coordx=tabd.returnvalue(tabd.findfield("shape"),rec).returncenter.getx
```

```
tabd.setvalue(xd,rec,coordx)
```

```
end
```

```
end
```

‘Choix du champ indiquant la coordonnées Y des points

```
fy=msgbox.choice(listchamp,"Choisissez le champs correspondant à la coordonnées y du point","Coordonnées y")
```

```
If (fy<>nil) then
```

```
cy=fy.getalias
```

```
depy=tabd.findfield(cy)
```

```
else
```

```
Yd=field.make("Y",#FIELD_DECIMAL,10,2)
```

```
tabd.addfields({Yd})
```

```
cy=tabd.findfield("Y")
```

```
cy="Y"
```

```
For each rec in tabd
```

```
coordy=tabd.returnvalue(tabd.findfield("shape"),rec).returncenter.gety
```

```
tabd.setvalue(yd,rec,coordY)
```

```
end
```

```
end
```

‘Choix du champ servant d’index aux points

```
fnumber=msgbox.choice(listchamp,"Choisissez le champs servant d'indexation aux points","Index de points")
```

```
num=""
```

```
If (fnumber<>nil) then
```

```
num=fnumber.getalias
```

```
numdep=tabd.findfield(num)
```

```
else
```

```
index=field.make("index",#FIELD_Byte,6,0)
```

```
tabd.addfields({index})
```

```
numdep=tabd.findfield("index")
```

```
num="index"
```

```
i=0
```

```
For each rec in tabd
```

```
i=i+1
```

```

        tabd.setvalue(index,rec,i)
    end
end
'Choix du champ indiquant la nature des points
fnomstructure=msgbox.choice(listchamp,"Choisissez le champs déterminant la structure
dont est issus le point","Nom de la structure")
depo=""
If (fnomstructure<>nil) then
    depo=fnomstructure.getalias
    deporigine=Tabd.findfield(depo)

else
    struct=field.make("Structures",#Field_CHAR,100,0)
    tabd.addfields({struct})
    deporigine=tabd.findfield("structures")
    depo="Structures"
    For each rec in tabd
        Tabd.setvalue(struct,rec,"pas d'information")
    end
end

tabd.seteditable(false)

Taba=Tabd.clone
numa=Taba.findfield(num)
arrx=Taba.findfield(cx)
arry=Taba.findfield(cy)
arrorigine=Taba.findfield(depo)
'Ouverture d'une table intermédiaire qui servira à la création du thème linéaire
nomvtabc=Filedialog.put("table_de_correspondance".asfilename,"*.txt","Attribuer un
nom à la table de correspondance")
vtabc=vtab.makenew(nomvtabc,dbase)
depor=field.make("Structure départ",#Field_CHAR,100,0)
error=field.make("Structure arrivée",#Field_CHAR,100,0)
numptdepart=field.make("Pt de départ",#FIELD_LONG,4,0)
numptarrive=field.make("Pt d'arrivée",#FIELD_LONG,4,0)
xd=field.make("Xd",#FIELD_DECIMAL,10,2)
yd=field.make("Yd",#FIELD_DECIMAL,10,2)
xa=field.make("Xa",#FIELD_DECIMAL,10,2)
ya=field.make("Ya",#FIELD_DECIMAL,10,2)

vtabc.addfields({numptdepart,numptarrive,xd,yd,xa,ya,depor,error})

For each rec in Tabd
    vnumd=tabd.returnvalue(numdep,rec)
    vdex=tabd.returnvalue(depx,rec)
    vdpy=tabd.returnvalue(tabd.findfield(cy.asstring),rec)
    vdpor=Tabd.returnvalue(deporigine,rec)
    For each recl in Taba
        vnuma=Taba.returnvalue(numa,recl)

```

```

varrx=Taba.returnvalue(arrx,rec1)
varry=Taba.returnvalue(arry,rec1)
varror=Taba.returnvalue(arrorigine,rec1)
ww=vtabc.addrecord
vtabc.setvalue(numptdepart,ww,vnumd)
vtabc.setvalue(xd,ww,vdexp)
vtabc.setvalue(yd,ww,vdepy)
vtabc.setvalue(numptarrive,ww,vnuma)
vtabc.setvalue(xa,ww,varrx)
vtabc.setvalue(ya,ww,varry)
vtabc.setvalue(depor,ww,vdepor)
vtabc.setvalue(armor,ww,varror)
end

```

end

‘Création de la table attributaire du thème linéaire

```

nomftabligne=Filedialog.put(("Vecteur
de"++letheme.getname).asfilename,"*.shp","Attribuer un nom au thème ligne à créer")

```

```

Ftabligne=Ftab.makenew(nomftabligne,polyline)
ldepor=field.make("Structure départ",#Field_CHAR,100,0)
larmor=field.make("Structure arrivée",#Field_CHAR,100,0)
longueur=field.make("longueur",#FIELD_DECIMAL,10,2)
orient=field.make("orientation",#FIELD_DECIMAL,10,2)
numdepa=field.make("N°départ",#FIELD_LONG,10,0)
numarri=field.make("N°arrivée",#FIELD_LONG,10,0)
nconcat=field.make("Vecteur",#Field_CHAR,10,0)

```

```

Ftabligne.addfields({orient,longueur,numdepa,numarri,nconcat,ldepor,larmor})

```

```

shapeline=ftabligne.findfield("shape")

```

```

shapeline.setvisible(true)

```

```

Ftabligne.StartEditingWithRecovery

```

```

Ftabligne.BeginTransaction

```

‘Calcul de l’orientation et de la longueur des segments

```

For each rec in vtabc

```

```

l=list.make

```

```

nd=vtabc.returnvalue(numptdepart,rec)

```

```

na=vtabc.returnvalue(numptarrive,rec)

```

```

xdep=vtabc.returnvalue(xd,rec)

```

```

ydep=vtabc.returnvalue(yd,rec)

```

```

xarr=vtabc.returnvalue(xa,rec)

```

```

yarr=vtabc.returnvalue(ya,rec)

```

```

ordep=vtabc.returnvalue(depor,rec)

```

```

orarr=vtabc.returnvalue(armor,rec)

```

```

ltrait=(((xdep-xarr)*(xdep-xarr)) + ((ydep-yarr)*(ydep-yarr))).sqrt

```

```

nconca=(nd.asstring+"-"+na.asstring)

```

```

dx=(xdep-xarr)

```

```

dy=(ydep-yarr)

```

```

t=(dx/dy).atan*(180/3.14159)

```

```

signe=(dx/((dx).abs))*(dy/((dy).abs))
  If(dx=0) then
    angle=0
  elseif(dy=0) then
    angle=90
  else
    angle=(90-(signe*90)) + t
  end
pd=point.make(xdep,ydep)
pa=point.make(xarr,yarr)
l.add(pd)
l.add(pa)
ligne=polyline.make({l})
zz=Ftabligne.addrecord
  Ftabligne.setvalue(shapeligne,zz,ligne)
  Ftabligne.setvalue(longueur,zz,ltrait)
  Ftabligne.setvalue(numarri,zz,na)
  Ftabligne.setvalue(numdepa,zz,nd)
  Ftabligne.setvalue(nconcat,zz,nconca)
  Ftabligne.setvalue(orient,zz,angle)
  Ftabligne.setvalue(ldepor,zz,ordep)
  Ftabligne.setvalue(larror,zz,orarr)

end

```

```

Ftabligne.EndTransaction
  ‘Création du thème linéaire
Ffthemeligne=Ftheme.make(Ftabligne)
Ftabligne.StopEditingWithRecovery(true)
  ‘Ajout du nouveau thème à la vue
lavue.addtheme(Ffthemeligne)

```

4. Compiler, corriger les erreurs éventuelles, sauvegarder et fermer la boîte d’édition du script
5. Ouvrir l’éditeur de la boîte de dialogue et associer ce script au bouton OK, par sa rubrique click
6. Compiler la boîte de dialogue, sauvegarder et fermer l’éditeur
7. Afficher la vue avec les différents thèmes puis lancer la procédure par le menu.

Si tout se passe bien un thème linéaire doit apparaître dans la fenêtre de vue avec les traits reliant les points deux à deux.
Rendez le visible et regardez sa table attributaire