General presentation of the platform, main principles

fr_FR.png ...version française de cette page

This section presents the philosophy and important concepts of **Computree**.

Philosophy

The goal of **Computree** is to allow the processing of 3D points clouds, in order to extract useful information, as for example:

- Indicators
- Simple geometrical forms (ex. : cylinders for detected tree trunks, in a forest plot)
- Geographical information (ex. : Digital Terrain Model as a raster)

The platform is design in a Research and Development approach. The main idea is for the user to create interactively an algorithmic sequence allowing from loaded data, to obtain wished results. This sequence, or more accurately this hierarchical tree (it is possible to have parallel algorithmic ramifications), is decomposed in unit algorithmic **steps**, managing specific processing.

Steps are brought together in **plugins**, what allow to easily enrich the platform with new processing.

After execution of the algorithmic sequence, transitional **results** of each unit **step** are available, exportable and displayable. It allow a deep analysis of processing quality.

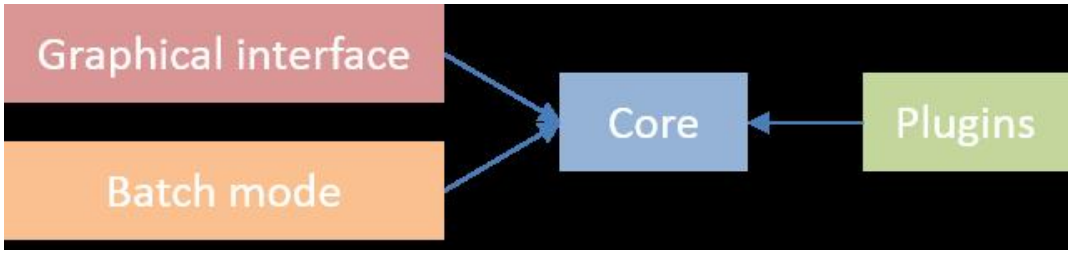
The platform has been developed with the following philosophy:

- **Modularity**: algorithms are brought together in independently developed plugins.
- **Openness**: the core of the system is under GPL/LGPL(http://www.gnu.org/) licence. Each plugin could have any licence, let to plugin developer choice.
- **Portability**: by the use of c++/Qt (http://qt-project.org) framework, the platform could be compiled under Windows (Xp and after), Mac OS X (Lion and after) and Linux (Ubuntu 12.04 and after. Please refer to Qt site for other distributions), in 32 or 64 bits.
- **Performance**: by the use of c++, and the multi-core processing capacities of Qt framework, the platform aims to optimize computing time.

The goal is so to allow to separate teams to collaborate in 3D forest scenes processing.

But even if the platform allows a Research collaborative approach, it is also designed to permit an easy transition of developped methods in a production context (in forest management for example). Indeed, the system allows an easy implementation of dedicated interface for specific needs, using Computree core and plugin transparently.

General software architecture



The Computree platform is composed of following elements:

- **A core**, which manage algorithmic steps, and offers generic data structures and functionality.
- **A standard graphical interface** (*ComputreeGUI*), destined to R&D / Research users. It allow to create and execute algorithmic

sequences, and to analyze their results.

- **A batch mode** allowing to use automated processing interfaces.
- **Plugins**, developed by team wanting to enrich Computree with new algorithmic steps (or other functionality).

Structure for data management and processing

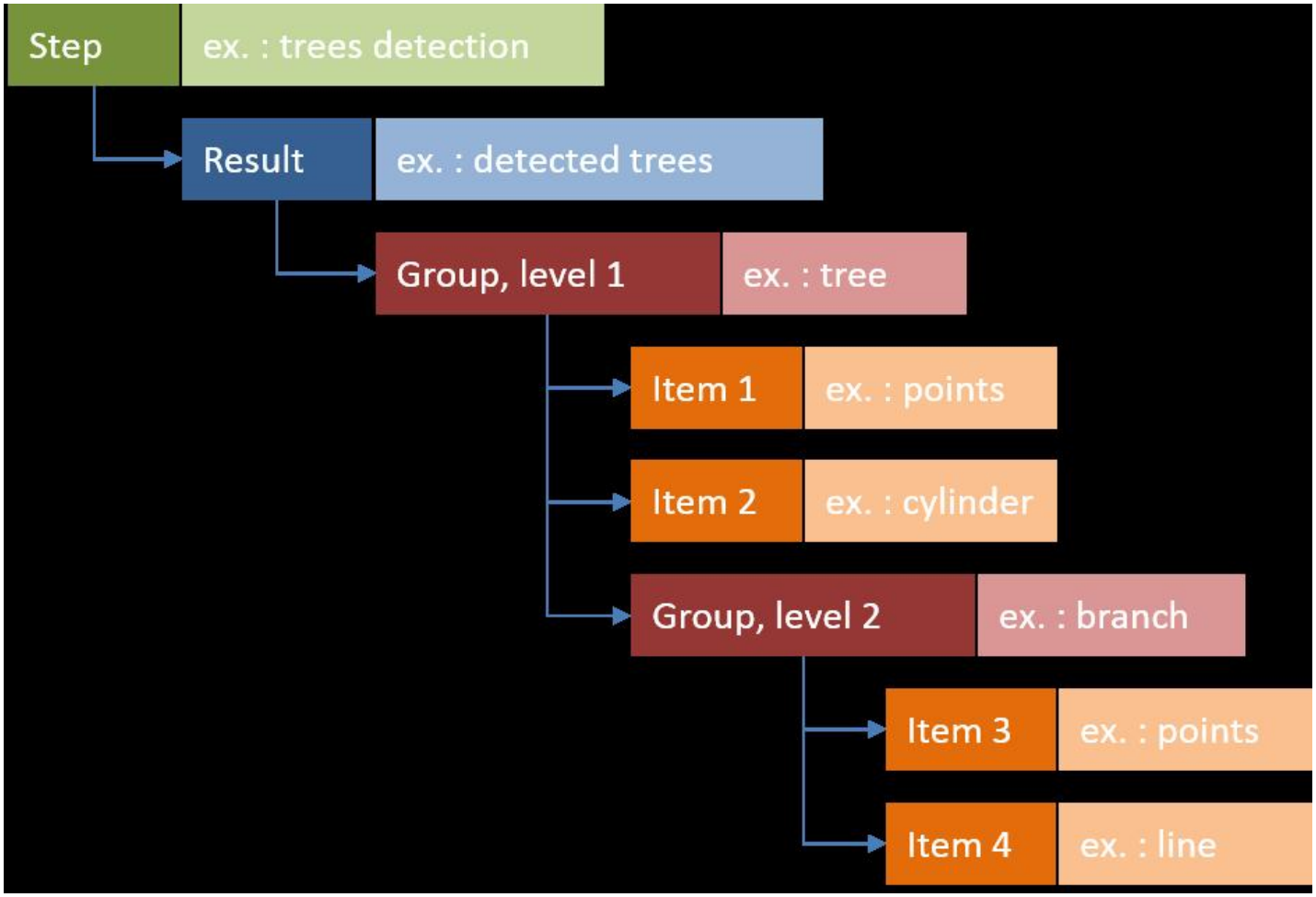
In order to process data, Computree use a **step tree**, exportable as a **script #□□ □□□□□□ □□□□□□□□□□ □□□**. The root of the step tree is a or a data creation step. After that, each step is attached to a mother step. The step tree can be completely constructed before execution (allowing batch or delayed processing).

Steps are implemented in #□□ □□□□□□□□**plugins**□□□. Each step take as input the results of its mother step, and generate other results as output.

Results store data for itself. In order to allow chaining independently developed steps, results store data in generic elements defined in the Computree core.

These elements are of two different kind:

- **Groups** allow to create a hierarchical data structure in the result
- **Items** are data for itself, stored in groups of different levels



*On this diagram, a **result** structure in represented, showing the structuring by **groups**, and data storage in **items** hosted in these groups. It should be noted that for each **group level**, the step can produce an undetermined number of instance for this structure. So if the algorithm has found n (forest) trees, the structure would be created n times, containing each time differents points and cylinders, and also a variable number of branches for each tree.*

In order for the **step tree** to be completely created before execution, it is necessary for the steps to communicate together, to determine their compatibility.

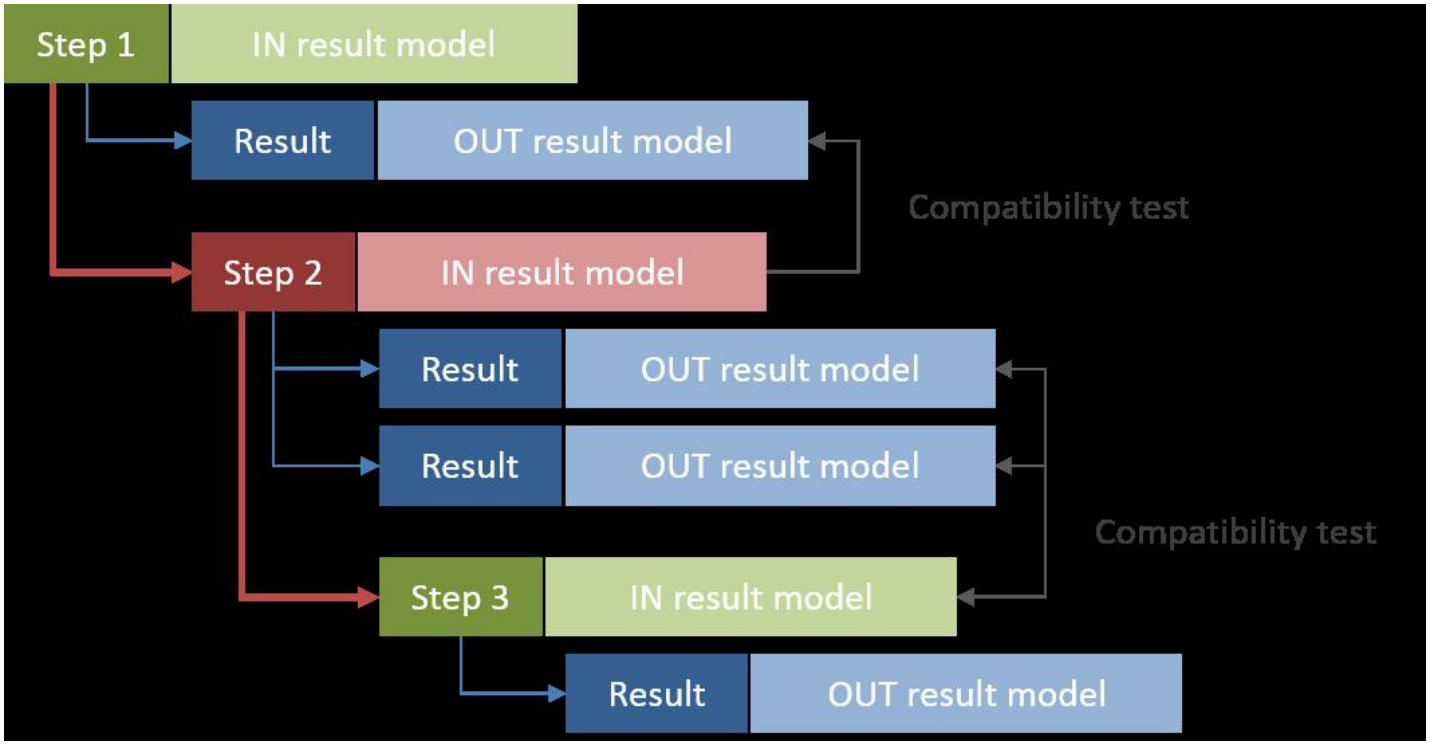
To do that, each step has its own **OUT result model** describing its structure:

- The hierarchical structure constituted by **groups**
- The **items** content, for each **group** level

These models are named “OUT”, because they describe the structure of **output** results of the step.

On the other side, steps have **IN result models**, which are like mandatory result structures needed as input for the step to process. These **IN result models** are compared to **OUT result models** of previous step, to decide if it could be used as input.

The models are named IN, because they describe structure needed as **input** by the step.



*On this diagram, steps 1 and 3 come from one plugin, whereas step 2 comes from another one. Red arrows show the succession of steps. Blue arrows shows the creation of results by the step. Black arrows represent the compatibility testing done by one step, thanks to its **in results models**, compared to **out results models** of previous step results, candidate as input.*

Plugins content

Plugins can extends **Computree** functionality, providing elements of different kind:

- **Step** : it is the most frequent plugin element. One step take one or mode results as input, and produce one or more results as output.
- **Items type** : even if it is not recommended (to allow compatibility between steps from different plugins), a plugin can define its own items types.
- **Reader** : it is a file reading format, creating specific kind of items.
- **Exporter** : it is a file writing format, allowing to export specific kind of items.
- **Action** : it is an interactive mode of data manipulation. An action could be used directly used in views, or be triggered by a **manual step**, so asking the user to interact with data.

Plugin Base, delivered with Computree core, provides generic steps, readers and exporters for classic 3D points clouds formats and also some actions. Please refer to project [[plugin-base:En_wiki|Plugin Base]] for more details.

Back to summary	Next Page (Items types)

Files

arbre_etapes_en.png	23.6 KB	05/03/2014	Piboule Alexandre
str_resultat_en.png	20.8 KB	05/03/2014	Piboule Alexandre