Le format de fichier .geo

Documentation par Nicolas Pourcelot © 2006

Dernière révision le 15 novembre 2006.

Table des matières

Avant-propos

Le format de fichier de fichier utilisé par geolib respecte les spécifications XML 1.0.

Cependant, je ferais ici deux remarques.

La première, c'est qu'une implémentation complète des spécifications XML aurait été assez pénible, et passablement superflue. Par conséquent, l'implémentation de XML utilisée par l'API de geolib est très partielle, et les restrictions imposées sont significativement plus importantes que pour un document XML « classique ».

La deuxième, c'est que la structure n'est pas entièrement traduite en XML.
WxGéométrie possède un interpréteur python « personnalisé » intégré, qui est accessible en ligne de commande. Cet interpréteur ne reconnaît pas les instructions potentiellement dangereuses, et embarque un espace de noms et des librairies propres à WxGéométrie.
Traduire en XML une suite d'instructions Python serait à mon avis contre-productif, au moins dans l'état actuel des choses[1].

1 Somme toute, il est peut probable que le format de fichier .geo devienne un jour un standard ;-)
Le but est essentiellement d'avoir un format de fichier lisible par un humain, et traitable au besoin par ordinateur – encore que cette dernière éventualité ne soit pas si probable !
Concernant le débat XML/ Python, on peut retenir qu'une utilisation plus systématique d'XML augmenterait le temps de traitement

Dernière remarque concernant XML, je suis loin de bien maîtriser le sujet. D'ailleurs, c'est aussi une des raisons de l'existence de ce document – sinon, j'utiliserais peut-être un « XML Schema », ou plus modestement un « Document Type Definition ».

I. Spécifications de forme

Le vocabulaire utilisé ici est autant que possible celui couramment employé pour la description de fichiers XML.

1. Caractères autorisés

L'encodage du document doit se faire en UTF-8.

Le caractère & est interdit, sauf au début d'un appel de caractères.
Il doit être remplacé par l'appel de caractère « *&* ».

Les caractères < et > sont interdits, sauf en début et en fin de balise.
Ils doivent être respectivement remplacés par les appels de caractères « *<* » et « *>* ».

Les caractères " et ' sont interdits dans les valeurs d'attributs.
Ils peuvent être utilisés ailleurs dans le document.

On pourra utiliser librement les retours à la ligne (Unix, Windows ou Mac, au choix) entre les balises.
De manière générale, les retours à la ligne peuvent s'effectuer indifféremment selon une de ces trois conventions[2].

2. Structure externe du document

Le document commencera par la ligne suivante :

```
<?xml version='1.0' encoding='utf-8'?>
```

Le reste du document sera contenu entre les balises <*Document*> et </*Document*>.

Un document vide ressemblera donc à ceci :

```
<?xml version='1.0' encoding='utf-8'?>
<Document type='WxGeometrie' version='0.106.9' module='geometre'>
</Document>
```

La balise <*Document*> doit comporter les attributs suivants :

- un argument de nom « type », et dont la valeur doit toujours être « WxGeometrie »[3].
- un argument de nom « version », et dont la valeur doit être le numéro de la version de WxGéométrie ayant servi à produire le document.
  Ce numéro de version doit être une succession de nombres entiers positifs ou nuls séparés par des points[4].

---

du fichier et le temps de développement et la taille du code source de WxGéométrie, . Enfin elle fait perdre en puissance et en flexibilité. Par contre, elle rendrait le fichier plus homogène, et plus facilement lisible par un programme externe, du fait de bibliothèques de traitement déjà existantes. Enfin, elle rendrait plus improbables d'éventuelles failles de sécurité toujours possibles.

2 Python supporte en effet nativement les trois. L'encodage se fera donc de préférence selon la plateforme de travail de l'utilisateur.

3 Ceci pourrait varier à l'avenir. En particulier, WxGeometrie pourrait devenir WxGéométrie. Le logiciel pourrait aussi changer de nom...

4 La convention respectée par WxGéométrie est que deux numéros de versions soient classés en comparant les premiers nombres. Si

- Un argument de nom « module », et dont la valeur doit être le nom du module ayant servi à créer le fichier[5].

Elle peut comporter d'autres arguments.

3. Structuration du contenu

a) Règles de structuration

Entre les balises <*Document*> et </*Document*>, et entre chaque couple de type « <*balise*> » et « </*balise*> », on aura une ou plusieurs balises ouvrantes et fermantes.

Les restrictions par rapport aux spécifications XML 1.0 sont :

- les balises d'éléments vides[6] sont interdites,
- les commentaires[7] sont interdits,
- les sections de type CDATA sont interdites,
- le contenu ne peut pas être mixte (par exemple, un élément ne peut pas contenir à la fois du texte et d'autres éléments),
- une balise (autre que le <*Document*> initial) **ne peut pas avoir d'attributs**.

Autrement dit, **entre chaque couple « balise ouvrante »-« balise fermante », on peut avoir *ou bien* du texte, *ou bien* un ou plusieurs couples « balise ouvrante »-« balise fermante », mais pas autre chose, et pas les deux à la fois**.

Les **noms de balise** doivent commencer par un caractère alphabétique majuscule ou minuscule : [A-Z] | [a-z][8].

Ils ne doivent comporter que des caractères alphanumériques, ou des tirets-bas : [A-Z] | [a-z] | [0-9] | "_"

Par convention, les balises de niveau 1 (celles qui prennent directement appui sur <*Document*>) commencent par une majuscule, et les autres par une minuscule, pour plus de lisibilité.

b) Exemples

Une exemple simple

```
<?xml version='1.0' encoding='utf-8'?>
<Document type='WxGeometrie' version='0.106.9' module='geometre'>
<Hello_>
</Figure>
</Document>
```

```
<?xml version='1.0' encoding='utf-8'?>
<Document type='WxGeometrie' version='0.106.9' module='geometre'>
<Figure>
</Figure>
```

cela ne suffit pas, on compare les deuxièmes nombres, et ainsi de suite.
Par convention, une absence de nombre est traité comme un « -1 ».
Par exemple, on a 0.105.99 < 0.106 < 0.106.0 < 0.106.1 < 0.106.1.3 .

5 Cf. « documentation de l'API.odt ».
Ce nom doit être celui du sous-dossier du dossier *modules/* qui contient le module.

6 Les balises à la fois ouvrantes et fermantes. Exemple : <*br /*> en XHTML.

7 Exemple : <*!--Mon commentaire--*>

8 Globalement, un nom de balise valide est un nom de variable valide pour Python, à une restriction près : l'interdiction des tirets-bas en début de balise. (Ceci car les noms d'attributs commençant par des tirets-bas sont traités différemment en Python).

```
<Figure>
<Proprietes>
Diverses propriétés peuvent prendre place ici.
</Proprietes>
<Contenu>
</Contenu>
</Figure>
<Autres_infos>
Blablablabla...
</Autres_infos>
</Document>
```

c) Contre-exemples commentés

Ces contre-exemples illustrent les restrictions supplémentaires par rapport aux spécifications XML.

```
<?xml version='1.0' encoding='utf-8'?>
<Document type='WxGeometrie' version='0.106.9' module='geometre'>
<Balise-perso>
</Balise-perso>
</Document>
```

Le nom de balise est incorrect (caractère « - » non supporté).

```
<?xml version='1.0' encoding='utf-8'?>
<Document type='WxGeometrie' version='0.106.9' module='geometre'>
<Ma_balise>
Salut !
<Autres>
</Autres>
</Ma_balise>
</Document>
```

Le contenu entre <*Ma_balise*> et <*/Ma_balise*> est de type mixte, ce qui est interdit : il contient à la fois du texte – « *Salut !* », et d'autres balises « <*Autres*> <*/Autres*> ».

```
<?xml version='1.0' encoding='utf-8'?>
<Document type='WxGeometrie' version='0.106.9' module='geometre'>
<Figure export="png">
</Figure>
<!--Version provisoire-->
</Document>
```

Les attributs sont interdits pour les balises autre que <*Document*>.
Les commentaires sont interdits.

```
<?xml version='1.0' encoding='utf-8'?>
<Document type='WxGeometrie' version='0.106.9' module='geometre'>
<Figure />
</Document>
```

Les balises vides sont interdites.

II. Balises courantes

1. Avertissement

Le but n'est pas ici de donner une liste exhaustive de toutes les balises possibles, d'autant que leur liste varie fréquemment au gré des ajouts de fonctionnalités à WxGéométrie.

On ne donnera ici que les balises utilisées par les modules courant, et de niveau 1[9], afin d'éviter d'éventuels conflits de noms.

2. Liste des balises de niveau 1

Balise	Module	Contenu	Commentaires
<Affichage>	*API graphique*	Paramètres d'affichage de la feuille.	Utilisé par tous les modules graphiques.
<Courbe>	Traceur	Équation et paramètres d'affichage d'une courbe.	Occurrences multiples. Seront regroupées dans <Courbes> ?
<Figure>	*API graphique*	Code python permettant de générer une figure.	Utilisé par tous les modules graphiques.
<Diagramme>	Statistiques	Données propres à un diagramme.	
<Macro>	*API graphique*	Bloc de code python.	Utilisé par certains modules graphiques. Va très probablement évoluer...
<Calculatrice>	Calculatrice	Sauvegarde de l'état de la calculatrice.	Réservée d'avance...
<Courbes>	Traceur		Réservée d'avance...
<Meta>	API graphique	Information diverses sur le document de travail. (Titre, auteur, date, description...)	Réservée d'avance...
<Macros>	*API graphique*		Réservée d'avance...

3. Exemples de fichier

Les balises prenant appui sur la racine du document sont en bleu et en gras, pour plus de lisibilité.
Les autres sont en bleu, sans gras.
Les balises de début et fin de document sont en gras et italique.
Le texte est en noir, sauf le code Python, en vert.

a) Fichier produit par le module Geometre.

```
<?xml version='1.0' encoding='utf-8'?>
<Document type='WxGeometrie' version='0.106.9' module='geometre'>
<Affichage>
<origine_axes>
(0, 0)
</origine_axes>
<taille>
{u'(': 5, u'+': 4, u'o': 3, u'&gt;': 10}
</taille>
<style_point>
```

9 C'est-à-dire qu'on trouve directement à la racine du document. Ou, dit encore autrement, qui dépendent de l'élément <Document> lui-même.

```
u'+'
</style_point>
<resolution>
1000
</resolution>
<affiche_fleches>
True
</affiche_fleches>
<liste_axes>
(0, 1)
</liste_axes>
<affiche_quadrillage>
True
</affiche_quadrillage>
<quadrillages>
(((None, None), u':', 0.5, u'k'),)
</quadrillages>
<orthonorme>
False
</orthonorme>
<utiliser_repere>
True
</utiliser_repere>
<gradu>
(1, 1)
</gradu>
<repere>
(u'O', u'I', u'J')
</repere>
<affiche_axes>
True
</affiche_axes>
</Affichage>
<Figure>
set_fenetre(-8, 8, -5, 5)
M7=Point(Variable(6.01634877384),Variable(-3.57971014493), **{'style':
'+', 'categorie': 'points', 'couleur': 'r', 'taille': 8, 'epaisseur': 1.0,
'label': '', 'visible': True, 'legende': 2, 'niveau': 10, 'fixe': False})
M6=Point(Variable(1.76566757493),Variable(-1.63768115942), **{'style':
'+', 'categorie': 'points', 'couleur': 'r', 'taille': 8, 'epaisseur': 1.0,
'label': '', 'visible': True, 'legende': 2, 'niveau': 10, 'fixe': False})
M5=Point(Variable(0.828337874659),Variable(1.66666666667), **{'style':
'+', 'categorie': 'points', 'couleur': 'r', 'taille': 8, 'epaisseur': 1.0,
'label': '', 'visible': True, 'legende': 2, 'niveau': 10, 'fixe': False})
M4=Point(Variable(2.41961852861),Variable(3.75362318841), **{'style': '+',
'categorie': 'points', 'couleur': 'r', 'taille': 8, 'epaisseur': 1.0,
'label': '', 'visible': True, 'legende': 2, 'niveau': 10, 'fixe': False})
M3=Point(Variable(-2.20163487738),Variable(-4.15942028986), **{'style':
'+', 'categorie': 'points', 'couleur': 'r', 'taille': 8, 'epaisseur': 1.0,
'label': '', 'visible': True, 'legende': 2, 'niveau': 10, 'fixe': False})
M2=Point(Variable(-5.55858310627),Variable(-3.92753623188), **{'style':
'+', 'categorie': 'points', 'couleur': 'r', 'taille': 8, 'epaisseur': 1.0,
'label': '', 'visible': True, 'legende': 2, 'niveau': 10, 'fixe': False})
M8=Point(Variable(6.43051771117),Variable(1.23188405797), **{'style': '+',
'categorie': 'points', 'couleur': 'r', 'taille': 8, 'epaisseur': 1.0,
'label': '', 'visible': True, 'legende': 2, 'niveau': 10, 'fixe': False})
M1=Point(Variable(-1.89645776567),Variable(2.94202898551), **{'style':
```

```
'+', 'categorie': 'points', 'couleur': 'r', 'taille': 8, 'epaisseur': 1.0,
'label': '', 'visible': True, 'legende': 2, 'niveau': 10, 'fixe': False})
p2=Polygone([M4,M5,M6,M7,M8], **{'style': '-', 'categorie': 'polygones',
'couleur': 'y', 'epaisseur': 1.0, 'label': '', 'visible': True, 'legende':
0, 'niveau': 0, 'alpha': 0.20000000000000001})
p1=Triangle(M1,M2,M3, **{'style': '-', 'categorie': 'polygones',
'couleur': 'y', 'epaisseur': 1.0, 'label': '', 'visible': True, 'legende':
0, 'niveau': 0, 'alpha': 0.20000000000000001})
p2.cotes[0].style(**{'style': '-', 'categorie': 'polygones', 'couleur':
'y', 'epaisseur': 1.0, 'label': '', 'visible': True, 'legende': 0,
'niveau': 0, 'alpha': 0.20000000000000001})
p2.cotes[1].style(**{'style': '-', 'categorie': 'polygones', 'couleur':
'y', 'epaisseur': 1.0, 'label': '', 'visible': True, 'legende': 0,
'niveau': 0, 'alpha': 0.20000000000000001})
p2.cotes[2].style(**{'style': '-', 'categorie': 'polygones', 'couleur':
(0.0, 0.50196078431372548, 0.0), 'epaisseur': 1.0, 'label': '', 'visible':
True, 'legende': 0, 'niveau': 0, 'alpha': 0.20000000000000001})
p2.cotes[3].style(**{'style': '-', 'categorie': 'polygones', 'couleur':
'y', 'epaisseur': 1.0, 'label': '', 'visible': True, 'legende': 0,
'niveau': 0, 'alpha': 0.20000000000000001})
p2.cotes[4].style(**{'style': '-', 'categorie': 'polygones', 'couleur':
'y', 'epaisseur': 1.0, 'label': '', 'visible': True, 'legende': 0,
'niveau': 0, 'alpha': 0.20000000000000001})
p1.cotes[0].style(**{'style': '-', 'categorie': 'polygones', 'couleur':
'y', 'epaisseur': 1.0, 'label': '', 'visible': True, 'legende': 0,
'niveau': 0, 'alpha': 0.20000000000000001})
p1.cotes[1].style(**{'style': '-', 'categorie': 'polygones', 'couleur':
'y', 'epaisseur': 1.0, 'label': '', 'visible': True, 'legende': 0,
'niveau': 0, 'alpha': 0.20000000000000001})
p1.cotes[2].style(**{'style': '-', 'categorie': 'polygones', 'couleur':
'y', 'epaisseur': 1.0, 'label': '', 'visible': True, 'legende': 0,
'niveau': 0, 'alpha': 0.20000000000000001})
</Figure>
</Document>
```

b) Fichier produit par le module Traceur

```
<?xml version='1.0' encoding='utf-8'?>
<Document type='WxGeometrie' version='0.106' module='traceur'>
<Courbe>
<Y>
 -{3}x -{318}+{120}ln(x +{10})
</Y>
<intervalle>
[15;40]
</intervalle>
<active>
False
</active>
</Courbe>
<Courbe>
<Y>
25
</Y>
<intervalle>
```

```
</intervalle>
<active>
False
</active>
</Courbe>
<Courbe>
<Y>
{0.026}x +{0.194}
</Y>
<intervalle>

</intervalle>
<active>
True
</active>
</Courbe>
<Courbe>
<Y>

</Y>
<intervalle>

</intervalle>
<active>
True
</active>
</Courbe>
<Courbe>
<Y>

</Y>
<intervalle>

</intervalle>
<active>
True
</active>
</Courbe>
<Courbe>
<Y>

</Y>
<intervalle>

</intervalle>
<active>
True
</active>
</Courbe>
<Courbe>
<Y>

</Y>
<intervalle>

</intervalle>
<active>
```

```
True
</active>
</Courbe>
<Courbe>
<Y>

</Y>
<intervalle>

</intervalle>
<active>
True
</active>
</Courbe>
<Courbe>
<Y>

</Y>
<intervalle>

</intervalle>
<active>
True
</active>
</Courbe>
<Affichage>
<origine_axes>
(0, 0)
</origine_axes>
<taille>
{u'(': 5, u'+': 4, u'o': 3, u'&gt;': 10}
</taille>
<style_point>
u'+'
</style_point>
<resolution>
1000
</resolution>
<affiche_fleches>
True
</affiche_fleches>
<liste_axes>
(0, 1)
</liste_axes>
<affiche_quadrillage>
True
</affiche_quadrillage>
<quadrillages>
(((None, None), u':', 0.5, u'k'),)
</quadrillages>
<orthonorme>
False
</orthonorme>
<utiliser_repere>
False
</utiliser_repere>
<gradu>
```

```
(5, 0.25)
</gradu>
<repere>
(u'O', u'I', u'J')
</repere>
<affiche_axes>
True
</affiche_axes>
</Affichage>
<Figure>
set_fenetre(-4.4027688208864504, 41.35652998191523, -0.1667630740360061,
1.6113870626714206)
M7=Point(Variable(28.0),Variable(0.93), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
False})
_E=Point(Variable(30.0),Variable(0.974), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'legende': 1, 'visible': True,
'fixe': False})
_D=Point(Variable(25.0),Variable(0.844), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'legende': 1, 'visible': True,
'fixe': False})
_A=Point(Variable(18.0),Variable(0.662), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'legende': 1, 'visible': True,
'fixe': False})
M10=Point(Variable(36.0),Variable(1.1), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
False})
t=Texte('aire en m2',Variable(-2.19384913265),Variable(1.19779420293),
**{'style': 'normal', 'angle': 0, 'extra': {'rotation': 90}, 'couleur':
'k', 'taille': 12.0, 'epaisseur': 4, 'label': 'Aire en m2', 'visible':
True, 'famille': 'sans-serif'})
M2=Point(Variable(8.0),Variable(0.4), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
False})
M4=Point(Variable(16.0),Variable(0.64), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
False})
M3=Point(Variable(12.0),Variable(0.5), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
False})
M6=Point(Variable(24.0),Variable(0.84), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
False})
M8=Point(Variable(30.0),Variable(0.98), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
False})
M1=Point(Variable(4.0),Variable(0.25), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
False})
G=Point(Variable(21.0),Variable(0.74), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'legende': 1, 'visible': True,
'fixe': False})
M9=Point(Variable(32.0),Variable(1.02), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
False})
M5=Point(Variable(20.0),Variable(0.74), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'visible': True, 'fixe':
```

```
False})
s=Texte('Poids en kg',Variable(35.7987204107),Variable(-0.0763486603051),
**{'style': 'normal', 'angle': 0, 'couleur': 'k', 'taille': 12.0,
'epaisseur': 4, 'label': 'Poids en kg', 'visible': True, 'famille': 'sans-
serif'})
C1=Point(Variable(0.0),Variable('_D.y'), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '0.84', 'visible': True, 'fixe':
False})
C2=Point(Variable(0.0),Variable('_E.y'), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '0.97', 'visible': True, 'fixe':
False})
B2=Point(Variable('_E.x'),Variable(0.0), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '', 'legende': 1, 'visible': True,
'fixe': False})
B1=Point(Variable('_D.x'),Variable(0.0), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '30', 'visible': True, 'fixe':
False})
C=Point(Variable(0.0),Variable('_A.y'), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '0.66', 'visible': True, 'fixe':
False})
B=Point(Variable('_A.x'),Variable(0.0), **{'style': '+', 'couleur': 'b',
'taille': 8, 'epaisseur': 1.0, 'label': '18', 'visible': True, 'fixe':
False})
BA=Segment(B,_A, **{'visible': True, 'style': '--', 'couleur': 'g',
'epaisseur': 1.0, 'label': None})
v=Segment(C1,_D, **{'visible': True, 'style': '--', 'couleur': 'g',
'epaisseur': 1.0, 'label': None})
CA=Segment(C,_A, **{'visible': True, 'style': '--', 'couleur': 'g',
'epaisseur': 1.0, 'label': None})
h=Segment(B1,B2, **{'visible': True, 'style': '-', 'couleur': 'g',
'epaisseur': 2, 'label': None})
n=Segment(C1,C2, **{'visible': True, 'style': '-', 'couleur': 'g',
'epaisseur': 2, 'label': None})
l=Segment(C2,_E, **{'visible': True, 'style': '--', 'couleur': 'g',
'epaisseur': 1.0, 'label': None})
m=Segment(B2,_E, **{'visible': True, 'style': '--', 'couleur': 'g',
'epaisseur': 1.0, 'label': None})
u=Segment(B1,_D, **{'visible': True, 'style': '--', 'couleur': 'g',
'epaisseur': 1.0, 'label': None})
</Figure>
</Document>
```

III. « Document Type Definition »

1. Notes

Version 0.1 du « Document Type Definition ».

Ce « Document Type Definition » est provisoire.

D'une part, je maîtrise mal le sujet, et il se peut que malgré mes soins, il soit incorrect sur la forme.
D'autre part, le format de fichier .geo est appelé à évoluer.

Il est à noter également qu'il est incomplet, dans la mesure où chaque développeur de module pourra rajouter ses propres balises, sous réserves de respecter les contraintes formelles mentionnées plus haut (section I.).

2. Contenu

A venir...