en_US.png ...english version of this page

Computree - Principes de fonctionnement

Computree est une plateforme de traitement de données issues de télédétection aéroportée (voire satellitaire) ou terrestre. La plateforme en elle-même propose un Framework de traitement de données fondé sur une logique de chaînes de traitement séquentielles (même si une chaîne peut avoir des ramifications).

Ainsi une chaîne de traitement est définie par une séquence d'*étapes* algorithmiques. Chaque étape :

- Définit quelles données lui sont nécessaires en entrée sous la forme de **modèles de données d'entrée**. Certaines étapes n'ont pas besoin de données d'entrée (étapes de chargement).
- Propose éventuellement un certain nombre de **paramètres** permettant de régler certains aspects de l'algorithme, ou de spécifier des chemins de fichiers ou dossiers d'entrées / sorties.
- Définit quelles données elle va générer en sortie sous la forme de **modèles de données de sortie**.
- Implémente l'algorithme en lui-même. Cet algorithme récupère les données d'entrée via les **modèles de données d'entrée**, exécute un traitement en tenant compte des **paramètres**, et produit des données de sorties qu'elle expose à l'aide des **modèles de données de sortie**.

Il est ainsi possible de constituer l'ensemble de la chaîne de traitement sans l'exécuter, dans la mesure où les étapes peuvent interagir via les modèles de données, sans nécessité d'exécution de l'algorithme en lui-même. Cela permet de préparer à l'avance des traitements longs, ou de les exécuter en mode batch.

De plus le système des modèles de données permet à des étapes issues de plugins développés totalement indépendamment d'être utilisées ensembles dans une même chaîne de traitement.

De plus, Computree propose plusieurs alternatives aux « étapes standards » afin de simplifier certaines tâches :

- Les **Actions** permettent un traitement faisant appel à des interventions interactives de l'utilisateur. Elles peuvent être directement mises à dispositions (mode boite à outils), ou être utilisées dans les étapes, entraînant dans ce cas une pause de la chaîne de traitement pour permettre des actions de l'utilisateur. Les étapes utilisant des actions nécessitent l'utilisation de l'interface graphique (la partie interactive est désactivée en mode batch).
- Les **Readers** sont des classes permettant de charger un format de fichier donné. Leur intérêt est de pouvoir dissocier dans le temps le choix du fichier de son chargement effectif. De plus, cela permet d'utiliser le format de lecture soit en tant qu'étape de chargement, soit au sein d'un étape de traitement.
- Les **Exporters** sont, symétriquement aux readers, des classes permettant d'exporter un format de fichier donné. Leur intérêt est de permettre via une même classe, d'une part des exports dans les chaînes de traitement, ou d'autre part des exports en mode interactif sur la base d'une sélection utilisateur. De plus, cela permet d'utiliser les formats d'écriture au sein d'étapes de traitement.
- Les **Filtres** sont des versions pré-paramétrées d'étapes. Chaque famille de filtre fixe les modèles de données d'entrée et de sortie. De plus il s'agit forcément d'étapes proposant un filtrage de certains types d'éléments, ce qui permet à la classe fille de ne définir que le contenu du filtre en lui-même ce qui en simplifie la création, et évite les redondances.
- Les **Métriques** sont des versions pré-paramétrées d'étapes. Chaque famille de métrique fixe les modèles de données d'entrée et de sortie. De plus il s'agit forcément d'étapes proposant un calcul d'indicateur à partir de certains types d'éléments pré-définis, ce qui permet à la classe fille de ne définir que le contenu du calcul en lui-même ce qui en simplifie la création, et évite les redondances.

Structures de données

Dans Computree les données sont organisées selon le formalisme suivant.
Les données en elles-mêmes sont stockées dans des **Items**, qui sont définis dans le cœur de la plateforme. La logique est qu'un item soit la plus petite entité possible, dans le but de la rendre la plus générique possible. Ainsi un item peut être : une forme géométrique 3D élémentaire (cercle, sphère, parallélépipède rectangle, cône, cylindre,…), une forme géométrique 2D élémentaire (cercle, rectangle, polygone,…), un raster, une grille voxel 3D, un nuage de point, un maillage,…

Pour les points d'un nuage ou les faces d'un maillage, l'élément unitaire stocké dans l'item est le nuage de points ou le maillage et non le point ou la face pour des raisons d'efficacité dans les traitements et de stockage en mémoire.
L'organisation des items en structures plus complexes est faite via des **Groupes**, dont le rôle est de construire toute structure de

données nécessaire dans une étape. Ainsi un groupe peut contenir, d'une part des items, d'autres part des listes de groupes. Ainsi par composition hiérarchique il est possible de générer des structures de données complexes.

Dans ces structures de données, chaque niveau de groupe ou d'item est associé à un modèle de donnée (modèles de données de sortie décrits avant, qui permettra de l'indexer et d'y accéder.

Le premier niveau de groupe d'une structure de données est placé dans un **Résultat**. Une étape peut produire de 0 à n résultats.

Tout item doit être placé directement dans un groupe. Tout groupe est placé dans une liste correspondant à son modèle de donnée, soit située dans un groupe de niveau supérieur, soit à la racine du résultat.

Les items (exemple : un cylindre) contiennent un certain nombre d'attributs par défaut (exemple : centre, direction, rayon, longueur). Il est possible d'ajouter des attributs supplémentaires aux items, dans les étapes (exemple : erreur d'ajustement, type de cylindre). On ne peut pas ajouter d'attributs aux groupes.

Lorsqu'une étape a vocation à compléter un résultat d'une étape précédente (exemple : ajuster des cylindres pour chaque cluster de points), elle va faire une copie du résultat d'entrée, la compléter, et la fournir en résultat de sortie. La copie n'est pas totale, pour optimiser l'utilisation de la mémoire : les groupes sont dupliqués (nouvelle structure, potentiellement modifiée), par contre les items préexistants sont simplement référencés (pointeurs) dans les nouveaux groupes. Cette approche permet de pouvoir exporter et visualiser toute résultat intermédiaire, sans qu'il n'ait été modifié par les étapes suivantes. On peut ainsi tracer, étape par étape, les résultats produits par la chaîne de traitement.

Architecture logicielle de la plateforme

La plateforme Computree est composée de différentes briques, dans une logique d'un design pattern Modèle Vue Contrôleur, même si dans sa version actuelle elle ne suit pas toujours ce design à la lettre pour des raisons d'efficacité (en particulier les contrôleurs ne sont pas toujours totalement isolés).

Ainsi la plateforme est composée :

- D'un cœur, sous la forme d'une série de libraires, définissant :
  - Les structures de données (résultats, groupes, items, attributs, gestion des nuages de points et de maillages),
  - Les classes de traitements (étapes, actions, readers, exporters, filtres, métriques),
  - La gestion de la chaîne de traitement (modèles de données, gestionnaire d'étapes, export/import de scripts, gestion de plugins).
- D'une interface graphique standard, adaptée à la création, l'utilisation et l'analyse détaillée (via de la visualisation 3D, 2D ou tabulaire) de chaînes de traitement complexes. Il est tout à fait possible de créer des interfaces alternatives utilisant le même cœur.
- D'une interface batch standard, permettant le lancement de scripts sans interface graphique.

En lui-même le cœur de la plateforme ne contient pas d'étape, d'action, de reader, d'exporter, de filtre ou de métrique. Ces éléments sont ajoutés via des librairies (pour les readers et exporters), ou des plugins (tous les éléments, plus éventuellement des items spécifiques à un plugin donné).

Les plugins sont totalement indépendants (un plugin ne connaît que le cœur et éventuellement des librairies). Les librairies quant à elles peuvent être utilisées dans les plugins. Chaque plugin / librairie va générer une dll qui sera dynamiquement reconnue et chargée au lancement d'une des interfaces.

Pourtant les éléments implémentés dans les plugins (et librairies dans certains cas) pourront être utilisés de concert par l'utilisateur, au sein d'une même chaine de traitement (tant que leurs modèles de données coïncident).

Maintenance et évolution de la plateforme

Le **Groupe Computree** fournit les ressources pour le maintien et l'évolution des éléments suivants :

- Libraires cœur
- Interface graphiques standard
- Interface batch standard
- Certaines librairies d'intérêt général (readers et exporters de formats courants par exemple)
- Certains plugins jugés d'intérêt général par les membres du Groupe Computree

Les plugins (ne figurant pas dans la liste de ceux maintenus par le Groupe Computree) sont développés indépendamment par les équipes / personnes qui le souhaitent. Pour les y aider, le Groupe Computree fournit la présente documentation en ligne, ainsi que des formations (web ou en présentiel). Sur demande, en fonction des ressources disponibles et dans la mesure où le plugin développé est mis à disposition de la communauté ou non, le Groupe Computree peut fournir un soutien supplémentaire aux développeurs de plugins, dont le contenu précis est à déterminer au cas par cas.

La Charte Computree décrit plus en détails la gouvernance de la plateforme. Son acceptation est un prérequis à l'utilisation de Computree.

Retour à l'index