

UTILISATION D'ACTIVE X ET MICROSOFT .NET POUR LABVIEW

Fabien Marco Insa Rennes



Activex :

- Il désigne l'une des technologies du Component Object Model de Microsoft avec COM+ et Distributed COM utilisées en programmation pour permettre le dialogue entre programmes.
- Contrairement aux DLL (qui sont des bibliothèques partagées de fonctions et donc, à lier également au programme lors de sa conception), les contrôles ActiveX sont de véritables petits logiciels indépendants (mais dynamiques et encapsulés, donc, connectables à d'autres) qui peuvent avoir ou non, une interface graphique, non seulement lors du fonctionnement du logiciel les utilisant mais également lors de la création (dans l'IDE) afin d'aider le développeur à l'utiliser ou le configurer.

Microsoft .NET :

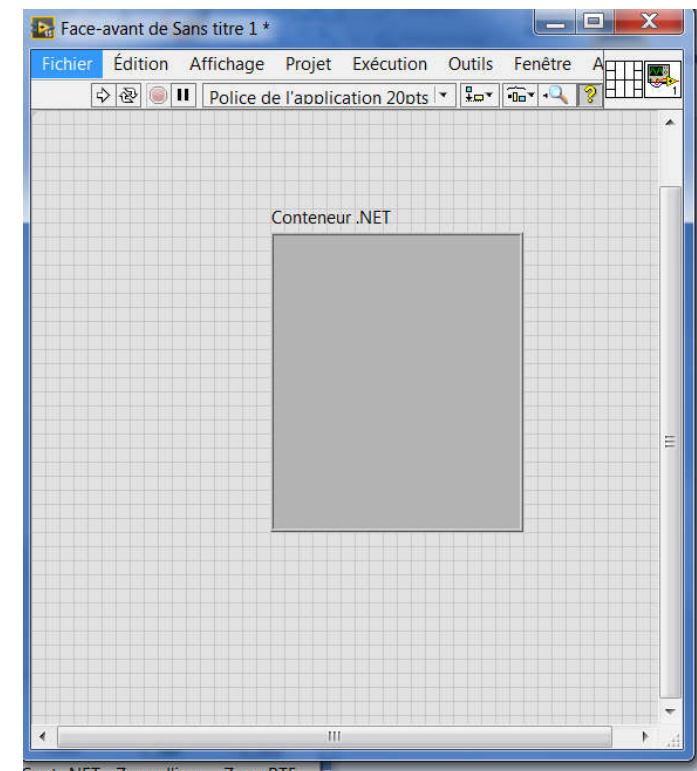
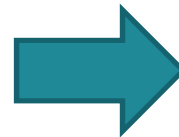
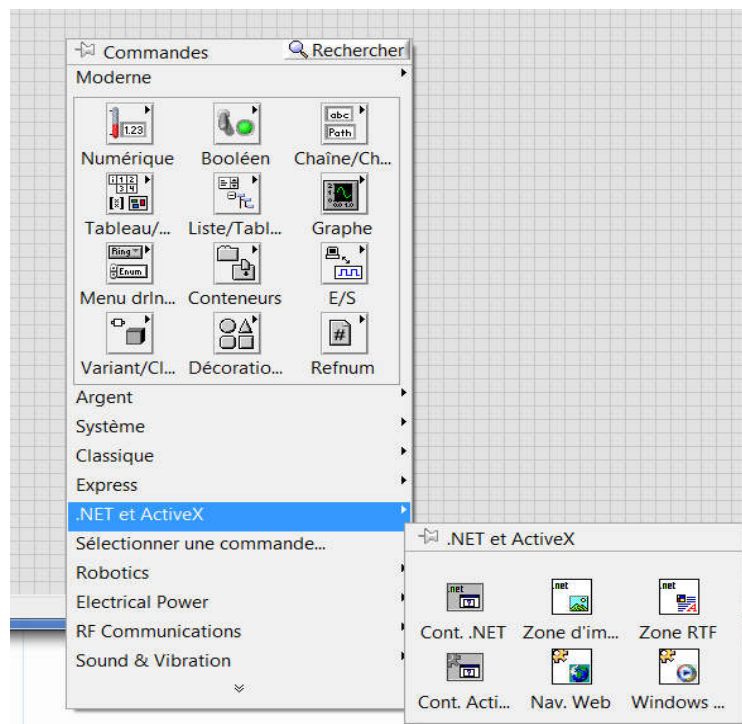
- Microsoft .NET (prononcé « dot net ») est le nom donné à un ensemble de produits et de technologies informatiques de l'entreprise Microsoft pour rendre des applications facilement portables sur Internet. Le but est de fournir un serveur web local permettant de gérer des services et évitant d'externaliser des données privées sur un service web de stockage ou un hébergement web tiers.

Activex a laissé au fur et mesure la place à l'environnement microsoft.net

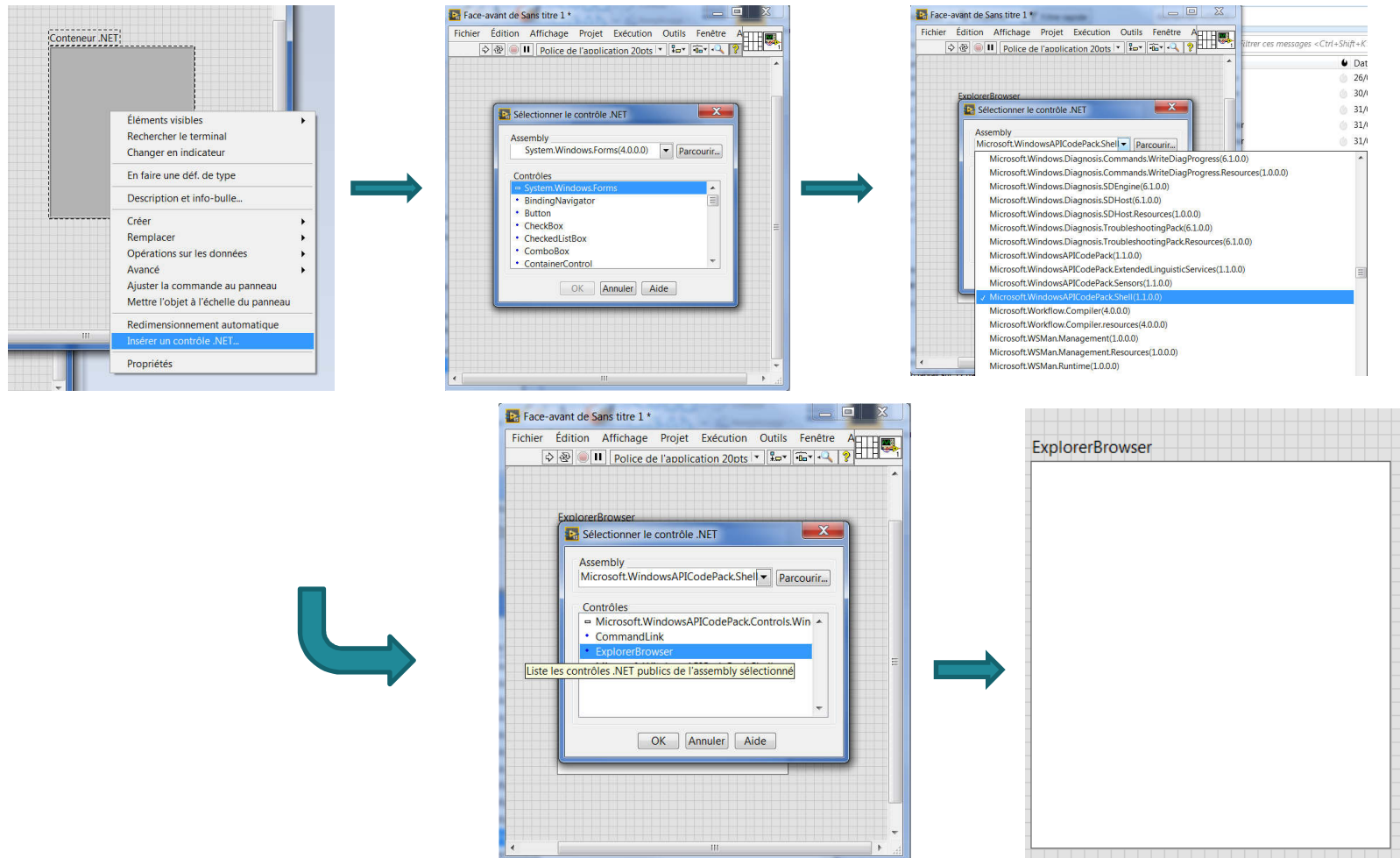
Activex et *.net permettent de contrôler un logiciel aussi complexe soit-il (bien sûr, il faut que le concepteur du logiciel ait laissé cette possibilité lors de la création de son logiciel), mais il permet surtout d'éviter de réinventer la roue.

Nous allons voir dans cet exemple comment appeler, utiliser l'explorateur windows, dans labview, afficher l'explorateur, récupérer le chemin d'un fichier sélectionné par l'utilisateur afin d'extraire des données, etc.

Afin d'appeler ce mini-logiciel, nous devons tout d'abord mettre dans notre face avant un composant (activex ou .net). Pour se faire aller dans la palette .net et activex, choisissez celui qui vous convient à votre application, puis déposer votre conteneur :



Une fois votre conteneur posé sur votre face avant, il faut ensuite choisir l'application qu'il va exécuter, pour se faire :



Une fois la face avant préparée, nous devons « câbler » le diagramme, celui-ci se fait exclusivement via des nœuds de propriétés et de méthodes.

Nœud de propriété :

Le nœud de propriété permet d'obtenir et de définir diverses propriétés d'application ou de VI.

Les exemples suivants montrent comment les propriétés peuvent faciliter l'utilisation d'une application ou d'un VI :

- Définir le focus de texte à la commande principale la plus utilisée.
- Désactiver ou masquer les commandes
- Guider l'utilisateur d'étape en étape en mettant des commandes en évidence.
- Changer les couleurs pour attirer l'attention sur les conditions d'erreur.

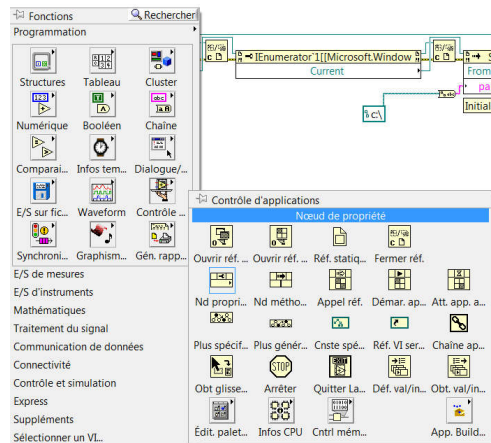
Nœud de méthode :

Le nœud de méthode permet d'effectuer des actions ou méthodes sur une application ou un VI. À la différence du nœud de propriété, un seul nœud de méthode n'exécute qu'une seule méthode sur une application ou un VI.

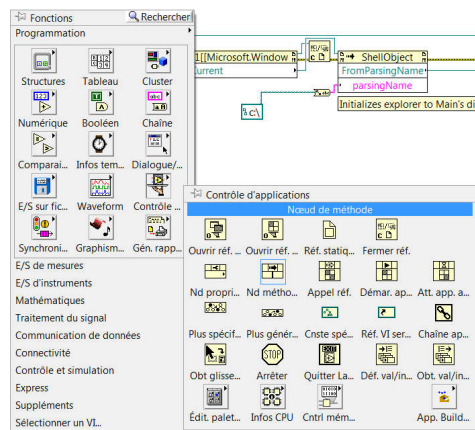
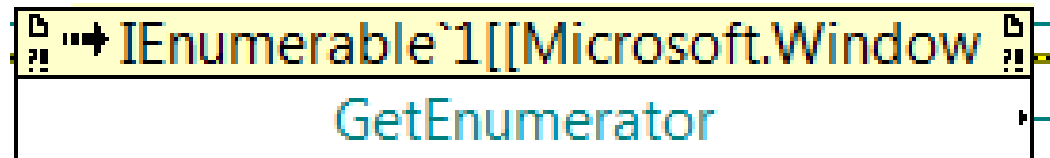
Par exemple pour un graphique :

- Exporter dans divers formats
- Rétablir la valeur par défaut

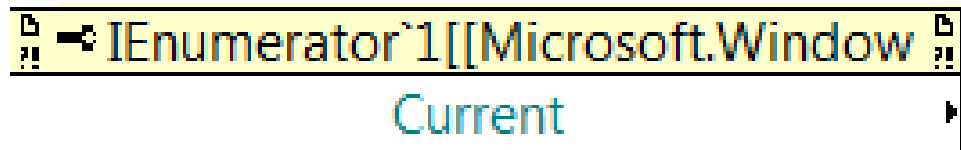
Pour se faire nous allons utiliser les deux éléments de la palette
Contrôle d'application:



Symbole de nœud de propriété



Symbole de nœud de méthode



Cahier des charges du programme

Description :

Nous souhaitons afficher une trajectoire sur un graphique 3D, afficher simultanément les données de celle-ci et récupérer les extractions de celle-ci (min et max pour chaque axe).

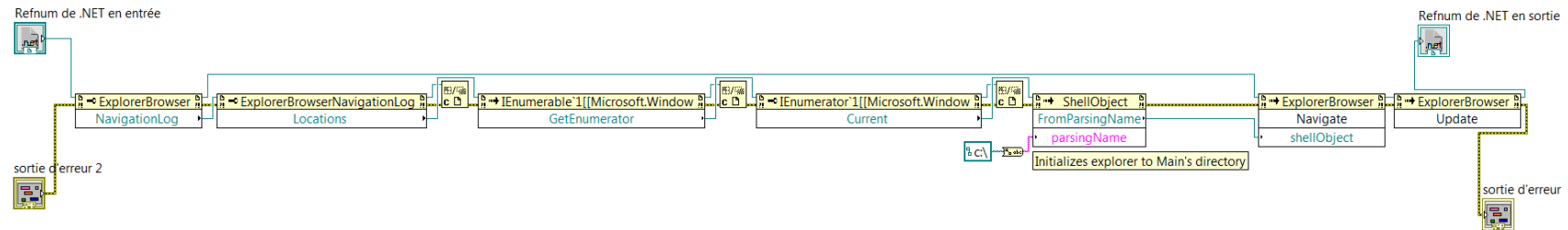
Il faudra :

- Arborescence de fichier (type windows)
- Visualiser les données lorsqu'un fichier est sélectionné
- Afficher le graphe de la trajectoire
- Extraire les données (min et max pour chaque axe)

Il sera donc plus simple d'utiliser directement l'explorateur de windows, celui-ci permettant de prévisualiser un fichier et proposant une interface connue de l'utilisateur.

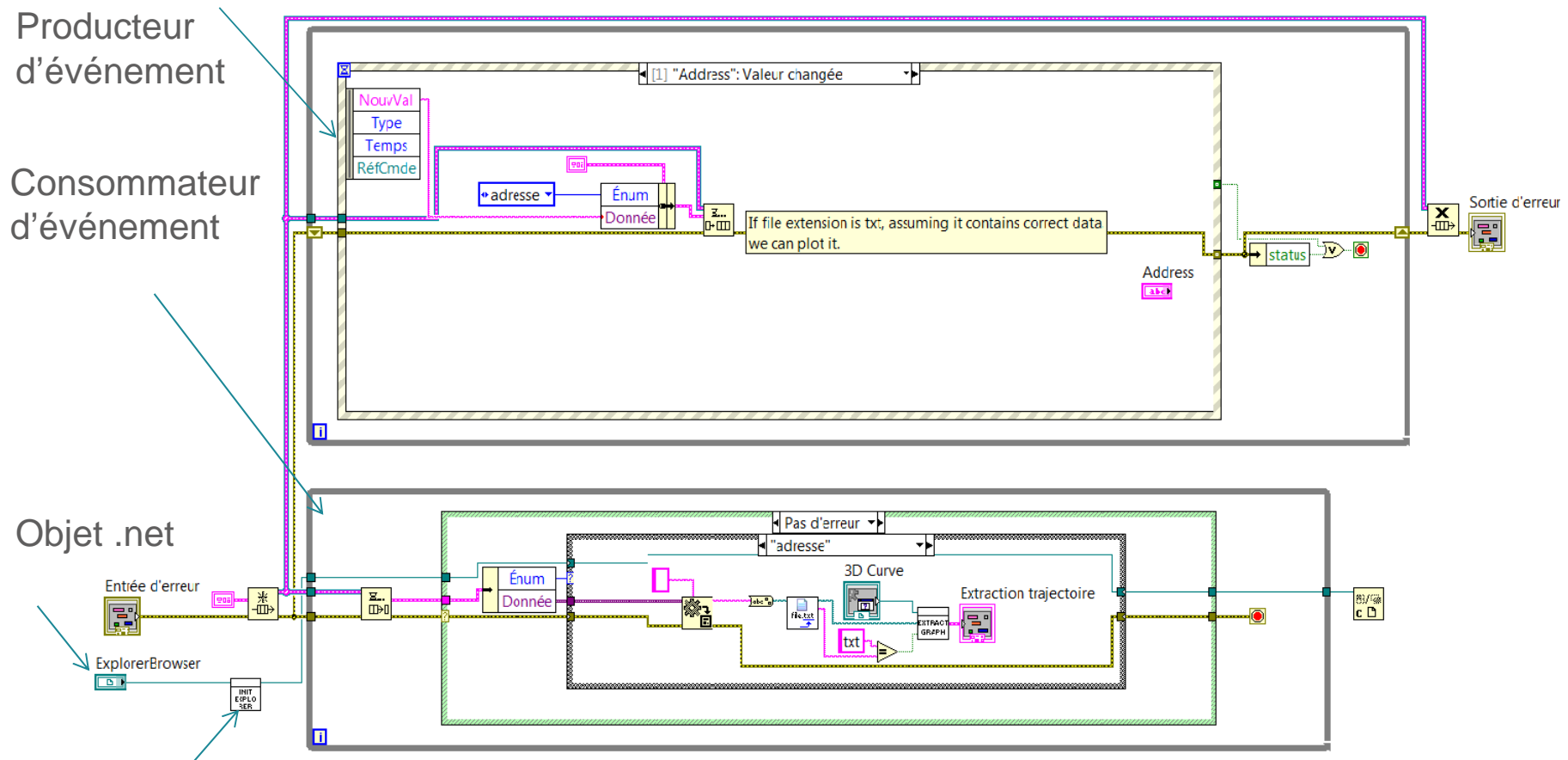


Une fois votre face avant configurée, vous allez interagir avec ce conteneur, et pouvoir contrôler le logiciel ouvert dans celui-ci.
Voici un exemple d'initialisation de l'explorateur :



Grace à ces quelques nœuds, nous avons configuré le chemin que doit ouvrir au départ l'explorateur, ici nous avons mis en dur le chemin (c:\) mais il est possible de mettre le répertoire de l'application, ou bien un chemin préconfiguré pour aider l'utilisateur.

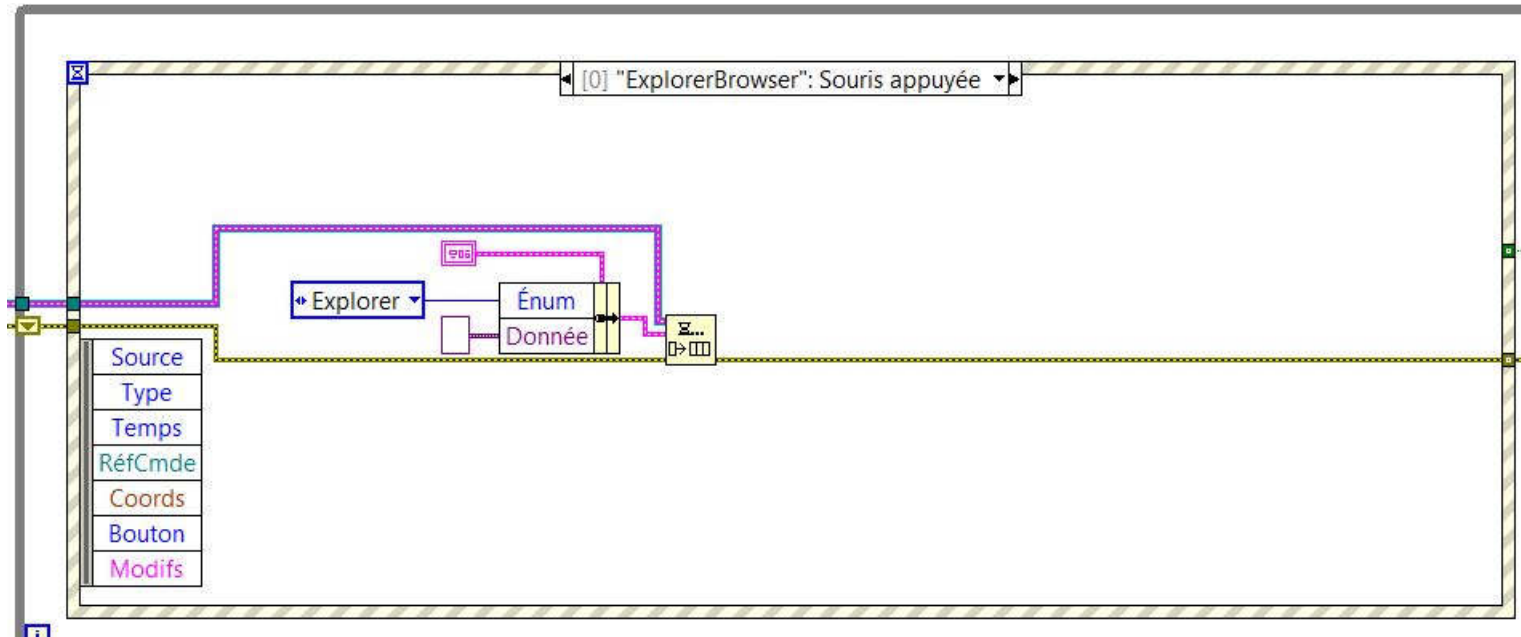
Nous allons utiliser ici, une structure producteur consommateur d'événement afin de récupérer les informations de navigation.



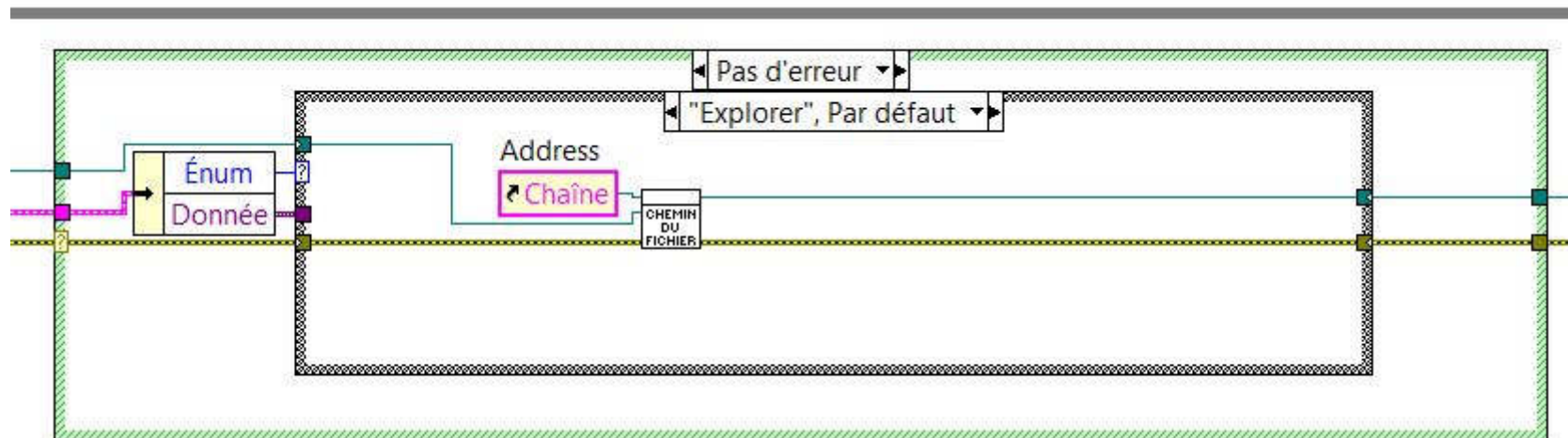
Initialisation de l'explorateur

Exemple d'utilisation du .net, l'explorateur windows.

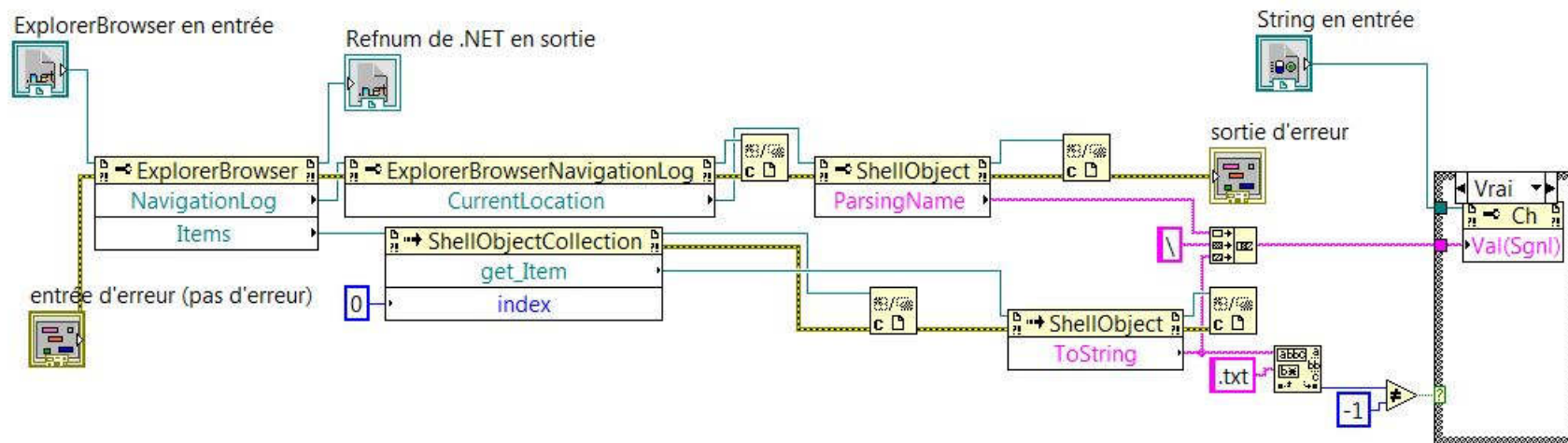
Le conteneur a les mêmes propriétés d'évènement que n'importe quel élément d'une face avant. Nous allons donc utiliser « souris appuyée » afin de détecter le clic de souris que va faire l'utilisateur quand il va sélectionner un fichier . On va lancer la commande « explorer » dans la liste d'énum.



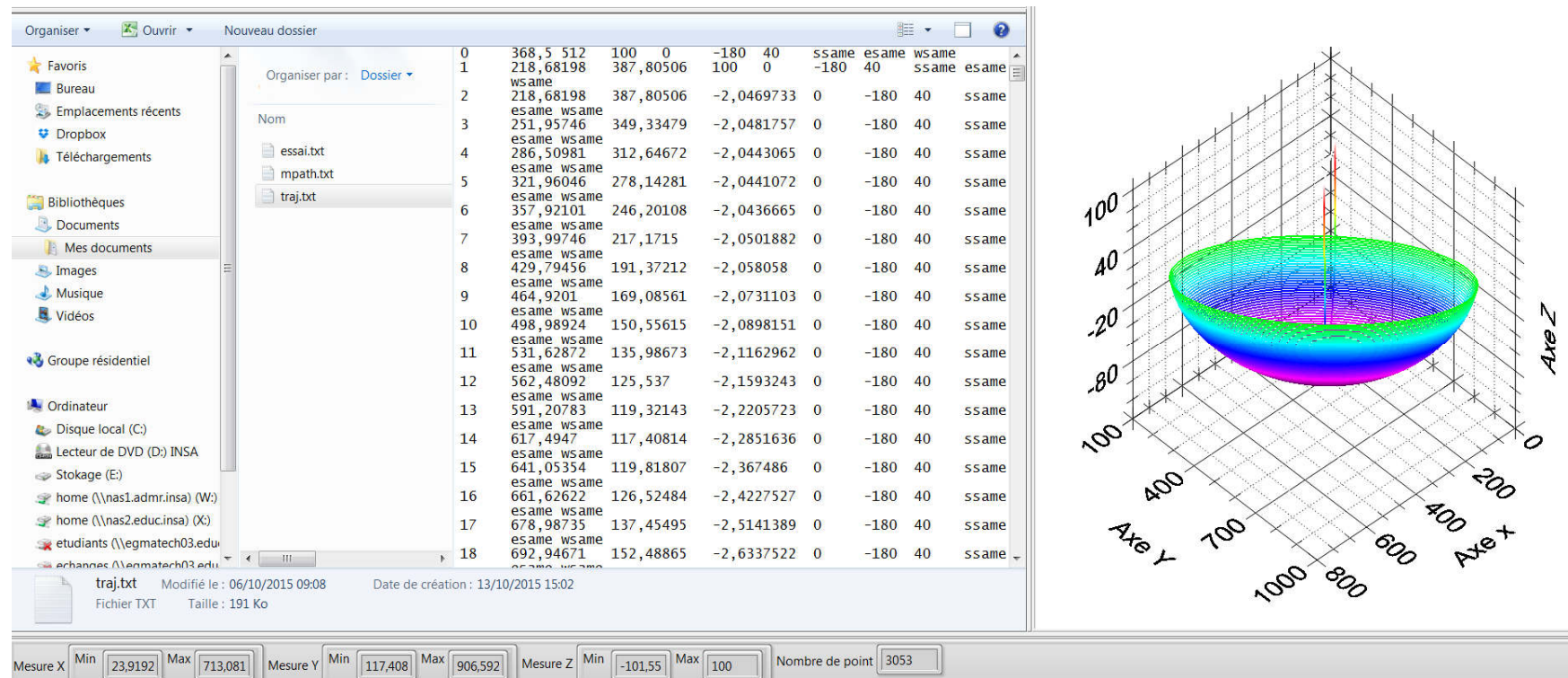
Le consommateur de données va filtrer les informations que l'on souhaite récupérer. En effet, l'utilisateur va naviguer dans l'arborescence en « cliquant » sur les dossiers. A ce moment là, il n'y a rien à extraire sinon cela occasionnerait des erreurs. On va extraire le code « explorer » afin d'opérer notre filtrage.



Voici la structure qui permet de faire un filtre de fichier *.txt. Ces fichiers contiennent la trajectoire que l'on souhaite afficher.



Une fois le chemin du fichier trouvé, nous allons pouvoir extraire les informations du fichier. Nous avons laissé la fenêtre de prévisualisation de l'explorer afin que l'utilisateur voit le contenu du fichier lorsqu'il ouvre le fichier (il met à jour la barre d'adresse)

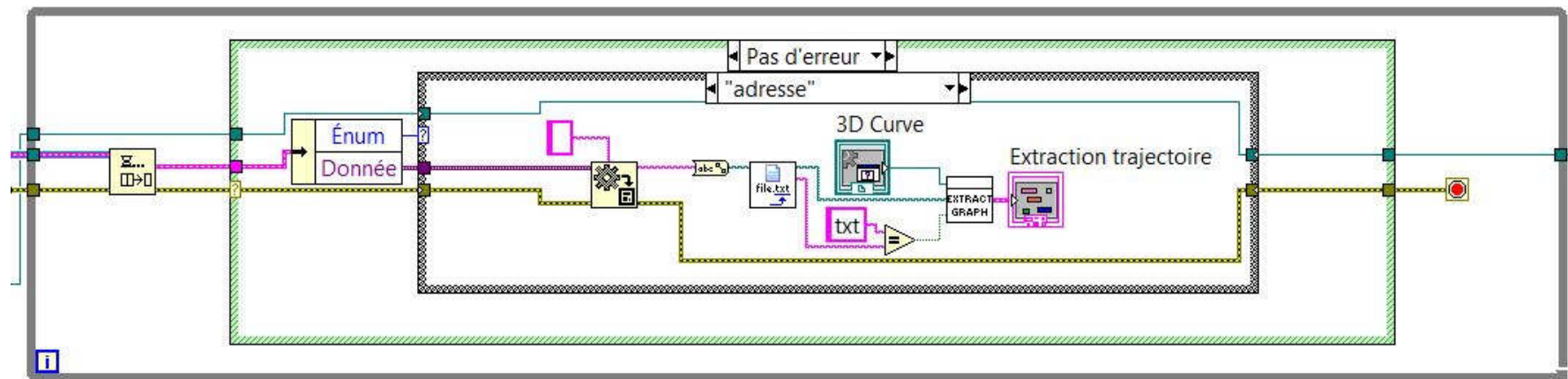


Dès que l'utilisateur sélectionne le fichier qu'il souhaite visualiser, le graphique de la trajectoire, et les informations utiles sont directement extraites du fichier (max et min pour chaque axe, nombre de points).

Pour se faire, nous avons utilisé l'adresse qui se trouve au dessus du conteneur. Qui nous donne le chemin du fichier à extraire.

Le changement de la barre d'adresse va provoquer un évènement qui va nous amener au cas « adresse » (attention il faut provoquer un déclenchement de la face, grâce à la commande Val(signl)).

Nous n'avons plus qu'à visualiser les données qui nous intéresse
:



Avez-vous des questions ?

