



Auteur(s) : Arthur Villarroya-Palau

Versionnement : outils d'étude sur l'évolution du code,
des connaissances et des données

Sommaire

- Introduction
- Enjeux
- Versionnement du code
- Versionnement de la documentation
- Versionnement des données
- Table de correspondance

Introduction

Contexte :

- Ville intelligente
- Production et utilisation de beaucoup de données hétérogènes et complexes
- Une multitudes de projets impliqués et de connaissance générées

Problématique :

Comment concevoir une méthode capable de **suivre l'évolution urbaine** sans dissocier les **données**, le **code**, les **connaissances** produites, les décisions et les contextes dans lesquels ces connaissances ont été générées ?

Enjeux

accès-aux-articles-scientifiques

science-ouverte

normes

réplicabilité

reproductibilité

données-ouvertes

1. Code

Plusieurs outils bien connus :



Git



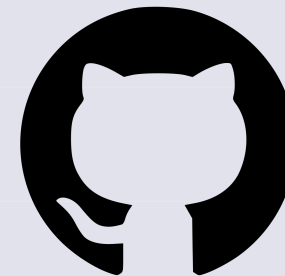
SVN



Mercurial



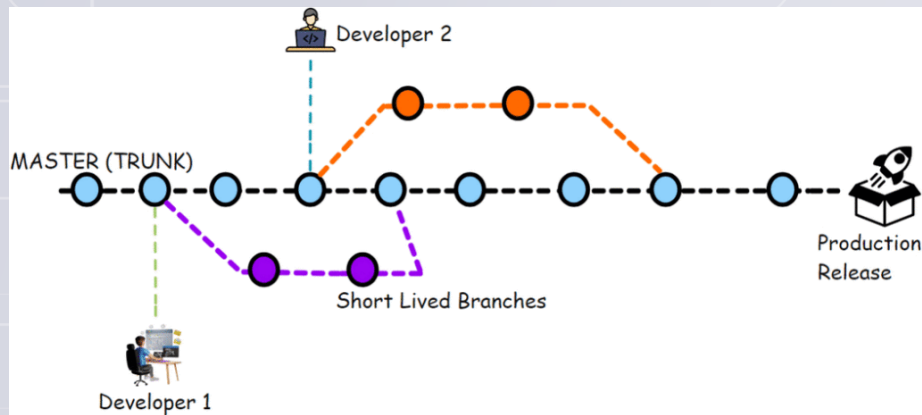
Gitlab



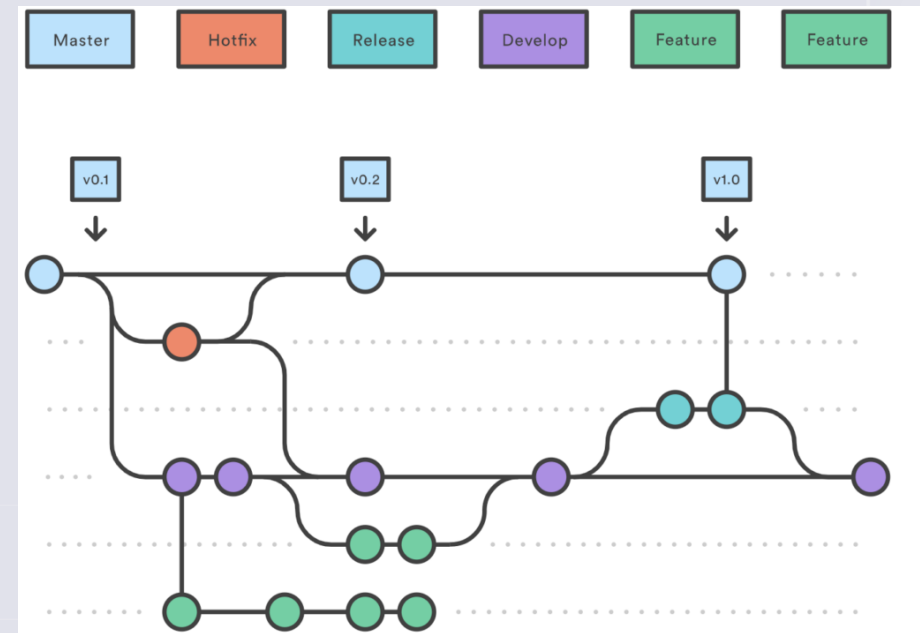
Github

1. Code

Modèles de branchement :



Trunk-Based Developement



Git Flow

1. Code



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Archive mondiale du code source avec comme objectif de préserver le patrimoine logiciel.

Un identifiant unique pour chaque code versionner.

Fonctionne comme un DOI (Digital Object Identifier) mais pour le code.

Geler une version de code dans le temps, pour garantir la répliquabilité.

2. Documentation

Type de documentation :

- **Technique (code)** : Description de fonction, de classes
- **Fonctionnelle / Méta** : Objectif, choix de conception, contexte
- **Utilisateur** : installation, configuration, utilisation

2. Documentation

Stratégie de versionnement	Description
Dépôt Git parallèle	Documentation est versionnée comme le code
GitHub Pages	Héberge un site web à partir d'un dépôt Git
Générateurs de sites statiques (MkDocs, Docusaurus, Read the Docs) + git ou plug-in	Outils transformant une documentation en site web statiques. Permettant de naviguer entre les versions

3. Données

Plusieurs types de données :

- **Données attributaires (ou tabulaires)** : tableaux de mesures ou bases de données
- **Données textuelles** : descriptions, rapports ou extraits de documents
- **Données vectorielles en 2D ou 3D** : géométries précises (points, lignes, polygones, volumes), cartographie ou modélisation urbaine
- **Données raster 2D ou 3D** : grilles de pixels, voxels pour des images, cartes ou champs de valeurs (température, élévation, etc.)
- **Données image** : photographies, plans scannés, captures satellite ou vues d'objets

3. Données

Plusieurs formats de données:

Type de format	Exemples	Usages principaux
Tabulaire	CSV, TSV, Excel	Données classiques, statistiques
Semi-structuré	JSON, XML, YAML	Données API, logs, configurations
Scientifique	netCDF, HDF5, GeoTIFF	Climat, environnement, imagerie
3D sémantique	CityGML, IFC, BIM	Modélisation urbaine, architecture
Nuages de points	LAS, LAZ	Relevés 3D, LiDAR, photogrammétrie
SIG vectoriel	Shapefile, GeoPackage	Cartographie, données géospatiales
Graphe de connaissance	RDF, RDF Quad, TTL	Ontologies, données liées, SPARQL
Informel / créatif	PDF, image, prototype physique	Création, documentation, art/design

3. Données

Outils de versionnement de données :

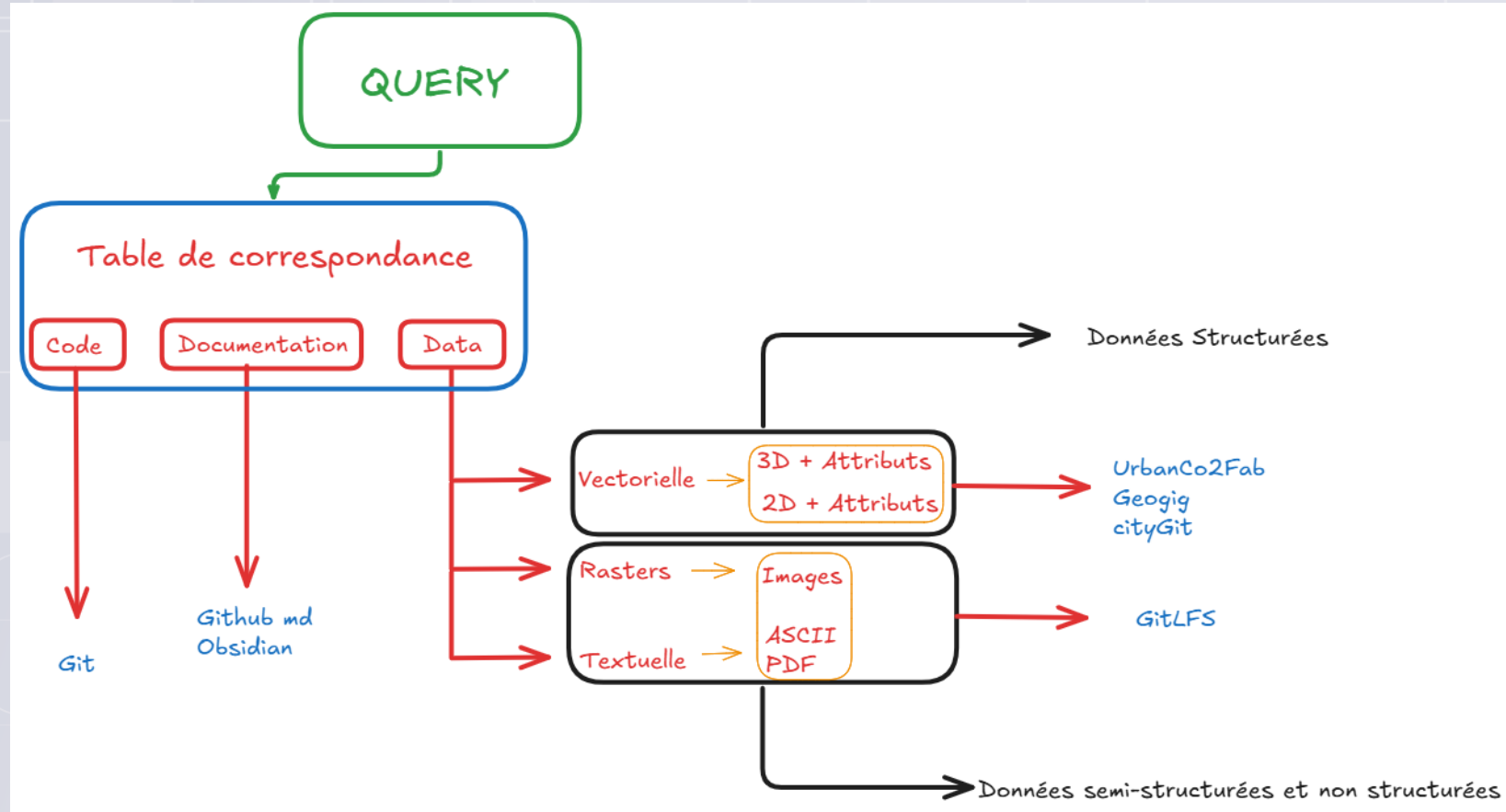
- **Git** : pour versionner des petits fichiers texte (CSV, JSON)
- **Git LFS** : étend Git pour gérer de gros fichiers (images, vidéos)
- **DVC** : versionne les datasets et les modèles, tout en gardant le lien avec le code
- **LakeFS** : pour les data lakes (S3)
- **Dolt / DoltHub** : bases SQL : permet de faire des diff, branches et merges comme avec Git, mais dans des données relationnelles.
- **GeoGig** : versionne les objets spatiaux (points, polygones) dans PostGIS ou Shapefile
- **Kart** : plus moderne que GeoGig, avec interface CLI et QGIS. Gère aussi les raster, et propose des diffs sémantiques.

3. Données


Outils issus de la recherche :

- **URBANCO2FAB** : applique Git à la ville en 3D : chaque changement (bâtiment, volume) devient un commit.
- **VC for Point Clouds** : versionne les nuages de points en ne stockant que les différences.
- **Quit Store** : Versionne des graphes RDF à un niveau très fin.
- **ConVer-G** : Va plus loin : permet d'interroger plusieurs versions de graphe RDF en parallèle.

4. Table de correspondance



4. Table de correspondance



```
{
  "correspondances": [
    {
      "id": "T1",
      "date": "2024-11-01",
      "description": "Version initiale du projet",
      "donnees": "v1.3",
      "code": "v3.0",
      "documentation": "v1.2",
      "metadonnees": "v1.2",
      "resultat": "modele_3D_ville_v1"
    },
    {
      "id": "T2",
      "date": "2024-12-15",
      "description": "Mise à jour des données urbaines, ajustement du code",
      "donnees": "v1.4",
      "code": "v3.1",
      "documentation": "v1.3",
      "metadonnees": "v1.2",
      "resultat": "modele_3D_ville_v2"
    }
  ]
}
```

Liens utiles

Lien de la synthèse : <https://github.com/VCityTeam/UD-IArbre-Research/tree/master/deliverables/T2.6>

Lien pour les propositions : <https://github.com/VCityTeam/UD-IArbre-Research/issues>



Merci pour votre attention ! Des questions ?