

CES471808

Comment implémenter l'automatisation au sein de votre organisation

David LICONA
Autodesk

Objectifs d'apprentissage

- Identifier les opportunités d'automatisation sur des processus de conception existants
- Penser analytiquement pour créer des solutions d'automatisation
- Développer des solutions industrialisables
- Mettre en place l'automatisation au sein des équipes

Description

L'industrie de l'infrastructure est en retard par rapport à d'autres industries en ce qui concerne l'automatisation lors de la phase de conception. Dynamo pour Civil 3D aide à résoudre ce problème, mais les entreprises ont du mal à industrialiser l'utilisation de l'automatisation au sein de leurs équipes. Quelles sont les opportunités d'automatisation ? Comment la mettre en place dans une équipe ? La présentation couvre l'implémentation de l'automatisation dans des processus existants : identifier les opportunités, développer une solution industrialisable et la mettre en place dans les organisations existantes. Nous utiliserons un cas courant dans l'industrie: les travaux de terrassement afin d'illustrer chaque étape pour mettre en œuvre l'automatisation ; tout en donnant les éléments clés pour réussir sa prise en main par les équipes: standardisation, codifications des fichiers, exigences technologiques, ressources clés, etc.

Speaker(s)

Guatémaltèque d'origine, français d'adoption. Je suis un ingénieur en génie civil avec plusieurs années d'expérience sur la conception, la construction et l'opération des infrastructures linéaires. J'ai eu la chance de pouvoir participer à différentes phases de projets d'infrastructure (émergence, conception, réalisation) et dans des rôles divers (MOA, MOE, entreprise de construction). Cependant, pendant toutes ses expériences, j'ai toujours cherché à me faciliter la vie en automatisant des processus, avec les moyens que j'avais à l'époque. Je ne supporte absolument pas de perdre mon temps avec des tâches sans valeur ajoutée. Aujourd'hui, j'ai la chance que mon travail consiste justement à faire ça, à aider les clients d'Autodesk à découpler la valeur qu'ils obtiennent de nos logiciels et à faciliter leur processus, pour se concentrer dans leurs objectifs et les tâches à forte valeur ajoutée.

Introduction

L'intérêt d'utiliser Dynamo vient de son accessibilité auprès des utilisateurs, qu'ils soient programmeurs ou non. Dynamo permet à l'utilisateur de programmer visuellement des tâches quotidiennes utilisées pour le traitement des données ou la conception de géométrie ainsi que d'étendre les capacités des logiciels interactives (tels que Civil 3D).

Dynamo en tant que technologie permet aux utilisateurs de développer des modèles paramétriques et d'automatiser les processus métier. Il peut aider à réduire l'effort d'exécution de tâches répétitives, à accroître la cohérence des processus de modélisation et d'information, à réduire les erreurs, à augmenter la productivité et à étendre les capacités des équipes de projet.

L'objectif de ce document est de décrire les meilleures pratiques pour l'identification des cas à automatiser, le développement de solutions et le déploiement de Dynamo pour une implémentation d'entreprise. Pour mieux comprendre comment développer une solution, un cas d'étude pratique sera décrit.

Limites du document

Ce document est fourni comme un guide des conseils pratiques pour mettre en place l'automatisation au sein d'une organisation. Il ne constitue pas un guide de formation de Dynamo pour Civil 3D.

J'attire l'attention sur le ton impersonnel que je vais prendre en rédigeant ce document. Je rédige souvent de la documentation technique et je me suis dit que ça serait plus sympa de tenter de faire autrement. C'est un premier exercice que je fais, à voir avec le temps si c'est une réussite...

L'intention est de fournir des informations sur les points pertinents des aspects de Dynamo qui ont été traités par des consultants Autodesk dans divers ateliers et présentations chez des clients.

Dynamo Foundation

Les informations données ici font partie d'une offre créée par Autodesk Consulting : Dynamo Foundation.

Cela est basé sur l'expérience des plusieurs consultants à travers le monde et dans tout type d'organisation et est composé donc des meilleures pratiques. Le but de cette offre est d'aider les clients avec les cas d'automatisation les plus complexes et à garantir le bon déploiement et l'adoption de l'automatisation en utilisant Dynamo.

Identifier les opportunités d'automatisation

Tout d'abord, il est nécessaire de définir le cadre dont nous parlons quand nous parlons d'automatisation.

Pour faire très simple, nous pouvons définir l'automatisation comme « La réduction des interventions humaines pour réaliser des tâches ».

Dans la pratique, nous pouvons voir l'automatisation un peu partout dans notre vie quotidienne, bien évidemment avec différents degrés d'automatisation. Mais je pense que nous avons tous été dans une situation où nous avons face à nous un fichier Excel avec une macro juste géniale, qui permet de mettre en forme le fichier, extraire les informations dont nous avons besoin et même imprimer le fichier. Le tout juste en utilisant un bouton.

C'est un peu la même chose dont nous allons parler dans ce document, mais cette fois-ci appliquée à nos conceptions d'infrastructure. Un bouton pour créer des points le long d'un alignement, un bouton pour exporter vers Excel les jeux de propriétés de notre dessin, un bouton pour faire la conception de nos plateformes de terrassement, etc.

Ce sont tous des exemples de ce que nous pouvons réaliser avec Dynamo pour Civil 3D.

Programmation Visuelle

La programmation visuelle est un type de programmation dans laquelle les programmes (le code) sont « écrits » par assemblage d'éléments graphiques. Pas besoin de connaître un langage particulier ni une syntaxe particulière. Juste besoin de connaître le logiciel ou l'interface que nous utilisons pour réaliser la programmation visuelle.



Figure 1 : Programmation Visuelle (source images : Photo par [Daniel Tuttle](#) et Photo par [Vitae London](#))

Si vous regardez les images ci-dessus, nous pouvons faire l'analogie avec un piano. La programmation visuelle est comme les touches du piano, derrière il y a un codage, des API, etc. que nous ne voyons pas, mais quand nous pressons une touche, nous écoutons de la musique.

Dans cet exemple, chaque touche représente une note en particulier, tout comme dans Dynamo, chaque nœud est une opération.

Dynamo pour Civil 3D

Dans Dynamo, nous avons donc des nœuds qui représentent des opérations et que l'on va relier les uns avec les autres pour créer des objets et des flux de travail automatisés.

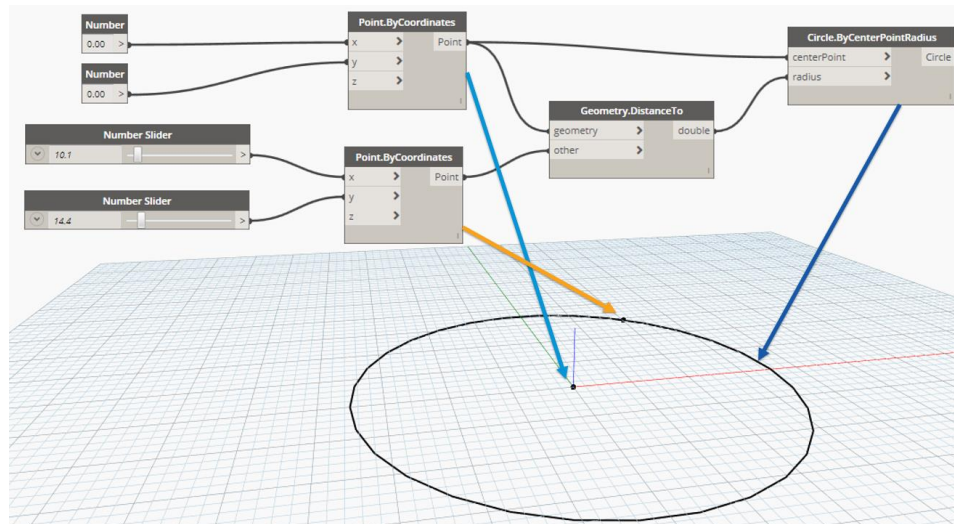


Figure 2 : Programmation visuelle sur Dynamo

Comme je l'ai dit, ce document n'est pas un cours de Dynamo et pour s'initier sur Dynamo, il y en a plusieurs sources d'information et notamment le Dynamo Primer. A la fin de ce document, je donne d'autres ressources pour en apprendre plus sur Dynamo.

Je vais donc prendre l'hypothèse que le lecteur des pages suivantes connaît un minimum Civil 3D et Dynamo pour Civil 3D. Il est là pour savoir plus concrètement comment faire pour être plus actif dans son utilisation de Dynamo, comment profiter de cet outil pour automatiser ses processus de conception et surtout comment mettre en place l'automatisation dans son organisation, au-delà de l'utilisateur ou l'équipe unique.

Il est à noter que je vais utiliser les termes « graphes » et « scripts » pour parler des programmes développés dans Dynamo.

But de l'automatisation

L'automatisation permet, entre autres, de :

- Traiter **plus** de cas ou des cas **plus** complexes ;
- **Travailler mieux**, en évitant les erreurs humaines et en améliorant la qualité ;
- Avec **moins** de ressources et en passant **moins** de temps.

Pour être plus concret sur les cas d'usages et commencer à les imaginer, l'automatisation aide à :

- Automatiser des tâches répétitives ;
- Créer des workflows complexes, connectant plusieurs sources des données ;
- Explorer des options de conception rapidement ;
- Accéder aux données du modèle, souvent obscures ;
- Créer et analyser rapidement des formes géométriques complexes.

Dynamo est particulièrement efficace pour en ce qui concerne les formes géométriques complexes, grâce à sa vaste bibliothèque de nœuds sur cette section (géométrie).

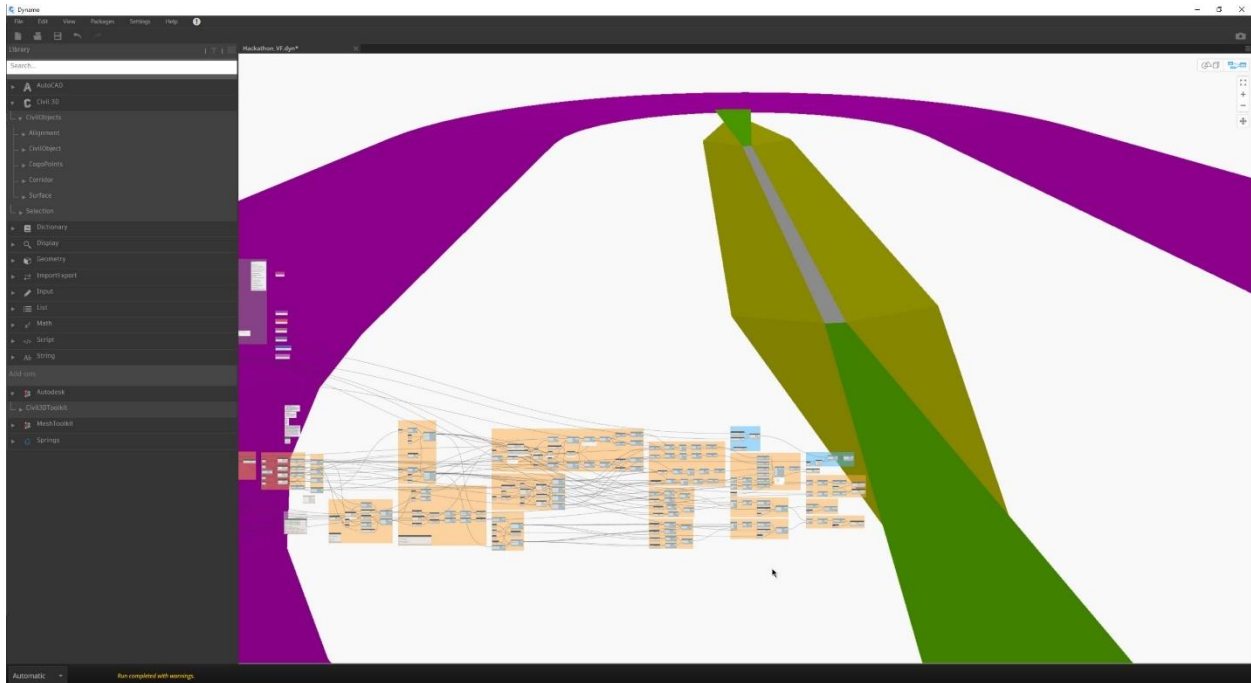


Figure 3 : Exemple de l'utilisation de Dynamo pour la création des solutions complexes

Enfin, il y a de la valeur pour tout type d'organisation, peu importe sa taille :

- Organisations petites : Elles peuvent concurrencer les plus grandes grâce à l'automatisation ;
- Organisations moyennes : Elles peuvent réduire les coûts en interne et réinvestir dans la recherche et l'innovation ;
- Organisations grandes : Elles peuvent déployer les meilleures pratiques plus efficacement.

Identification des opportunités d'automatisation

Alors comment faire pour identifier les tâches/processus qui peuvent être améliorés ?

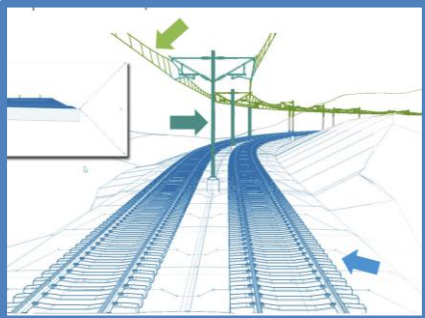
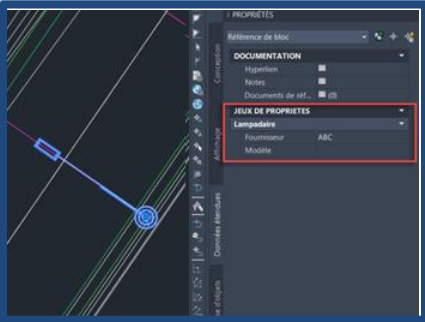
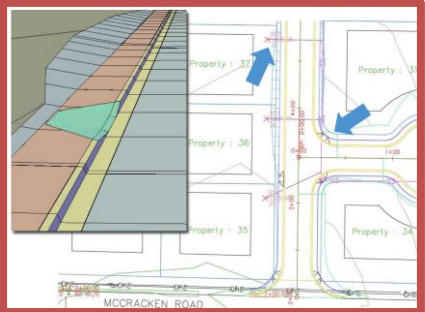
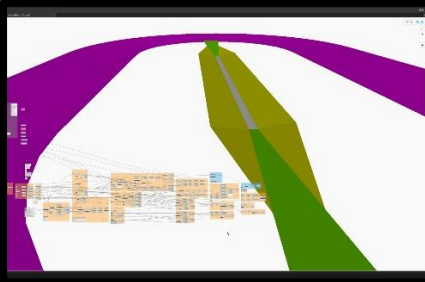
Il faut se poser des questions :

- « Est-ce que c'est une tâche répétitive ? »
- « Est-ce que je n'ai pas besoin de réfléchir pour la faire ? »
- « Est-ce que c'est une tâche à faire systématiquement, c'est-à-dire dans la majorité de mes projets ? »
- « Est-ce que cela prend trop de temps à faire manuellement ? »
- « Est-ce que c'est une tâche particulièrement sensible à l'erreur humaine ? »

Souvent, la réponse est oui pour plusieurs de ces questions. Une tâche qui est trop répétitive est très susceptible de générer des erreurs humaines.

Une autre façon simple d'identifier des tâches à automatiser est tout simplement réfléchir à ces tâches qui nous frustrant au quotidien. Bien souvent, ces tâches frustrantes sont répétitives avec peu de valeur ajoutée et donc idéales à automatiser.

Nous avons donc des cas d'usage métier, très bien représentés dans les [exemples de Dynamo](#) fournis avec l'installation ou dans des webinars disponibles sur [youtube](#), dans lesquelles nous pouvons trouver les caractéristiques décrites au-dessus :

Tâche répétitive	Placement des blocs le long d'un projet (traverses, lampadaires poteaux caténaires)	
Tâche systématique	Application des jeux de propriétés	
Tâche longue	Modélisation des rampes pour des trottoirs	
Tâche complexe	Surfaces imaginaires pour la conception aéroportuaire	

Classement des idées

Une fois une liste des idées d'automatisation a été établi, il faut les trier pour déterminer par lesquelles commencer. Pour cela je conseille d'utiliser une matrice dont je me servais souvent quand j'étais chef de projet. Cela consiste à donner une note aux idées sur deux critères principaux :

- Leur facilité de réalisation ;

- Leur potentiel impact.

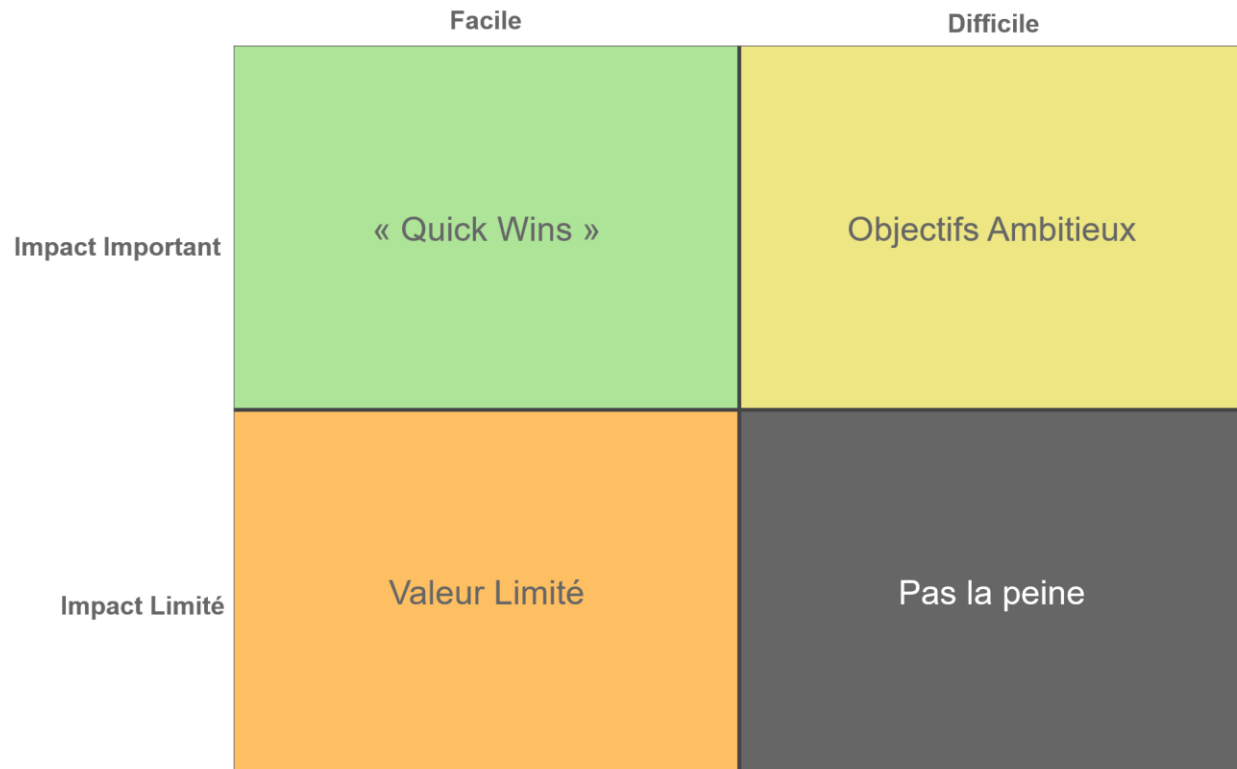


Figure 4 : Matrice proposée pour classer les possibles cas d'automatisation

Par exemple, si une idée est très facile à réaliser mais que cela concerne juste un métier ou très peu de projets, il s'agit donc d'une idée niche et à valeur limitée.

Au début, il faut se concentrer sur les « Quick Wins » : des cas à fort impact et faisables avec un minimum d'effort. C'est très important d'avoir des résultats visibles rapidement pour faciliter l'adoption de l'automatisation.

Bien évidemment pour juger si un cas est facile ou difficile à faire, cela dépend de l'expérience de chacun pour manipuler Dynamo, Civil 3D et des langages de programmation permettant de manipuler Dynamo.

La base pour plus d'automatisation

La programmation visuelle pour automatiser des tâches est juste une étape dans l'automatisation des process et ce n'est pas la plus puissante. Nous pouvons donc évoquer le « Generative Design » comme la prochaine étape, où nous pouvons automatiser la génération d'une multitude de variantes pour évaluer une solution. A la fin de ce rapport, je vous conseil un cours AU pour connaître plus sur le Generative Design.

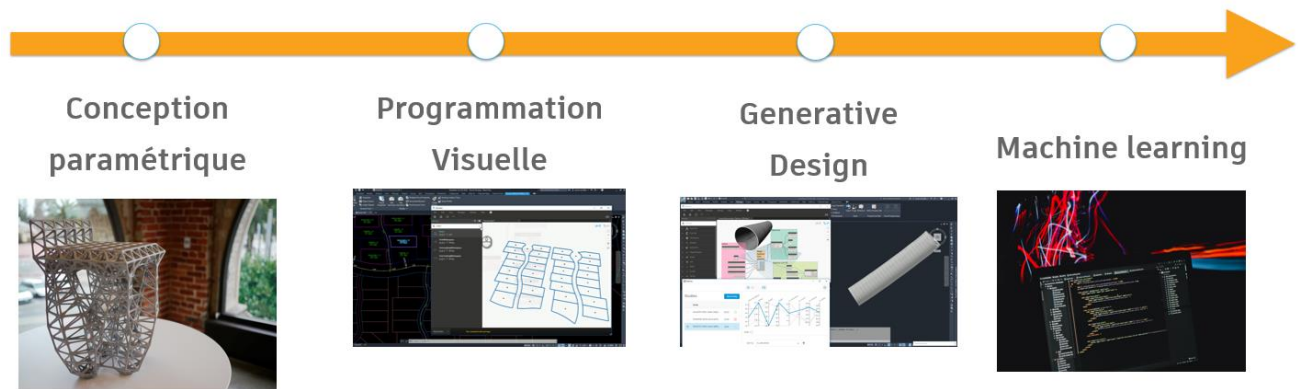


Figure 5 : De la Conception Paramétrique au Machine Learning (source : [AltumCode](#) - [Unsplash](#))

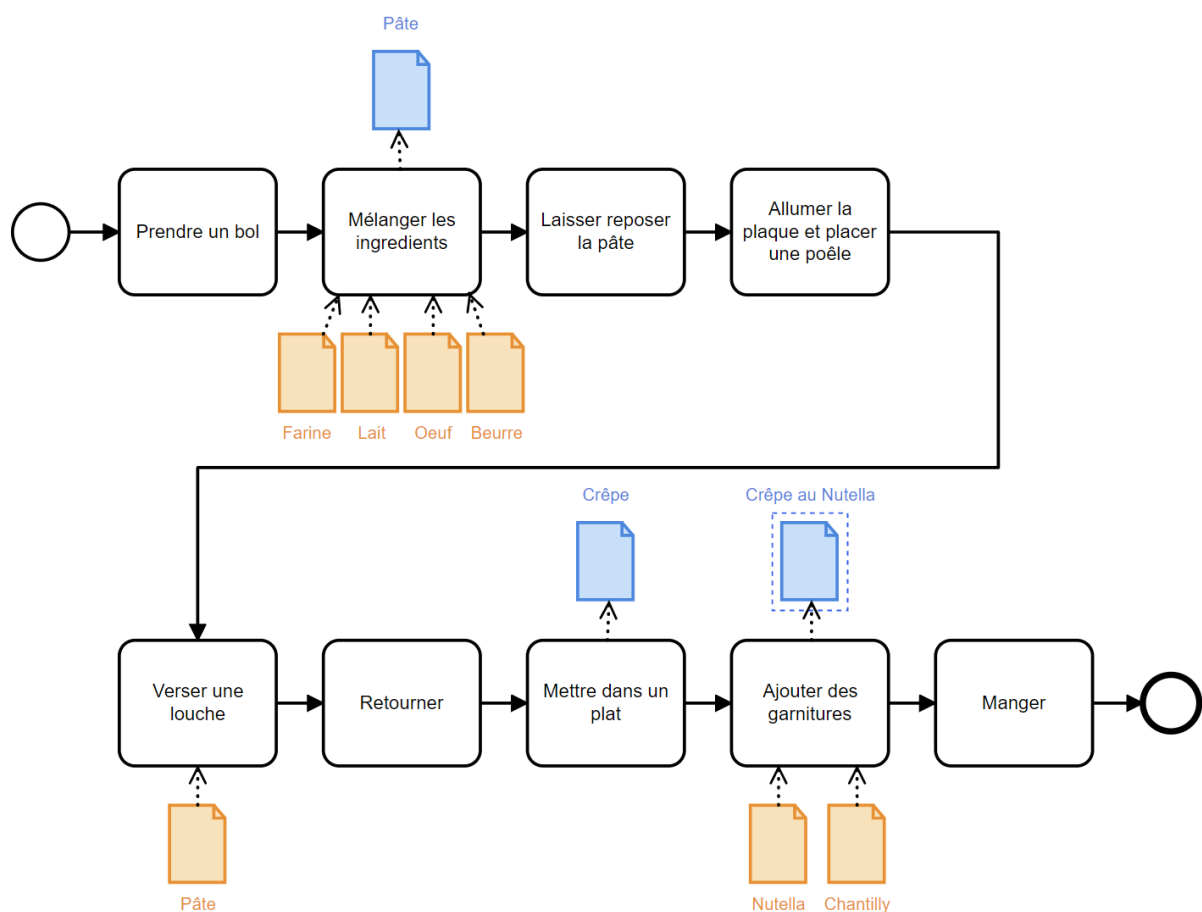
En conclusion

- L'automatisation nous aide à faire plus, mieux et avec moins de ressources ;
- Nous avons plusieurs cas d'usage, pour les identifier il suffit de regarder les procédures répétitives, systémiques, itératives ou avec peu de valeur ajoutée ;
- Une fois les cas susceptibles à être automatisés ont été identifiés, il faut prioriser les « quick wins », des cas avec un impact fort et un minimum d'effort ;
- Il ne faut pas avoir peur de se lancer dans l'automatisation.

Penser analytiquement pour créer les solutions d'automatisation

Quand je suis arrivé en France, une de premières choses que j'ai appris c'est comment faire des crêpes.

Je pourrais écrire la recette comment une séquence logique de tâches qui décrivent les étapes pour arriver jusqu'à l'objectif (les crêpes !). J'identifie les ingrédients dont j'en ai besoin en jaune, ils représentent les entrées nécessaires pour la réalisation d'une tâche. En bleu j'identifie mes résultats intermédiaires, les sorties produits par une tâche.



Ecrire les étapes pour résoudre une problématique avant de s'y lancer est une très bonne pratique. Cela permet de :

- Visualiser la problématique comme une séquence des tâches logiques à suivre ;
- Se rendre compte si nous avons loupé une étape ;
- Vérifier si les résultants sont ceux que nous attendions ;
- Vérifier si c'est possible de récupérer les entrées dont nous avons besoin.

Nous pouvons donc très vite réaliser si nous sommes en train de sauter des étapes, ou si les sorties attendues ne correspondent pas avec les entrées nécessaires.

Outils

Pour construire ce type de diagramme, je recommande d'utiliser [Cawemo](#). Un outil gratuit, en ligne, très simple à utiliser, un peu limité vu sa simplicité mais qui donne un gros avantage : il permet de construire des diagrammes de forme collaboratifs, sur lesquels plusieurs personnes peuvent faire des modifications en même temps.

Développement des solutions

Lors de la construction des graphes Dynamo, il faut avoir en tête que nous avons besoin de graphes qui soient :

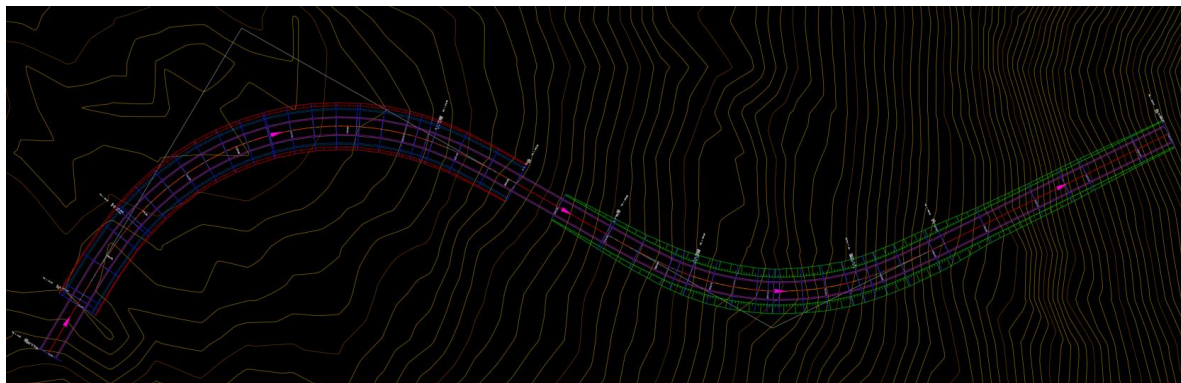
- Performants ;
- Génériques ;
- Extensibles

Mais surtout, nous voulons des graphes qui fonctionnent. Je conseille donc de réaliser d'abord une « Proof of Concept (POC) » avant de se lancer dans la création d'un graphe performant.

Cela permet de savoir rapidement si ce que nous voulons faire est réalisable. Nous pouvons ensuite changer le graphe pour qu'il ait les caractéristiques mentionnées ci-dessus.

Dataset Générique

Pour réaliser des tests, j'utilise un data set générique et très simple. Il s'agit d'un petit projet 3D. Il n'est pas du tout représentatif de la réalité mais permet de faire tourner le graphe Dynamo beaucoup plus rapidement.



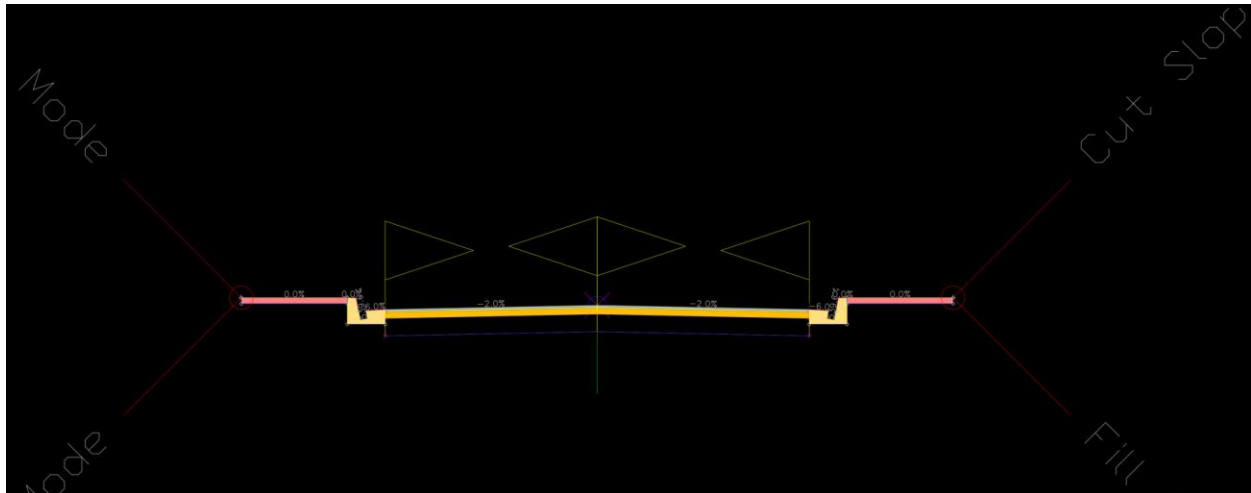


Figure 6 : Dataset générique

J'ai aussi trouvé que les cas d'exemple fournis avec l'installation de Civil 3D sont un bon point de départ à utiliser pour les tests. Vous pouvez trouver ces fichiers [ici](#) :

Quelques concepts clés sur Dynamo

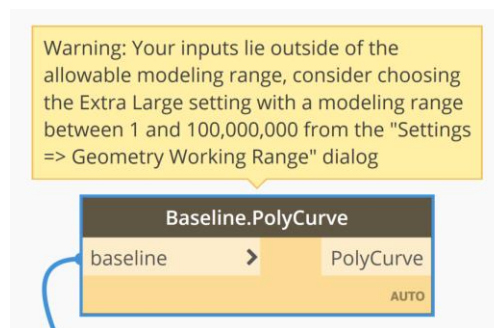
Même si ce cours est adressé aux personnes qui connaissent déjà Dynamo, il est important d'avoir quelques concepts en tête lors de la création des graphes.

Intervalle de Géométrie

Un des avertissements que nous rencontrons le plus souvent dans Dynamo concerne l'intervalle utile de géométrie. Cet avertissement apparaît quand nous travaillons loin du 0,0 0 dans Dynamo et donc il apparaît systématiquement quand nous travaillons avec des modèles géolocalisés.

En soit, cet avertissement ne devrait rien changer sur l'exécution du graphe et il devrait s'exécuter sans problème. Il est d'ailleurs recommandé de ne **JAMAIS** changer l'intervalle de géométrie, car la précision sera réduite et nous risquons de perdre des données.

Cependant, quand nous utilisons le Dynamo Player pour exécuter un graphe, il dira que c'est fait avec des erreurs et cela peut être problématique pour des utilisateurs qui ne connaissent pas Dynamo. Ils peuvent perdre confiance dans le résultat. Pour effacer ces avertissements, il est possible d'utiliser le nœud « Function Apply »:



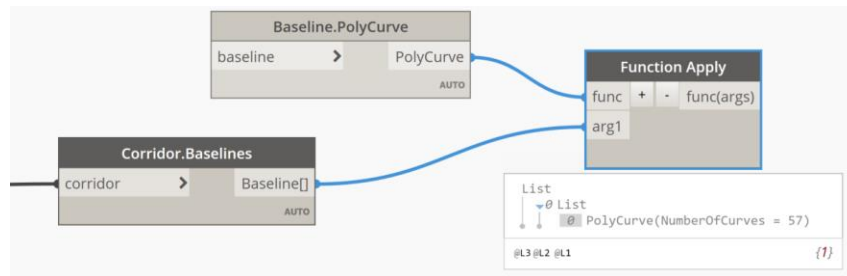


Figure 7 : Utilisation du « Function Apply » pour éliminer des avertissements liés à l'intervalle de géométrie

Il faut noter que dans certains cas, si la modélisation ne se réalise pas correctement, il pourra être nécessaire d'envoyer la représentation du modèle dans Dynamo proche du 0, 0, 0 et réaliser les opérations nécessaires là-bas. Voici une [entrée](#) sur le forum concernant ce sujet.

Binding

Dynamo créera une liaison entre le graphe et les éléments créés par lui-même dans le modèle. Cela veut dire que Dynamo va se rappeler quels ont été les éléments qu'il a créés. Si nous faisons tourner un graphe plus d'une fois sur le même modèle, il va effacer les éléments créés dans les exécutions antérieures du graphe. Cela est très important pour le placement des blocs. Il faut donc soit placer tous les blocs d'un coup, soit avoir des graphes différents pour le faire. Pour plus d'information voici une [discussion](#) du forum de Dynamo à ce sujet.

Objets temporaires

Certains nœuds qui interagissent directement avec des objets Civil 3D et des objets AutoCAD vont créer des objets sur AutoCAD. Parfois aussi, nous avons besoin des créer des objets tels que des polygones temporaires pour nous aider dans l'exécution du graphe. Nous ne souhaitons pas polluer le dessin original alors il est possible de créer un code block pour rassembler tous les objets temporaires créés et les supprimer en utilisant le nœud « Object Delete » :

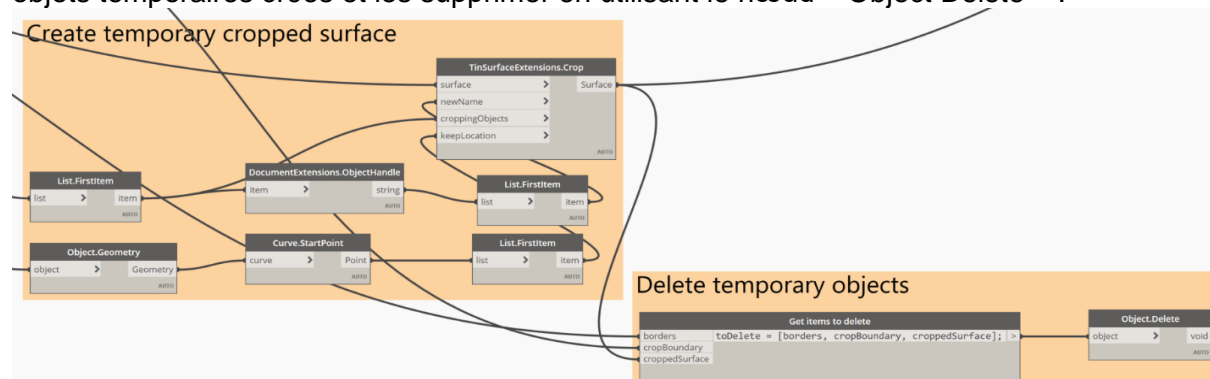


Figure 8 : Suppression d'une surface temporaire lors de notre cas d'étude

En conclusion

- Décrire la séquence logique pour résoudre la problématique afin de mieux visualiser le process, tout en indiquant les entrées et sorties ;
- Créer un « POC » rapidement, utiliser des datasets génériques pour aller plus vite ;
- Ne pas changer l'intervalle de géométrie et avoir en tête le binding et l'utilisation des objets temporaires

Développement des solutions industrialisables

Dans la mesure du possible, nous voulons avoir des graphes qui fonctionnent avec le Dynamo Player. C'est-à-dire qu'ils peuvent être exécutés directement depuis Civil 3D et sans besoin d'aller dans Dynamo.

Le Dynamo Player permet de changer les données d'entrée ainsi que des paramètres, permettant de tester plusieurs solutions rapidement (par exemple le placement d'un bloc tous les 25, 50 ou 100 m). Il permet aussi de déclarer des sorties du graphe pour montrer des informations à l'utilisateur.

Le Dynamo Player permet d'exploiter la logique (graphe) créée par quelqu'un d'autre. Aucun besoin de connaître Dynamo mais seulement la fonction du graphe. Cela est très puissant car il n'y a pas besoin de former toutes les personnes à Dynamo mais juste celles qui vont construire les graphes. Cependant, toutes les personnes peuvent potentiellement profiter des graphes.

Instructions

Les graphes construits avec Dynamo doivent être imbriqués de certaines informations afin de garantir leur bonne utilisation et surtout en cas de problèmes. C'est mieux d'avoir l'information directement dans Dynamo que dans un document externe qui puisse se perdre facilement. Les informations suivantes doivent être incluses dans un graphe :

- Le but du graphe ;
- La version ;
- Les hypothèses prises ;
- Les données d'entrée attendues ;
- Les optimisations possibles ;
- Les problématiques identifiées ;
- Les résultats à attendre ;
- Les tests qui ont été réalisés ;
- L'auteur (qui contacter en cas de problème) ;

Modularisation

En fin, pour avoir des graphes performants, génériques et extensibles ; il est conseillé d'avoir des graphes qui réalisent une fonction. C'est mieux d'avoir 4 graphes différentes qui ont 1 seule fonction chacun, que 1 graphe qui réalise 4 fonctions.

Cela facilite la construction et réalisation de tests, la compréhension par les utilisateurs et la recherche des erreurs lors des problèmes.

En conclusion

- Faire des solutions qui puissent être utilisées avec le Dynamo Player ;
- Imbriquer les graphes avec des notes ;
- Modulariser les solutions pour tester et débbugger plus facilement.

Cas d'étude

Nous utiliserons comme cas d'étude un projet dans lequel j'ai travaillé cet été.

Le but du projet consistait à automatiser la modélisation des terrassements d'une autoroute en tant qu'objets, afin de pouvoir réaliser du BIM 4D et 5D.

En effet, lors de la conception autoroutière sur Civil 3D, les terrassements ne sont pas modélisés en tant qu'objets.



Figure 9 : Modèle Civil 3D fourni par le client

Un utilisateur de Civil 3D sait qu'il y existe une fonctionnalité permettant justement de créer de solides entre deux surfaces. Cependant, réaliser le processus manuellement peut prendre beaucoup de temps et, selon les conditions du modèle, des problèmes peuvent être générés (par exemple, la création des solides entre surfaces ne marche pas avec des surfaces composées de plus de 100K triangles).

D'où l'idée de créer un flux de travail automatisé qui ferait tout cela dans quelques clics. Nous cherchons donc à créer un script Dynamo pour :

- Modéliser des solides 3D représentant les terrassements pour n'importe quel projet 3D ;
- Diviser les solides :
 - Déblai/Remblai ;
 - Par la ligne de base ;
 - Par corps de terrassement.
- Ajouter des jeux de propriétés et des propriétés :
 - Hauteur maximum ;
 - Hauteur moyenne.

Le tout dans 4 jours pour réaliser l'ensemble des tâches : réunions avec le client, développement, documentation finale.

Décomposition de la problématique

Voici la traduction du résultat attendu dans un workflow :

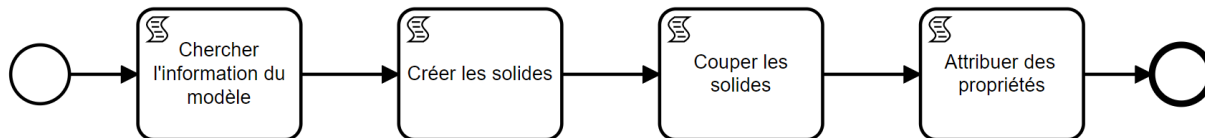


Figure 10 : Flux de travail représentant le script à créer

Une fois le problème a été décortiqué, il faut continuer par :

- Identifier les entrées ;
- Identifier les sorties ;
- Et surtout, **identifier les tâches que vous ne savez pas faire.**

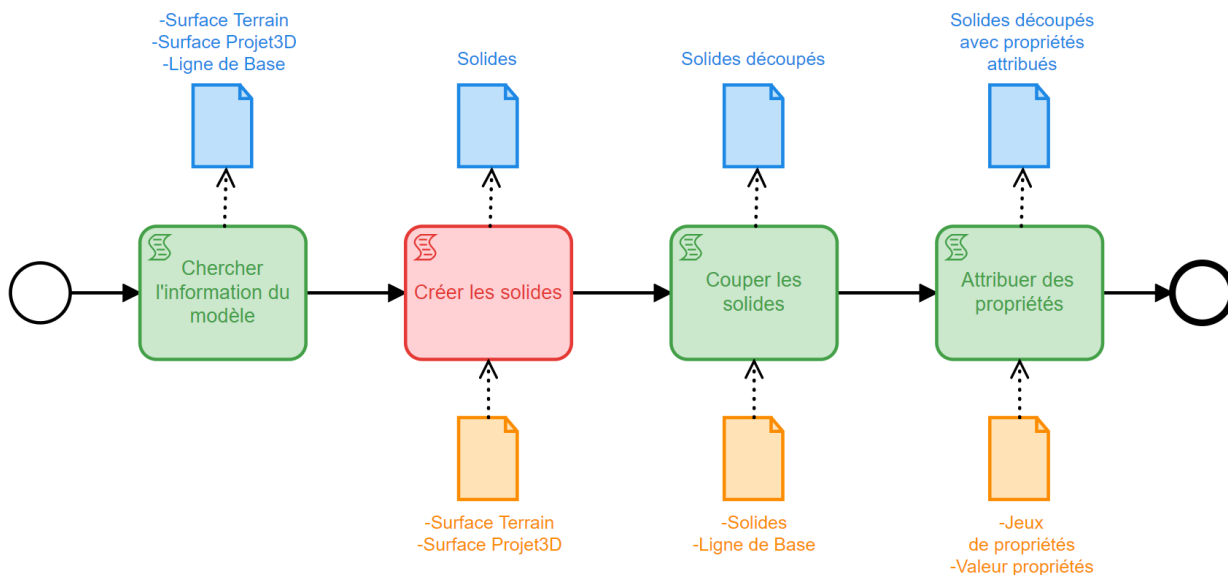


Figure 11 : Identification des entrées/sorties et tâches compliquées

A ce moment, dans Dynamo, il n'y avait aucun nœud permettant de réaliser des solides entre deux surfaces. J'ai donc discuté avec mon collègue Paolo Serra et nous avons identifiés trois possibilités :

- Vérifier si la commande pour créer les solides entre deux surfaces était disponible sur l'API.
- Utiliser les lignes caractéristiques du projet 3D pour réaliser des surfaces et puis créer les solides ;

- Utiliser les profils en travers pour dessiner les sections et créer les solides en lissant les sections ;

La commande était disponible dans l'API et Paolo a donc pu développer des nœuds pour appeler les fonctionnalités. Nous en parlerons plus en détail après mais **cela montre l'importance d'avoir de ressources spécialisées au sein de votre organisation.**

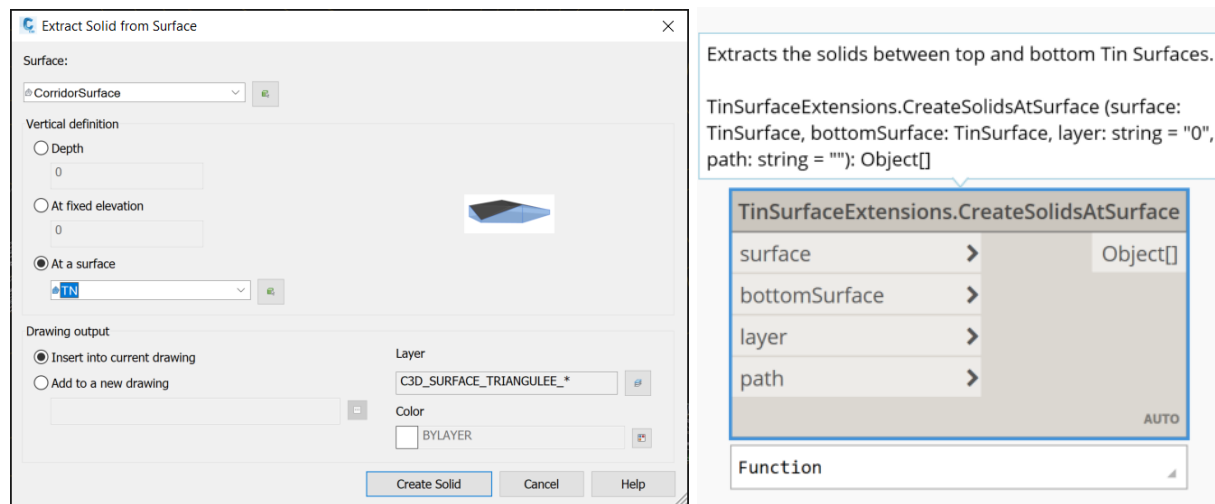
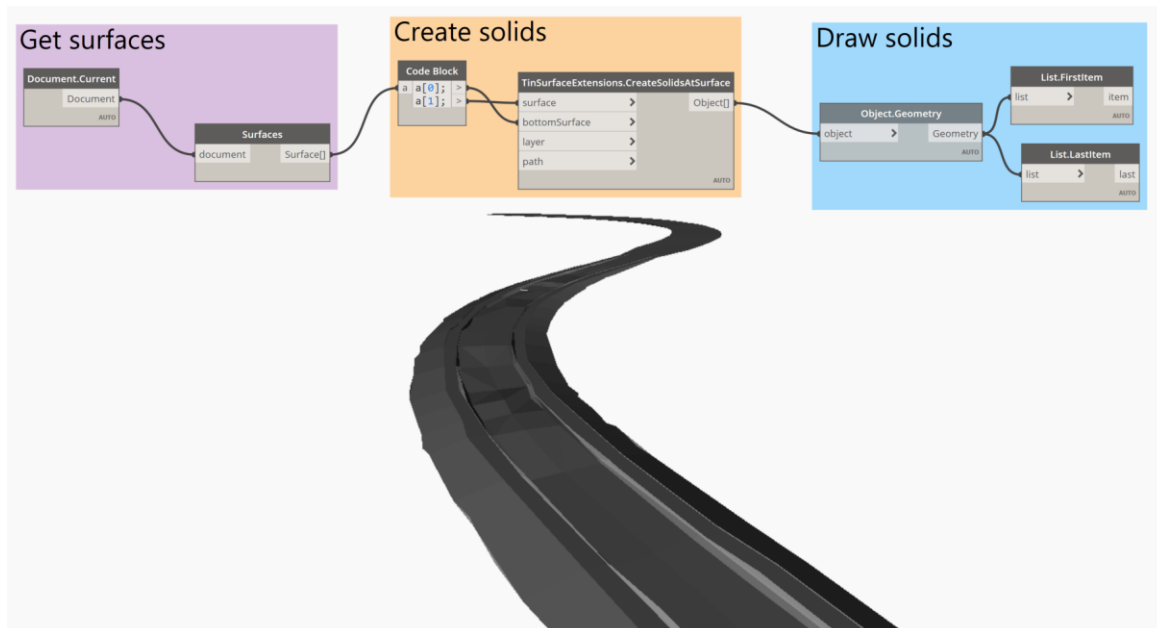


Figure 12 : Nœud développé pour créer des solides entre surfaces

Création d'un POC

La première étape consiste à tester les différentes fonctionnalités attendues par le graphe. La figure ci-dessous montre un test, très rapide à réaliser qui montre la création des solides entre deux surfaces (ce dont nous avons besoin). Nous pouvons observer que la façon dont l'information est récupérée depuis Civil 3D n'est pas idéale et doit être amélioré ultérieurement.



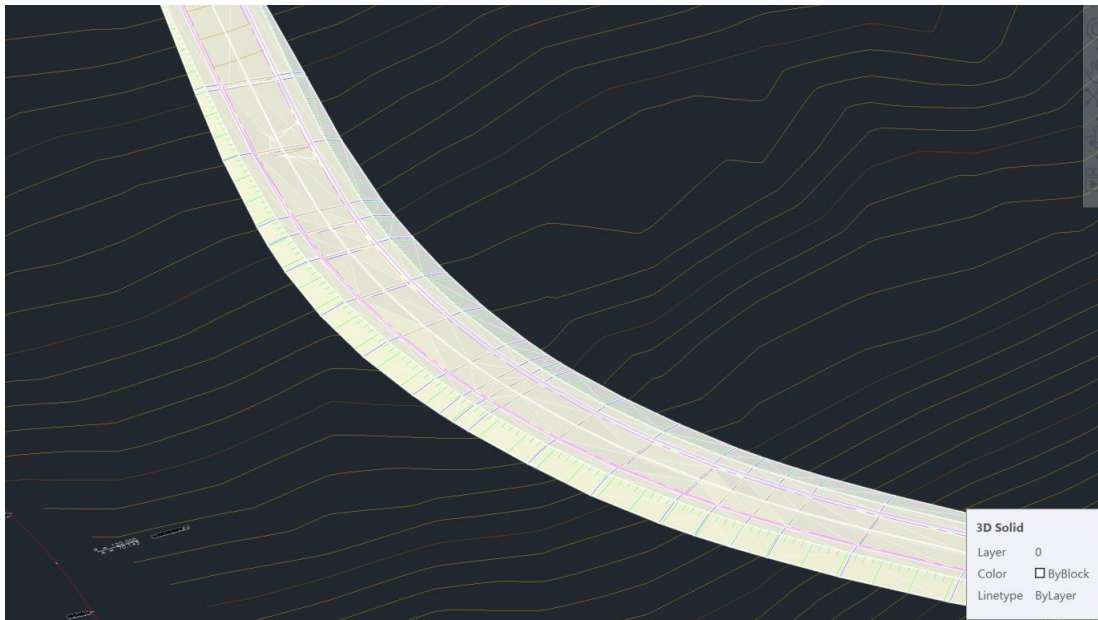
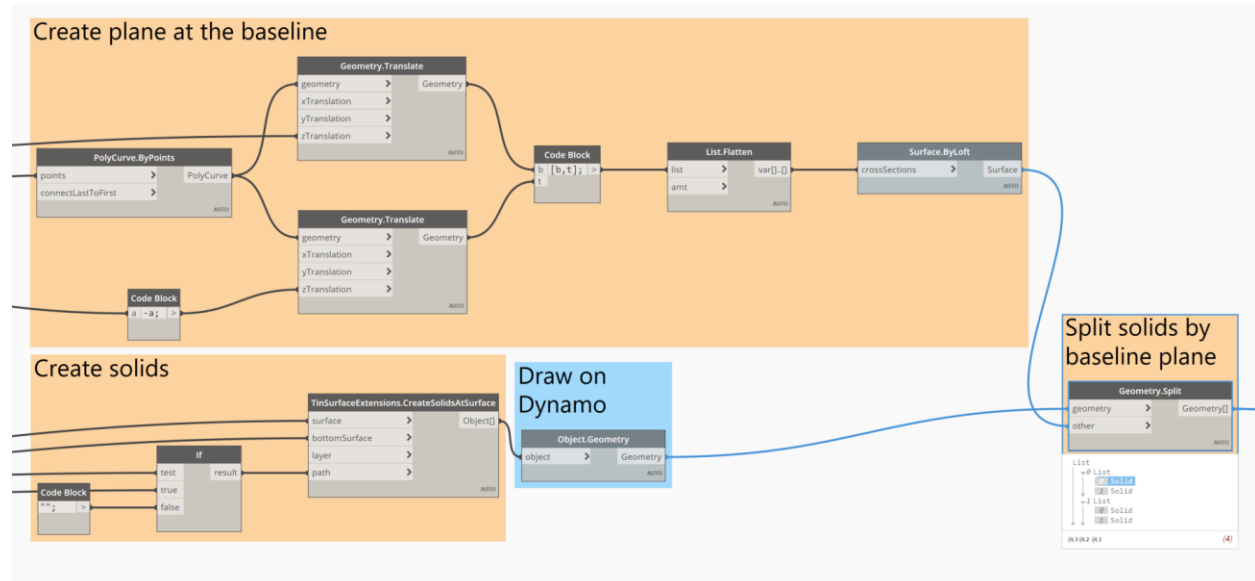


Figure 13 : Résultat d'un POC pour créer de solides entre une surface de terrain et une surface du projet 3D

Pour prouver que les solides pouvaient être découpés. Le test suivant a été fait pour les couper à la ligne de base :



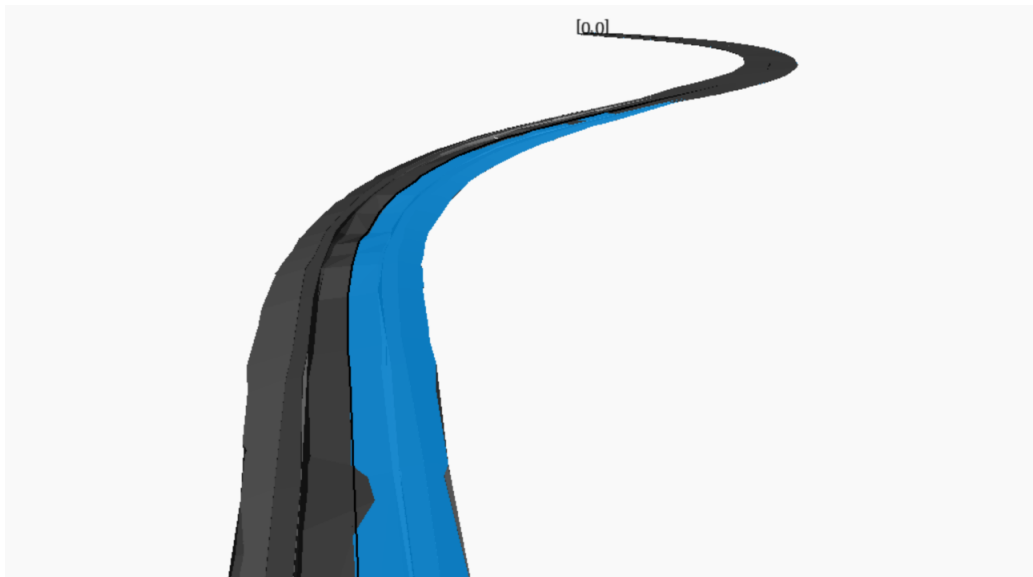


Figure 14 : Test pour découper les solides

De la même façon, l'application des jeux de propriétés a été testée :

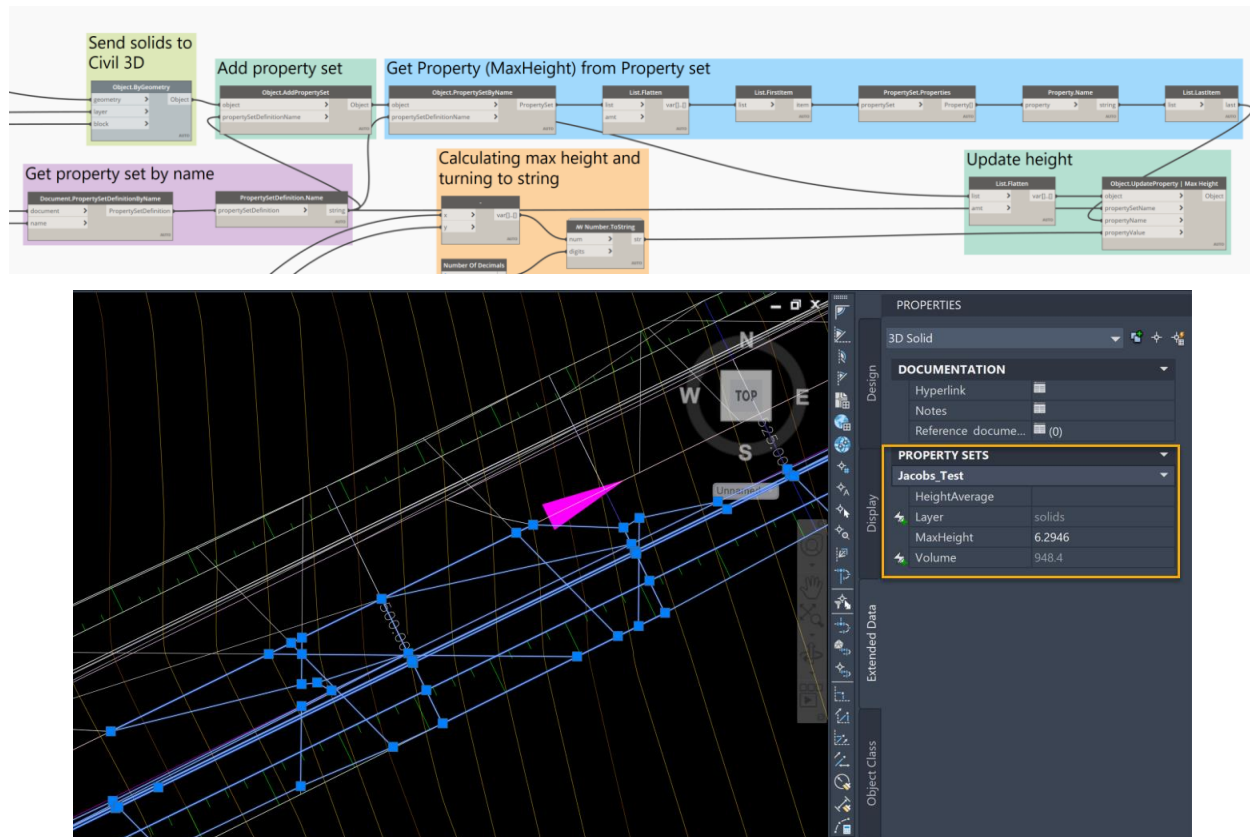


Figure 15 : Test pour application des jeux de propriétés

Voici un aperçu du graphe complet :

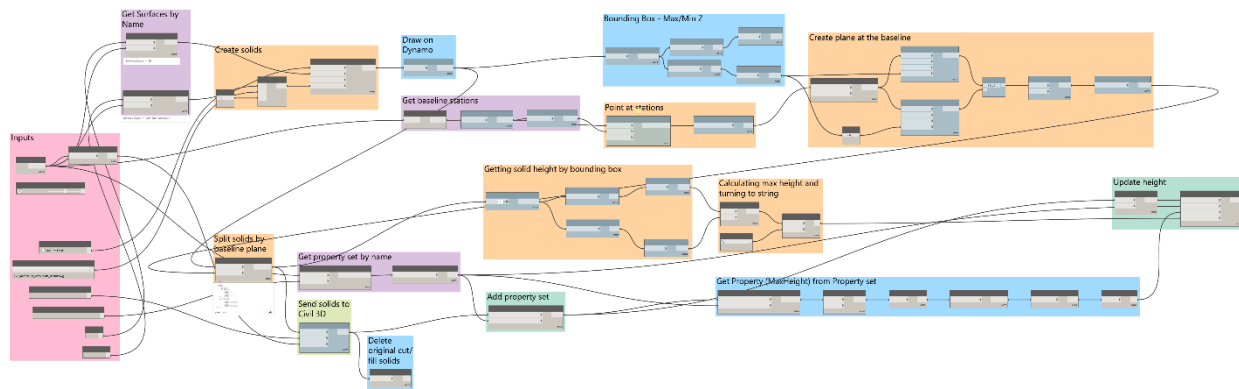


Figure 16 : POC pour réaliser des solides représentant les terrassements d'une autoroute

La réalisation de ce POC permettait de prouver que la logique pensée pour la construction du graphe fonctionnait. A ce moment, il fallait donc changer le graphe pour pouvoir l'industrialiser et accomplir tous les objectifs.

Industrialisation

Pour pouvoir industrialiser le graphe, les modifications suivantes devaient être faites au script ci-dessous :

- Faire les choix des entrées paramétriques pour être utilisés depuis le Dynamo Player ;
- Réaliser le découpage de solides comme demandé par le client, c'est-à-dire par corps de terrassement ;
- Calculer toutes les propriétés requises, c'est-à-dire ajouter la hauteur moyenne ;
- Tester sur les vrais modèles fournis par le client.

Le choix a été fait de modulariser le graphe pour faciliter ces opérations :

1. Création de solides ;
2. Manipulation de solides (découpage par corps de terrassement) ;
3. Application, calcul et mis-à-jour des jeux de propriétés.

Dans les premiers tests réalisés dans le modèle fourni par le client, les solides n'étaient pas créés. Grâce aux tests réalisés avant, nous savions que le nœud Dynamo marchait bien et donc que le problème venait de Civil 3D. En essayant de répliquer l'opération manuellement, le problème a été détecté.

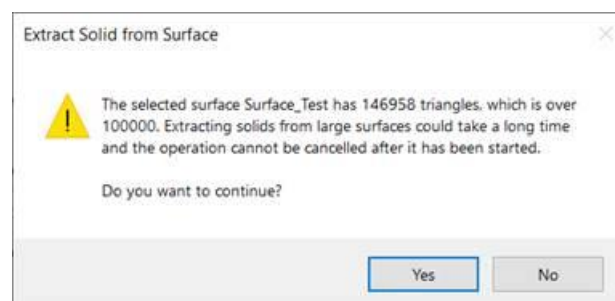


Figure 17 : Problème avec l'extraction de solides depuis de surfaces composées de plus de 100K triangles

Il fallait donc découper la surface de terrain avant de pouvoir lancer la création de solides :

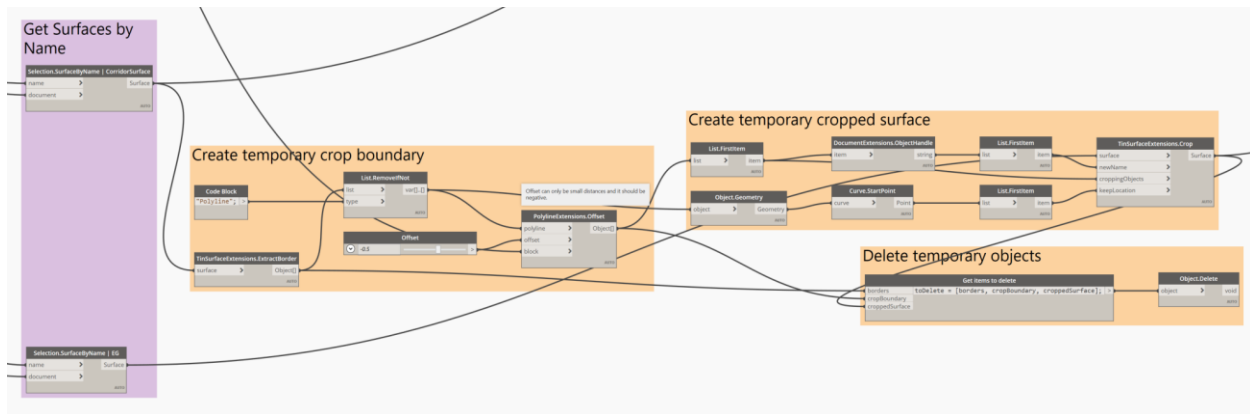


Figure 18 : Création d'une surface découpée

Avec l'utilisation de la surface découpée, les solides étaient bien créés :

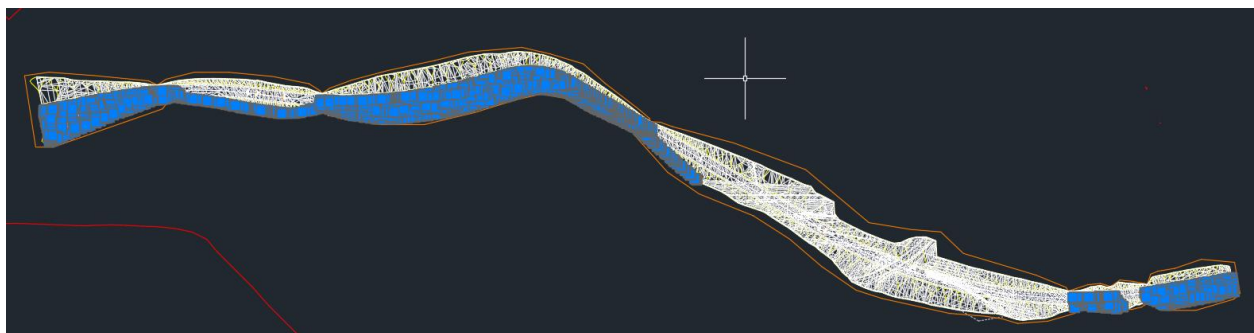


Figure 19 : Solides créés dans le modèle fourni par le client

En ce qui concerne le découpage des solides par corps de terrassement, la réalisation de l'automatisation demandait un grand effort (au-delà des 4 jours prévus). Le choix a été fait, pour l'instant, d'accepter la manipulation manuelle pour réaliser une étape (la création des polygones pour montrer la limite de chaque corps de terrassement et donc l'endroit où le solide sera découpé). Cependant, le process pour diviser ou couper les solides a été automatisé.

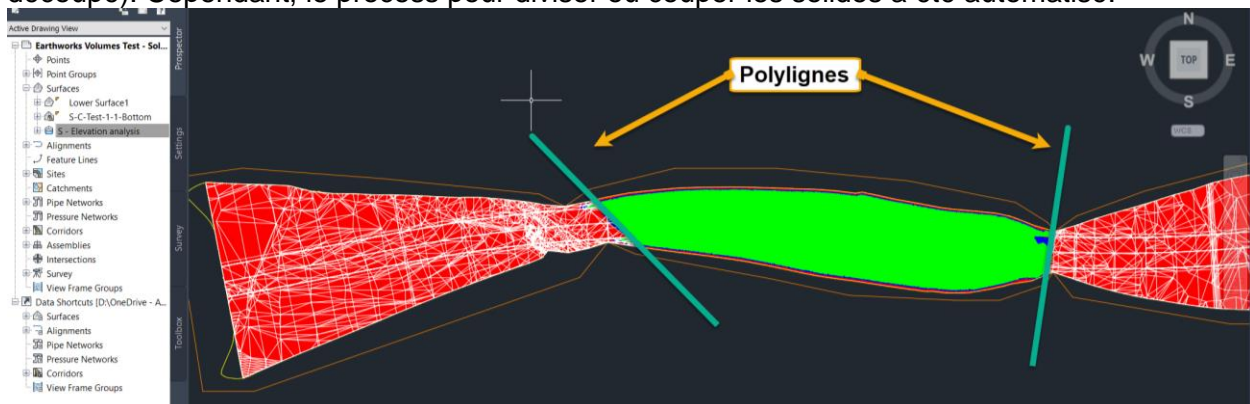


Figure 20 : utilisation des polygones créés manuellement pour déterminer le plan où le solide serait coupé

Figure 21 : Utilisation des dictionnaires pour avoir des choix paramétriques

Finalement, voici le résultat du graphe modularisé en trois graphes différents :

- Créer des solides

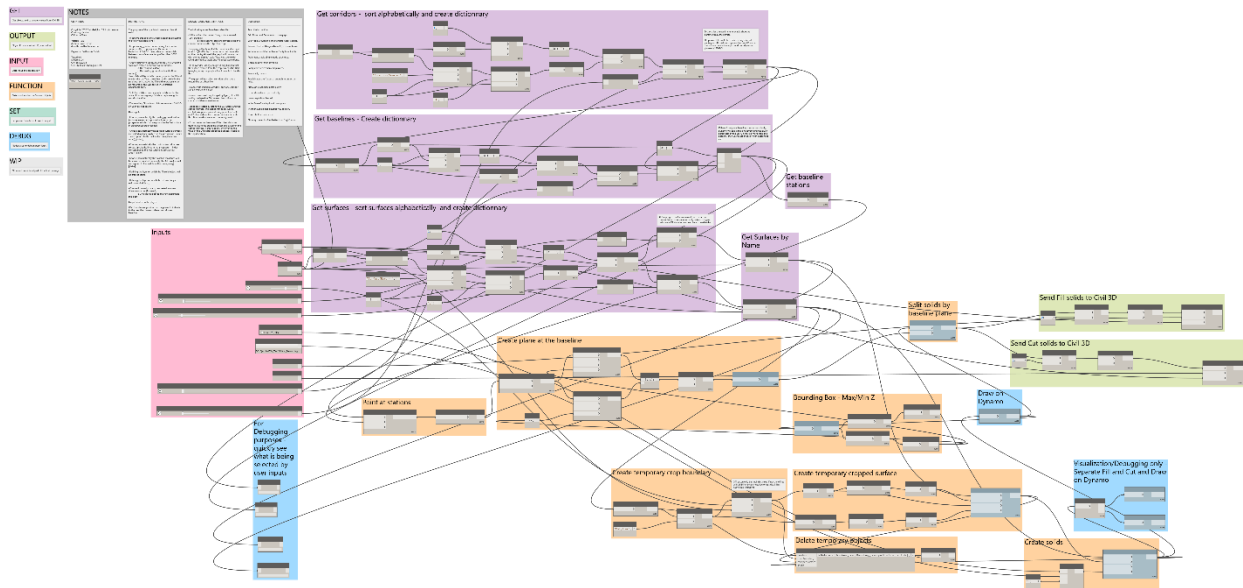


Figure 22 : Graphe pour créer des solides représentant le déblai et remblai d'une route

- Manipuler les solides

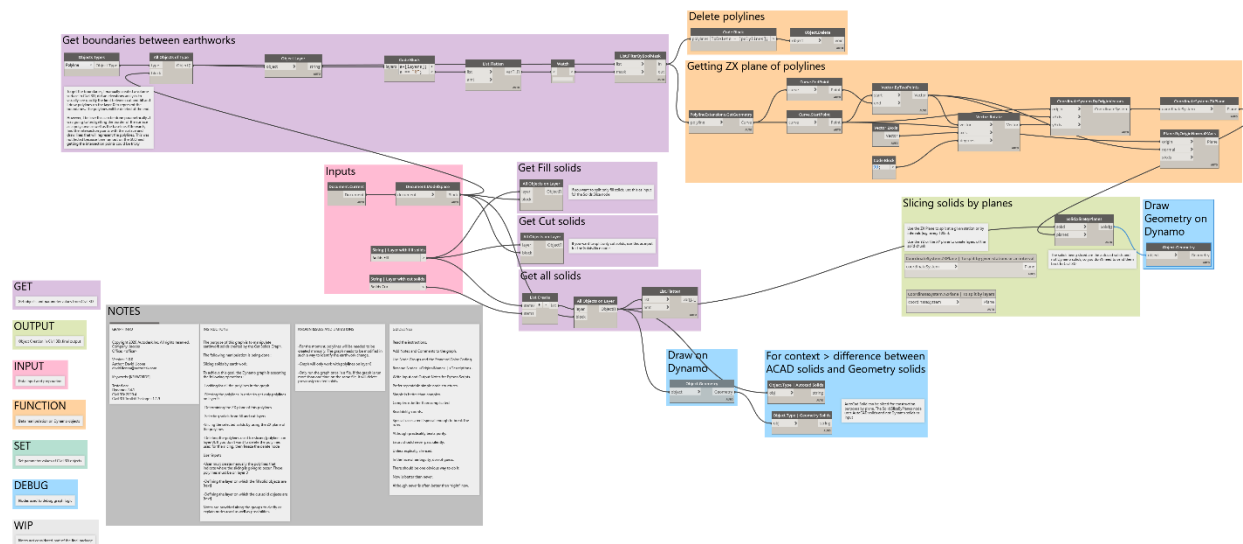


Figure 23 : Graphe pour découper les solides par corps de terrassement

- Attribuer des jeux de propriétés

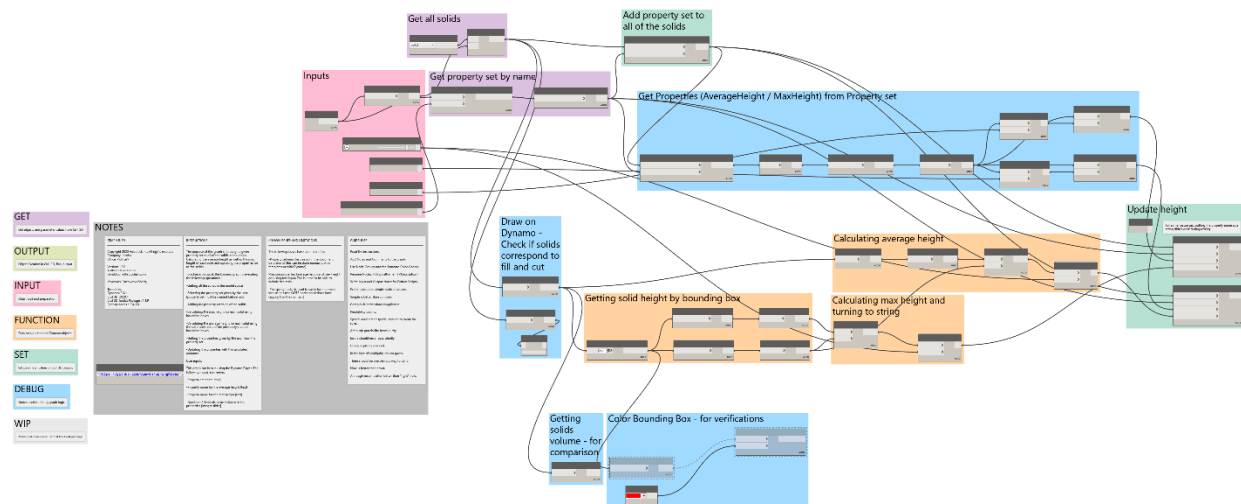


Figure 24 : Graphe pour attribuer des jeux de propriétés et calculer les propriétés des solides

Implémentation / Adoption

Pour mettre en œuvre un changement dans une organisation, les aspects suivants doivent être pris en compte :

- **Personnes** : il pourrait être nécessaire d'embaucher de nouvelles personnes possédant les bonnes compétences ou de former les parties prenantes actuelles ;
- **Organisation** : la nouvelle approche peut nécessiter la création de nouveaux rôles et responsabilités, la définition d'un plan de déploiement de la nouvelle stratégie et la gestion de la communication avec les parties prenantes concernées ;
- **Processus** : la bonne coordination des rôles impliqués dans un ensemble de tâches et de procédures qui constitue le cœur de la solution métier ;
- **Technologies de l'information** : les outils d'information qui permettent d'utiliser la solution.

Les sections suivantes vont montrer des meilleures pratiques pour que ce changement soit accepté.

Convention de nommage des fichiers

Des conventions de nommage pour les différents fichiers sont appliquées à la plupart, sinon à tous, les fichiers au sein des organisations d'ingénierie. Les fichiers Dynamo ne doivent pas échapper à cette règle. Les conventions de nommage clarifient le contenu, permettent de structurer la gestion du contenu et rendent possible la recherche de solutions.

Un exemple peut être le suivant :

- Logiciel_Métier_UsageBIM_But_Format_Version
- « **C3D_VRD_Documentation_CréationProfilsTravers_DYN_V1..dyn** »

Gabarit de référence

Un gabarit de référence doit exister pour l'organisation et doit être appliqué dans chaque graphe. Il est conseillé de prendre avantage des couleurs dans Dynamo pour introduire un codage de couleur. Le gabarit doit aussi inclure les informations minimales à fournir dans un graphe Dynamo.

Voici le gabarit utilisé chez Autodesk et qui peut servir d'exemple :

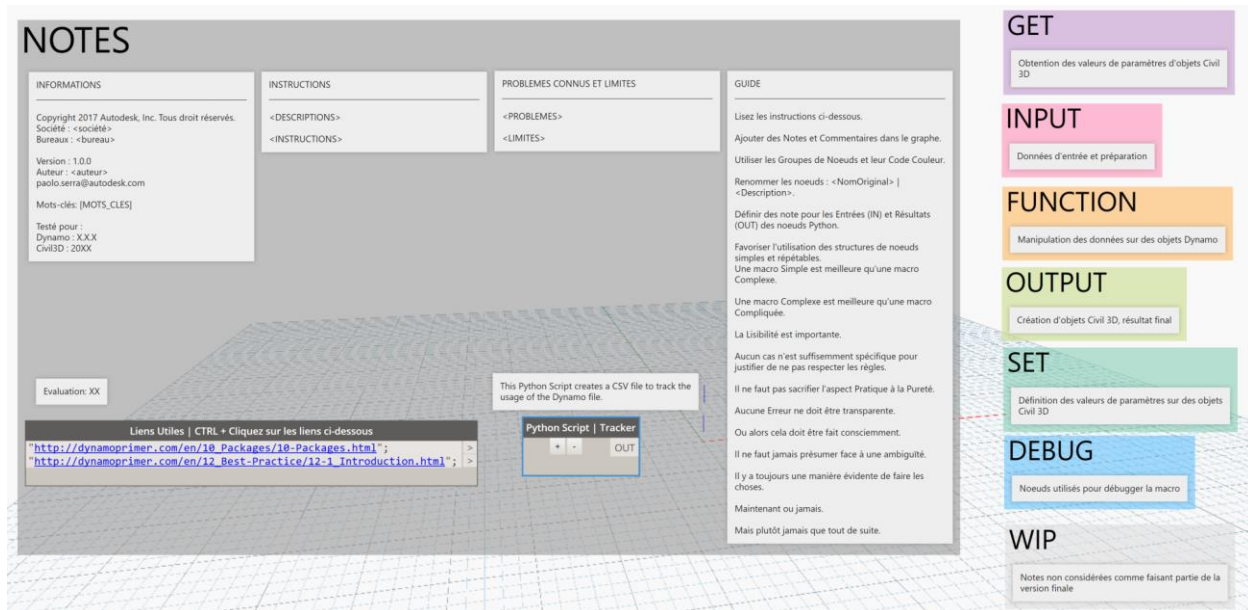


Figure 25 : Exemple du gabarit de référence à Dynamo

Stockage des fichiers

Les fichiers Dynamo doivent être centralisés dans une structure de dossiers par discipline et / ou fonctionnalité. Les fichiers sources doivent être séparés des fichiers qui seront utilisés dans des projets. Lorsque des solutions Dynamo sont demandées, les solutions doivent être copiées depuis la bibliothèque globale vers un dossier de projet afin qu'il soit possible de créer une relation entre un fichier Dynamo et un modèle Civil 3D. Cela permettra aussi de gérer les mises à jour du modèle et de s'adapter à une configuration de projet spécifique.

Les nouvelles solutions Dynamo doivent être développées dans un environnement distinct des solutions approuvées (cela peut être aussi simple qu'un dossier accessible uniquement par les développeurs de solution).

Testing et validation

Lors du développement d'un graphe Dynamo, il est important de valider la solution avec l'utilisateur final et d'intégrer tout retour d'information pertinent dans la solution. Il faut valider la solution Dynamo en exécutant non seulement le graphe Dynamo mais également en lisant les instructions et la documentation incluses pour s'assurer qu'elles sont lisibles et faciles à comprendre.

Les utilisateurs finaux doivent être encouragés à documenter dans quelle mesure la solution Dynamo a fonctionné dans le contexte du projet. Ceci est crucial pour sensibiliser le reste de l'organisation au succès de l'automatisation, afin que les autres coordinateurs d'équipe et les utilisateurs finaux puissent bénéficier de l'expérience précédente.

Suivi de l'utilisation

La collecte d'informations sur l'utilisation de la solution Dynamo peut aider à mesurer les solutions que les équipes trouvent les plus avantageuses, ainsi que la valeur financière associée à la solution elle-même.

Les graphes Dynamo doivent contenir un script de suivi d'utilisation qui collecte des informations sur l'utilisation du graphe (temps, utilisateurs, nom de l'ordinateur, nom du graphe, emplacement du graphe et nom du modèle). Ainsi, il est nécessaire d'avoir des indicateurs de performance permettant de mesurer les gains et la valeur d'un graphe.

Communication

Une campagne effective de communication est la clé pour garantir la bonne adoption de l'automatisation au sein des équipes.

Il faut communiquer sur les réussites obtenues. Pour chaque solution, il est donc idéal de demander des commentaires/remarques aux utilisateurs finaux afin de quantifier les avantages associés à l'automatisation. Par exemple, combien de temps a gagné l'utilisateur final, grâce à la solution ? Ces éléments peuvent être utilisés lors des campagnes de communication.

Il est aussi très important de se baser sur le témoignage des utilisateurs Dynamo. La communication sera mieux prise en compte si elle est transversale et pas verticale (c'est-à-dire ne vient pas que par une voie hiérarchique).

En termes d'outils, la vidéo est un très bon moyen pour faire la communication. C'est dynamique et visuel et donc plus attirant qu'une newsletter par exemple. Attention au storytelling, il faut bien avoir l'audience en tête et donc bien expliquer la problématique et décortiquer la solution tout en mettant en évidence la valeur gagnée.



Figure 26 : Se focaliser sur le storytelling (source : [Rawpixel](#))

Groupe d'utilisateurs

Essayez de former un collectif de super utilisateurs de Dynamo qui pourront s'appuyer les uns sur les autres et surtout conduire le changement des pratiques nécessaire pour implémenter l'automatisation au sein des équipes.

Ressources

Il est important d'être conscient que le développement des graphes Dynamo peut requérir des expertises qui ne soient pas présents aujourd'hui au sein d'une organisation. Pour garantir le succès, il faut donc :

- Former du personnel en interne ou ;
- Investir dans l'acquisition du talent.

Pour les premières solutions, une bonne idée est de se faire accompagner par un expert qui pourra aiguiller le développement et le déploiement.

En conclusion

- Appliquer des meilleurs pratiques comment : Utiliser des gabarits avec des codes couleurs, mesurer l'utilisation des graphes, suivre une convention de nommage ;
- Communiquer en montrant la valeur gagnée, s'appuyer sur le terrain pour communiquer ;
- Mettre en place une politique de formation ou d'acquisition du talent.

Références

Pour apprendre à utiliser Dynamo :

- [Primer](#) ;
- [Accesorize your Design – Dynamo for Civil 3D](#) | **Stacey Morkin et Dylan Kahle**

Pour s'améliorer dans la maitrise de Dynamo :

- [Dynamo Forum – Civil 3D](#) ;
- [Computational Design for Civil Engineers](#) | **Paolo Serra et Safi Hage**
- [Dynamo in Civil 3D Introduction—Unlocking the Mystery of Scripting](#) | **Jowenn Lua et Andrew Milford** ;
- [Generating, Transforming, and Analyzing Railway Design Data in Civil 3D and Dynamo](#) | **Wouter Bulens et Steve Crokaert**.

Cours AU 2020 (pas de lien disponible) :

- Generative Design | **Raquel Bascones et Paolo Serra** ;
- Civil 3D Toolkit | **Jowenn Lua**.

Remerciements

J'aimerais remercier plusieurs personnes qui m'ont appuyé pour la réalisation de ce cours : **Philippe BONNEAU** pour ses précieux conseils et mentoring, **Emmanuel LAGARDETTE** pour son encouragement à participer cette année à AU, **Jowenn LUA** et **Raquel BASCONES** pour le partage de leurs expériences, **Paolo SERRA** à qui je fais appel souvent quand j'ai des questions sur l'automatisation et Dynamo pour Civil 3D, **Hervé MORELLON** pour m'avoir aidé dans la session de Questions/Réponses et **Lucie MAGAUD** et **Margaux GRANDON** pour m'avoir donné du feedback.