Project Owner	Malcolm Francis
Code maintainers	Max Fellows James Leitch Malcolm Francis

PLEASE READ THIS BEFORE YOU CONTINUE:

This is an open source document. Everybody is welcome to contribute. To allow us to be efficient we have only a few rules:

- All contributors use comments to ask questions or make substantive and structural suggestions. Always suggest solutions in your comments not problems.
- Only Maintainers can resolve comments.
- All contributors make edits in “suggesting mode” (even if you are a maintainer or editor, so the other maintainers or editors can cross-check and accept your contribution.)
- Write suggested edits directly into the text in ‘suggesting mode’ do NOT use comments to suggest edits
- Only Editors assigned to the document can approve edits
- Spelling will be UK English
- Referencing Style is Harvard (author-date) type. Tools such as Zotero may be useful.

More info about how the moja global community works on projects can be found here.

Extending the FLINT framework to report on Pool Cohorts

May 31, 2021

Abstract

The Full Lands Integration Tool (FLINT) has been developed to support the GHG inventories development and the implementation of mitigation actions in the AFOLU sector. But, some functions and features of the FLINT need to be refined or added in order to fully support the effective implementation of the enhanced transparency framework of the Paris Agreement by developing countries, in particular, with regard to GHG inventories, mitigation actions and establishment and maintenance of sustainable GHG inventory management systems.

The UNFCCC secretariat has initiated a collaboration with Moja Global to refine the FLINT and make it available to developing countries across all regions.

As an addition to the Cohort framework addition, extend the FLINT framework to add Cohort information to the carbon movement recording, and extend various Modules written to output carbon movement to report based on Cohorts as required.

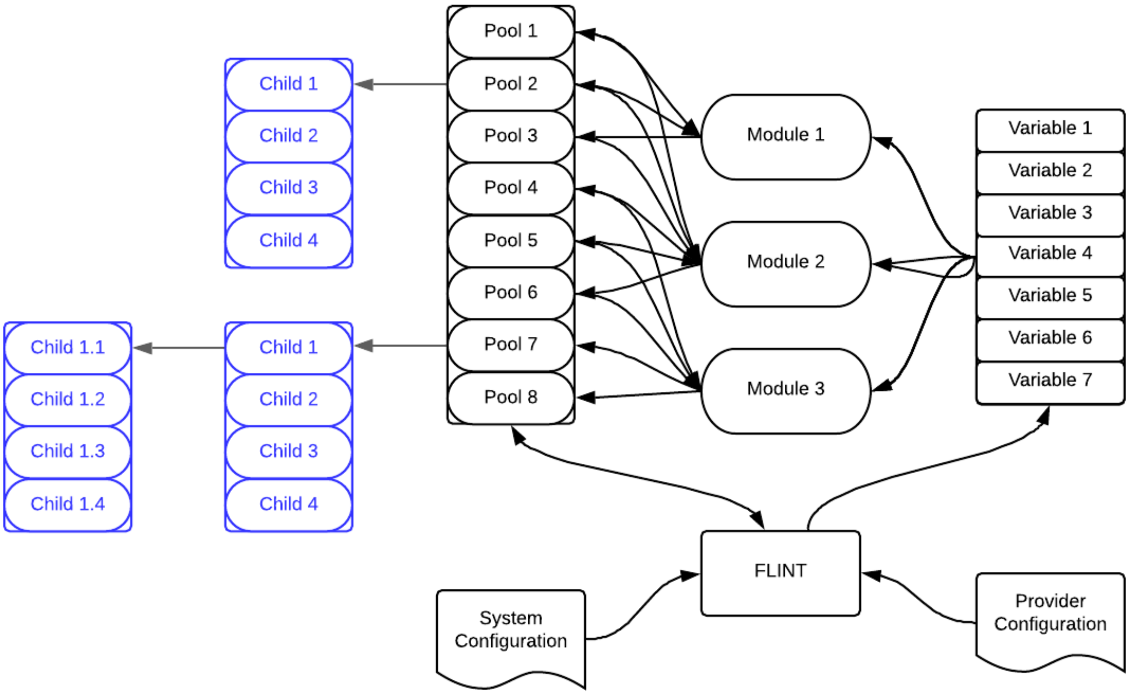
Introduction

This document's purpose is to detail changes required in the Full Lands Integration Tool (FLINT) framework to report on Pool Cohorts (or Nested Pools) - in both Stock and Fluxes. Cohorts are the concept of a Parent pool with Children pools (including rules on how these child pool values would be propagated). In functional terms, cohorts to a biomass pool that is present for a simulation unit (pixel) that is divided into sub-pools each with distinct properties. For example, a standard pool is dead organic matter (DOM). Through cohorts, it will be possible to model the DOM produced from one forest type separate from the DOM produced by a previous forest type, for example. The results can then be aggregated and reported out just for DOM.

The FLINT framework is designed to run Carbon Simulations both at single point (non-spatial) and at large scale (i.e. National level Spatial simulations). The results from these simulations are generally carbon values - both Stock and Movement (fluxes). The construct of the pools was initially simplified to ensure the core framework was stable with the reduced complexity of cohorts.

In conjunction with the FLINT enhancements documented in “**Extending the FLINT framework to support Pool Cohorts**” - adding the ability to identify Fluxes and report on Carbon Pools by a dynamic list of Cohorts (or Nested Pools) will greatly improve the use and value of the results generated.

Fig 1: Proposed Cohort System



Functional Requirements

- Framework extensions will need to be backward compatible, i.e. allowing existing systems to continue to work as intended.
- Allow both Stock and Flux reporting to identify Cohorts
- Allow Reporting of Aggregated Cohorts
- Allow reporting of Child Pools without aggregation

Non-Functional Requirements

- All Pool names will be unique
- Pool names will not contain Non-alphanumeric characters
- Pool names can be output in DOT notation

Additional Information

Dot Notation

When output in full format, nested Pool names will use dot notation to identify levels of hierarchy. Using "." in Pool name strings to identify levels. Some examples:

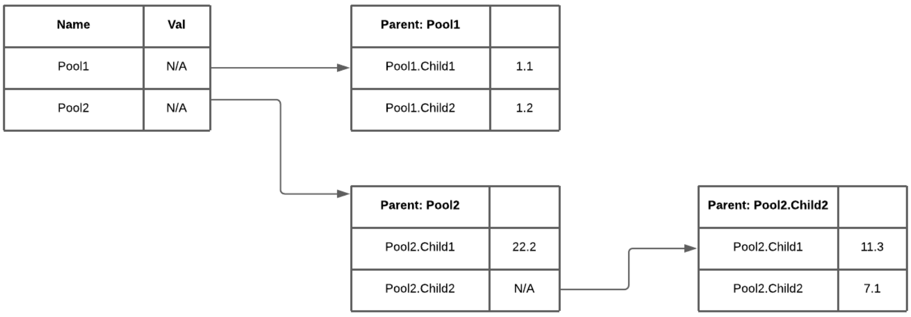
```
debrisCM                        # single Pool
debrisCM.eucalyptus             # referencing Child Pool
debrisCM.eucalyptus.BlueGum     # referencing Child of Child Pool
debrisCM.eucalyptus.SpottedGum  # referencing Child of Child Pool

aboveGroundCM
aboveGroundCM.Stems
aboveGroundCM.Branches
```

Examples of Nested Pool Value calculations

Below in the figures are some examples of how Pool values would be calculated.

Fig 4: Pool aggregation example



Name	Val
Pool1	N/A
Pool2	N/A

Parent: Pool1	
Pool1.Child1	1.1
Pool1.Child2	1.2

Parent: Pool2	
Pool2.Child1	22.2
Pool2.Child2	N/A

Parent: Pool2.Child2	
Pool2.Child1	11.3
Pool2.Child2	7.1

Pool name	Result	
Pool1	2.3	Sum of (1.1 + 1.2)
Pool2	40.6	Sum of (22.2 + (11.3 + 7.1))
Pool2.Child1	22.2	Sum of (22.2)
Pool2.Child2	18.4	Sum of (11.3 + 7.1)

Matrix of valid Pool transfers

Warning: Source and Sink Pools cannot be the same

	Sink Pool type			
Source Pool Type	**Pool**	**Parent Pool**	**Child Pool** **(Parent A)**	**Child Pool** **(Parent B)**
Pool	Yes	No	Yes	Yes
Parent Pool	No	No	No	No
Child Pool **(Parent A)**	Yes	No	Yes	Yes

NOTE: The system will throw an Exception when an invalid transfer is made.

FLINT Extension Summary

To handle reporting on Nested Pools, some additions will be required, these include:

- Framework extensions as described in the document "***Extending the FLINT framework to support Pool Cohorts***"
  - Specifically the extensions to the IPool interface, allowing identification of pools that are parents/children, aggregation of child pools, and access to the list of child pools.
- FLINT output module extensions
  - Module: *moja::flint::OutputerStream*
    - Configuration additions
    - Output of Cohort details as specified in this document
  - Module: *moja::flint::OutputerStreamFlux*
    - Configuration additions
    - Output of Cohort details as specified in this document

Framework extensions

As originally described in the document “***Extending the FLINT framework to support Pool Cohorts***”.

The current proposed extensions to the IPool interface is shown here:

```
class FLINT_API IPool {
  public:
   IPool() = default;
   virtual ~IPool() = default;

   virtual const std::string& name() const = 0;
   virtual const std::string& description() const = 0;
   virtual const std::string& units() const = 0;
   virtual double scale() const = 0;
   virtual int order() const = 0;
   virtual double initValue() const = 0;

   virtual int idx() const = 0;

   // Additions for parent pool                              <<---- Additions here ---->>
   virtual const IPool* parent() const = 0;
   virtual const std::vector<const IPool*>& children() const = 0;

   virtual double value() const = 0;
   virtual void set_value(double value) = 0;
   virtual void init() = 0;

   virtual const PoolMetaData& metadata() { return _metadata; };
  protected:
   PoolMetaData _metadata;
};
```

The methods `parent()` and `children()` can be used to determine the status of a Pool when reporting Cohort information.

The method `value()` will also be updated to calculate aggregated values - if the Pool is a parent. The result returned will be the sum of all children (and their children, etc).

Using the interface above, when reporting both Flux and Stock, a Pool can be recognised as a Parent and/or Child Pool if required.

State	Condition
Am I nested (a child of another Pool)?	`parent() != nullptr`
Am I a parent (have children)?	`children().len() > 0`

FLINT output module extensions

The handling of the Reporting for Nested Pools will be added to the Configuration JSON for the Modules that are used to output both Stock and Flux values:

- *moja::flint::OutputerStream*
- *moja::flint::OutputerStreamFlux*

The extensions will allow the user to adjust various settings of Cohort reporting. Settings for these modules will be:

Module: moja::flint::OutputerStream

This module inspects and outputs both Pool and Variable values on firing of defined FLINT System notifications. Outputs are either every timing step, or annually. The output format is CSV.

Warning: Summing Parent Pool values and Nested Children values will give incorrect results. As the Parent Pool value is already a sum of nested values.

Configuration additions

Current Settings

Setting	**Values**	**Description**
output_filename	string	Filename to write to
output_to_screen	TRUE|FALSE	Output to screen as well
output_info_header	TRUE|FALSE	Write info header to file
output_on_outputstep	TRUE|FALSE	Output on the OutStep notification
output_on_postdisturbanceevent	TRUE|FALSE	Output on the PostNotification notification
output_on_timingendstep	TRUE|FALSE	Output on the Timing Step notification
output_annual_only	TRUE|FALSE	Only output annually
variables	Array of strings	List of variable values to output

New Settings

Setting	Values	Description
cohort_aggregation_enabled	TRUE|FALSE Default: TRUE	Display aggregated totals.
cohort_aggregation_show_nested_values	TRUE|FALSE Default: TRUE	Display values of nested pools. If “*cohort_aggregation_enabled*” is FALSE, this will be forced to TRUE.
cohort_name_format	Long|Short Default: Short	Uses DOT notation (Long) or just pool name (Short)

Sample Module JSON config:

```
"Modules": {
   ...
   "OutputerStream": {
     "enabled": true,
     "library": "internal.flint",
     "order": 11,
     "settings": {
       "output_filename": "Stock.csv",
       "output_to_screen": true,
       "output_info_header": true,
       "output_on_outputstep": true,
       "output_on_postdisturbanceevent": false,
       "output_on_timingendstep": false,
       "output_annual_only": true,
       "variables": [
           "landUnitArea",
           "any_variable_name"
       ],
       "cohort_aggregation_enabled": true,            <<---- Addition here ---->>
       "cohort_aggregation_show_nested_values": true
       "cohort_name_format": "short"
     }
   },
   ...
},
```

Module: moja::flint::OutputerStreamFlux

This module inspects and outputs details of Fluxes that have been applied during a simulation. The output format is CSV.

Fluxes record moves between Pools, so there is no aggregation required. However, configuration flags can be used to output information to identify Cohort Parents and Children.

Configuration additions

Existing Settings

Setting	**Values**	**Description**
output_filename	string	Filename to write to
output_to_screen	TRUE|FALSE	Output to screen as well
output_info_header	TRUE|FALSE	Write info header to file

New Settings:

Setting	Values	Description
cohort_name_format	Long|Short Default: Short	Uses DOT notation (Long) or just pool name (Short)
cohort_show_parent_name	TRUE|FALSE Default: TRUE	Output columns with parent names in DOT notation. If Pool is not a Parent this will be an empty string.

Sample Module JSON config:

```
"Modules": {
   ...
   "OutputerStreamFlux": {
     "enabled": true,
     "library": "internal.flint",
     "order": 12,
     "settings": {
       "output_filename": "Flux.csv",
       "output_to_screen": false,
       "output_info_header": true,
       "cohort_name_format": "short",                 <<---- Addition here ---->>
       "cohort_show_parent_name": "short"
     }
   }
   ...
},
```

Module outputs

Stock

Current CSV format

Field name	Description
Notification	FLINT system notification fired when this record was recorded. Possible values are onOutputStep, onTimingEndStep, onPostDisturbanceEvent or onTimingPostInit
Step	Timing step
stepDate	Timing step date
fracOfStep	Fraction of step - if a disturbance event this will be a fraction indicating the length of the timing step the event occurred in.
stepLenInYears	Length of the step in Years
[pools]	Array of pool values
[variables]	Array of variable values

Proposed CSV format

Field name	Description
Notification	FLINT system notification fired when this record was recorded
Step	Timing step
stepDate	Timing step date
fracOfStep	Fraction of step - if a disturbance event this will be a fraction indicating the length of the timing step the event occurred in.
stepLenInYears	Length of the step in Years
[pools]	Array of pool values Using the settings described previously: ● Pool list displayed will vary, including Parent Aggregation and/or Child Pool values. ● Pool name format (DOT notation in some cases)
[variables]	Array of variable values

Flux

Current CSV format

Field name	Description
Step	Timing step
step_date	Timing step date
module_name	Name of module that created the Flux
disturbance_type	Integer representing a special identifier from when the operation was applied. Allowing extra info to be tagged onto a flux
source_pool	Source Pool for flux
sink_pool	Sink Pool for flux
flux	Flux amount

Proposed CSV format

Field name	Description
step	Timing step
step_date	Timing step date
module_name	Name of module that created the Flux
disturbance_type	Integer representing a special identifier from when the operation was applied. Allowing extra info to be tagged onto a flux
source_parent_pool	Parent of Source Pool for flux
sink_parent_pool	Parent of Sink Pool for flux
source_pool	Source Pool for flux
sink_pool	Sink Pool for flux
flux	Flux amount

Future Extensions

The current extension design is based around existing FLINT output Modules (*OutputerStream* and *OutputerStreamFlux*) which write results in a CSV format.

A logical future extension would be to keep the hierarchy intact in the output. There are various formats that enable this type of data structure, these include:

- JSON
- Avro
- Parquet

An example of the nested JSON format could be:

```
[{
   "year": 1999,
   "pools": [{
           "Name": "Debris",
           "Value": 11.11,
           "Nested": [{
               "Name": "debrisCM.eucalyptus",
               "Value": 11.22
           }, {
               "Name": "debrisCM.eucalyptus",
               "Value": 11.22,
               "Nested": [{
                   "Name": "debrisCM.eucalyptus.Child1",
                   "Value": 11.22
               }, {
                   "Name": "debrisCM.eucalyptus.Child2",
                   "Value": 11.22
               }]
           }]
       },
       {
           "Name": "aboveGroundCM",
           "Value": 12.22,
           "Nested": [{
               "Name": "aboveGroundCM.Child1",
               "Value": 12.22
           }]
       },
       {
           "Name": "belowGroundCM",
           "Value": 22.33,
           "Nested": [{
               "Name": "belowGroundCM.Child1",
               "Value": 12.22
           }]
       }
   ]
}]
```