Stock (using AusFarm in Apsim)

Introduction

The Stock component in Apsim is taken from the Stock component in a CSIRO sister product to Apsim called AusFarm. AusFarm is produced by the GrazPlan Decision Support Software group in CSIRO (Division of Plant Industry).

GrazPlan Official Website

GrazPlan at Plant Industry

The Ausfarm-Stock (Stock) component is used in APSIM to graze APSIM-Plant derived crops, such as Lucerne, Lablab and Wheat.

The Ausfarm-Supplement (Supplement) component can also be used as a supplementary feed source for stock.

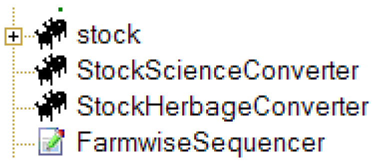
Management operations can be carried out such as buying and selling of stock, supplementary feeding and moving stock to another paddock.

Currently the paddock sizes are fixed at one hectare.

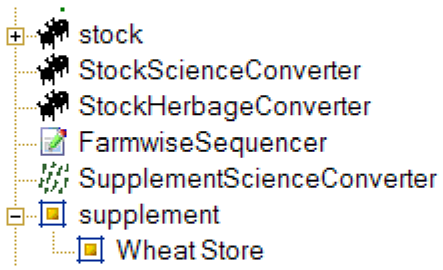
To use Stock in an APSIM simulation, three companion APSIMUI components are required

- StockScienceConverter,
- StockHerbageConverter and
- FarmwiseSequencer.

e.g.



When Supplement is used, its companion **SupplementScienceConverter** is also required. e.g.



All of these components are found in the APSIMUI Standard Toolbox in the Animals Folder. e.g.

- Animals
  - StockHerbageConverter
  - StockScienceConverter
  - stock
  - supplement
  - SupplementScienceConverter
  - FarmwiseSequencer

A sample simulation (Wether_Lucerne_Supplement) of grazing wethers on lucerne with supplementary feeding of wheat is provided under the*New* menu.

APSIMUI components

Stock

The Stockcomponent initialises with default properties and no animals.

StockScienceConverter

The StockScienceConverter initialises with the values specified in its properties grid. e.g.

Name	Value
debug	off
stock_module	stock
conversion_model	nonherbage
fraction_faeces_added	0.5
fraction_urine_added	0.5

StockHerbageConverter

The StockHerbageConverter initialises with the values specified in its properties grid. e.g.

Name	Value
debug	off
conversion_model	herbage
herbage_model	plant
herbage_module_name	lucerne

FarmwiseSequencer

The FarmwiseSequencer initialises with properties specified in its INI file. These properties translate APSIM events to Ausfarm events and should only be altered after consultation with the SEG.

Supplement

Supplement initialises with the values specified in its properties grid. e.g.

Name	Value
spoilage_time	0.0

The supplement store properties define the initial values of each supplement being used in the simulation. e.g.

Name	Value
name	wheat
stored	10000000
roughage	false
dm_content	0.89
dmd	0.9
me_content	13.8
cp_conc	0.14
prot_dg	0.92
p_conc	0.0038
s_conc	0.0014
ee_conc	0.02
adip2cp	0.02
ash_alk	0.18
max_passage	0.0

SupplementScienceConverter

The SupplementScienceConverter initialises with values specified in its properties grid. e.g.

Name	Value
debug	off
supplement_module	supplement

Pre-defined management rules of stock (cattle and sheep) and supplement feeding are found in the Stock folder under the Management folder in the Standard Toolbox. e.g.

- Stock
  - Cattle
  - Sheep
  - Supplement

Pre-defined cattle management rules for buying, selling and moving stock.

- Cattle
  - Buy cattle on a fixed date
  - Buy cattle on a fixed days after sowing
  - Move cattle on a fixed date
  - Move cattle on a weight change
  - Buy cattle on feed-on-offer
  - Buy cattle on feed-on-offer test
  - Sell cattle on a fixed date
  - Sell cattle on weight test
  - Sell cattle on weight loss
  - Sell cattle below minimum declining weight change
  - Sell cattle on minimum condition score change
  - Sell cattle on weight

Pre-defined sheep management rules for buying and selling stock.

- Sheep
  - Buy sheep on a fixed date
  - Buy sheep on feed-on-offer
  - Buy variable sheep on green feed-on-offer
  - Sell sheep on a fixed date
  - Sell sheep on weight
  - Sell sheep on green feed-on-offer

Pre-defined supplement management rules for buying, mixing and feeding stock.

- Supplement
  - Buy supplement on a fixed date
  - Mix supplement on a fixed date
  - Feed supplement in a fixed date window
  - Feed supplement on animal condition
  - Feed supplement on animal condition loss
  - Feed supplement on animal weight loss
  - Feed supplement to reach a target weight

Stock Component Description

See the "Stock Component Description" PDF (also comes with your AusFarm installation)

Drivers for the Stock Component

Environmental information

The Stock component implements alternative interfaces to obtain weather data;
I have only provided the APSIM-compatible interface here.

Property	Type	Units	Permitted number of values	Description
time	record	-	Exactly 1	Current time step in standard format.
latitude	double	deg	Exactly 1	Latitude (south is negative).
daylength	double	hr	Exactly 1	Day length including civil twilight.
maxt	double	oC	Exactly 1	Maximum air temperature.
mint	double	oC	Exactly 1	Minimum air temperature.
rain	double	mm	Exactly 1	Precipitation in all forms other than snow.
wind	double	m/s	Exactly 1	Average wind speed
waterlog	double	-	Zero or more	Waterlogging index for each paddock.

Note: The first six of these drivers will cause no difficulty.

The wind driver is used in computing the energy requirement to maintain body temperature under cold conditions;
it can be set to (say) a constant 2.0 m/s if wind speed data are not available.

The waterlog driver is an index that describes the degree of waterlogging of the soil (which is modelled as affecting the time spent grazing);
it is optional and can be ignored in your context.

Paddock information

Property	**Type**	**Units**	**Permitted number of values**	**Description**
area	double	ha	Zero or more	Area of each paddock.

Property	Type	Units	Permitted number of values	Description
slope	double	deg	Zero or more	Slope of each paddock

In the Stock component, each group of animals is taken to reside in a paddock.

The list of valid paddock names is obtained by looking for the components that have the area property and then obtaining the names of these components, i.e. a paddock is defined as any entity with an area.
If no component in the simulation has the area property, then the Stock component sets itself up with a single "paddock" that has the null string for its name, an area of 1.0 ha and zero slope.

The slope values are used to compute the energy cost associated with movement.

The usual configuration is to have the paddock components/systems at the same level in the component tree as the Stock component.

Herbage information (plant2stock)

At each time step, the Stock component requests the plant2stock driving property. Each value of plant2stock is allocated to one of the paddocks by parsing the name of the component sending it. The herbage present in each paddock is then summarised before being used in the computations of animal intake. In the case where no paddocks have been identified, all the "plant" components that provide plant2stock are allocated to the null paddock.

Once the intake rates of all animals have been computed, the rates for all animals in a paddock are summed and then allocated between the components in that paddock that provided a value for the plant2stock driver. Each such component is then sent a remove_herbage event that contains the rate of herbage removal for that plant component.

The plant2stock driving property is a record with six fields:

Field	Type	Units	Description
herbage	**array of records**		Mass and quality of the herbage on offer. The herbage is split into a number of pools that are distinguished by their DM digestibility; each of these pools is described by one element of the array. The fields of the sub-records are set out below.
propn_green	double	-	Proportion of the total herbage mass that is green
legume	double	-	Proportion of the total herbage mass that is legume. Usually 0.0 or 1.0
select_factor	double	-	Species-specific effect on the relationship between digestibility and voluntary intake. Typical values are 0.0 for C3 grasses and legumes and 0.16 for C4 grasses.

Field	Type	Units	Description
seed	**array of records**		Mass and quality of unripe and ripe seeds (elements 1 and 2, respectively). The fields of the sub-records are the same as for the *herbage* field.
seed_class	array of integer4	-	"Equivalent digestibility class" for unripe and ripe seeds (elements 1 and 2, respectively). Valid values are 1-6 or 0, where 1 denotes that seeds will be selected at the same time as 80% digestible herbage, 2 corresponds to 70% herbage, etc. A zero value denotes that seeds are not grazed.

Both plant2stock:herbage and plant2stock:seed have array of records as a field.
The fields of the sub-records in them, are:

Field	Type	Units	Description
dm	double	kg/ha	Mass (dry matter basis) of the herbage or seed pool
dmd	double	-	Dry matter digestibility (0-1, not percentage)
cp_conc	double	kg/kg	Crude protein concentration. Can be estimated as 6.25 x N
p_conc	double	kg/kg	Phosphorus concentration. Can reasonably be estimated by assuming a fixed N:P ratio.
s_conc	double	kg/kg	Sulphur concentration. Can reasonably be estimated by assuming a fixed N:S ratio.
prot_dg	double	kg/kg	Protein degradability. In the absence of better information, can be estimated as *dmd* +0.10
ash_alk	double	mol/kg	Ash alkalinity.
height_ratio	double	-	An index of the bulk density of the herbage. The height ratio for herbage should be computed as 100/(0.03 x*BD*), where *BD* is the herbage bulk density in g/m 3 . This value should be set to 1.0 for seed pools.

One way to obtain a distribution of herbage mass into DMD classes is given on page 32 of the technical paper on GrazFeed that can be found at

GrazFeed Technical Paper
Enternal Link - http://www.pi.csiro.au/grazplan/files/TechPaperJan10.pdf

Supplementary feed information (supp2stock)

At each time step, the Stock component requests the supp2stock driving property. This driving property is optional. The supp2stock property returns the amount and quality of supplementary feed that is present in each paddock. All animals resident in a paddock can access the corresponding quantity of supplementary feed (it is allocated between them in proportion to their maximum intake rate). To provide this property, include the Supplement component in the simulation.

The supp2stock driving property is an array of records. Each sub-record has the following fields:

Field	Type	Units	Description
paddock	string		Name of a paddock
amount	double	kg	Amount of supplementary feed present in the paddock
roughage	boolean		TRUE i.f.f. the feed is a roughage.
dm_content	double	kg/kg	Dry matter content of the feed.
dmd	double	-	Dry matter digestibility of the feed (not including any portion that passes the gut undamaged).
me_content	double	MJ/kg	Metabolizable energy content of the feed.
cp_conc	double	kg/kg	Crude protein content of the feed.
prot_dg	double	kg/kg	Protein degradability of the feed.
p_conc	double	kg/kg	Phosphorus content of the feed.
s_conc	double	kg/kg	Sulphur content of the feed.
ee_conc	double	kg/kg	Ether-extractable content of the feed.
adip2cp	double	kg/kg	Proportion of crude protein that is insoluble in acid detergent.
ash_alk	double	mol/kg	Ash alkalinity of the feed.
max_passage	double	kg/kg	Maximum proportion of the feed that will pass undamaged through the gut of ruminants.

DDML definitions

This is for the programmers:

plant2stock:

```
<type>
  <field name = "herbage" array="T">
    <element>
      <field name="dm" unit="kg/ha" kind="double"/>
      <field name="dmd" unit="-" kind="double"/>
      <field name="cp_conc" unit="kg/kg" kind="double"/>
      <field name="p_conc" unit="kg/kg" kind="double"/>
      <field name="s_conc" unit="kg/kg" kind="double"/>
      <field name="prot_dg" unit="kg/kg" kind="double"/>
      <field name="ash_alk" unit="mol/kg" kind="double"/>
      <field name="height_ratio" unit="-" kind="double"/>
    </element>
  </field>
  <field name="propn_green" unit="-" kind="double"/>
  <field name="legume" unit="-" kind="double"/>
  <field name="select_factor" unit="-" kind="double"/>
  <field name = "seed" array="T">
    <element>
      <field name="dm" unit="kg/ha" kind="double"/>
      <field name="dmd" unit="-" kind="double"/>
      <field name="cp_conc" unit="kg/kg" kind="double"/>
      <field name="p_conc" unit="kg/kg" kind="double"/>
      <field name="s_conc" unit="kg/kg" kind="double"/>
      <field name="prot_dg" unit="kg/kg" kind="double"/>
      <field name="ash_alk" unit="mol/kg" kind="double"/>
      <field name="height_ratio" unit="-" kind="double"/>
    </element>
  </field>
  <field name = "seed_class" unit="-" kind="integer4" array="T"/>
</type>
```

supp2stock:

```
<type array="T">
  <element>
    <field name="paddock" kind="string"/>
    <field name="amount" unit="kg" kind="double"/>
    <field name="roughage" kind="boolean"/>
    <field name="dm_content" unit="kg/kg" kind="double"/>
    <field name="dmd" unit="-" kind="double"/>'
    <field name="me_content" unit="MJ/kg" kind="double"/>
    <field name="cp_conc" unit="kg/kg" kind="double"/>
```

```
    <field name="prot_dg" unit="kg/kg" kind="double"/>
    <field name="p_conc" unit="kg/kg" kind="double"/>
    <field name="s_conc" unit="kg/kg" kind="double"/>
    <field name="ee_conc" unit="kg/kg" kind="double"/>
    <field name="adip2cp" unit="kg/kg" kind="double"/>
    <field name="ash_alk" unit="mol/kg" kind="double"/>
    <field name="max_passage" unit="kg/kg" kind="double"/>
  </element>
</type>
```

Stock Science Converter

Summary

The Stock Science Converter APSIM module provides science conversions between the Ausfarm-Stock component and APSIM modules. This is done on an as-needed basis, converting incompatible data names, units and structures, translating events and transforming data to fit the science model. Compatible data communications occur directly between Ausfarm-Stock and APSIM modules.

The Stock Science Converter (Converter) must handle a number of two-way conversions. The stock can eat herbage from a number of sources within the field, such as multiple APSIM crops and surface organic matter. This requires the respective herbage sources being identified and the herbage stored into pools of digestibility classes. After the stock have eaten and removed herbage from these pools, the appropriate amounts must be removed from each of the herbage sources. To handle this, the APSIM module containing the herbage is represented to the stock by a copy (instance) of the converter. For example, suppose lablab is on offer as a herbage pool in the module APSIM-lablab. An instance of the converter, named cattlelablab, would represent APSIM-lablab to Ausfarm-Stock, acting as an intermediary or translator between the two modules.

Another set of conversions relate to the herd of stock in the field, which include management and the return of organic and inorganic matter to the field through excreta. These conversions are independent of the herbage sources and here the Converter interacts with the appropriate APSIM modules, such as SoilN, SurfaceOM and Manager. In this case, another instance or copy of the Converter handles these whole of herd conversions e.g. cattleconverter.

To achieve these different conversions, the Converter has two major components, non-herbage and herbage.

The non-herbage component services the whole of herd information not directly related to herbage, such as return of excreta components, management events for buy, move and sell and data translation such as daylength.

The herbage component services information directly related to grazing of APSIM crops and surface organic matter. Because the data relating to each is different, the herbage component consists of two sub-components, plant and surfaceOM. The plant and surfaceOM sub-components service the removal of herbage and residues by the stock module from the plant and surface organic matter modules respectively.

Introduction

Plug-n-Play simulation architectures such as APSIM provide the framework and protocol for data exchange between modules. This facilitates the adaptation of other models by providing an APSIM communications protocol interface and creating a system boundary to the model that aligns with the system boundaries of the APSIM modules it communicates with.

APSIM uses a common modelling protocol which allows components from other systems using the same protocol to be plugged into APSIM without modification, such as AusFarm components, one being a Stock component.

Models adapted to the APSIM environment sometimes have different system boundaries and more commonly their science communication is not compatible with other APSIM modules. This incompatibility is often in the form of data content, data units, data structure or events which can be corrected by converting the science each way. One approach would be to enclose the module in a wrapper which would provide a complete interface between the module and the APSIM system, thus hiding the modules interface from other APSIM modules. The approach taken here is to provide a science converter module which interacts with the introduced module and other APSIM modules on an as-needed basis, thus only providing conversions for the incompatible data. This approach relies on the names of the incompatible data being different to data names and events elsewhere in the system.

The Stock Science Converter APSIM module provides science conversions between the Ausfarm-Stock component and APSIM modules. This is done on an as-needed basis, converting incompatible data names, units and structures, translating events and transforming data to fit the science model. Compatible data communications occur directly between Ausfarm-Stock and APSIM modules.

Module components

The Stock Science Converter (Converter) must handle a number of two-way conversions. The stock can eat herbage from a number of sources within the field, such as multiple APSIM crops and surface organic matter. This requires the respective herbage sources being identified and the herbage stored into pools of digestibility classes. After the stock have eaten and removed herbage from these pools, the appropriate amounts must be removed from each of the herbage sources. To handle this, the converter represents the sources with instances of itself, each instance representing a herbage source provided to the stock e.g. *cattlelablab*.

Another set of conversions relate to the herd of stock in the field, which include management and the return of organic and inorganic matter to the field through excreta. These conversions are independent of the herbage sources and the Converter interacts with the appropriate APSIM modules, such as SoilN, SurfaceOM and Manager. Another instance of the Converter handles these whole of herd conversions e.g. *cattleconverter*.

To achieve these different conversions, the Converter has two major components, non-herbage and herbage.

The non-herbage component services the whole of herd information not directly related to herbage, such as return of excreta components to surface organic matter, soilN and soilP, management events for buy, move and sell and data translation such as daylength.

The herbage component services information directly related to grazing of APSIM crops and surface organic matter. Because the data relating to each is different, the herbage component consists of two sub-components, plant and surfaceOM. The plant and surfaceOM sub-components service the removal of herbage and residue by the stock module from the plant and surface organic matter modules respectively.

StockScienceConverter Module

Description

The Stock Science Converter APSIM module provides science conversions between the Ausfarm-Stock component and APSIM modules. This is done on an as-needed basis, converting incompatible data names, units and structures, translating events and transforming data to fit the science model. The module is instantiated once for the non-herbage conversions and then for herbage conversions for each crop being grazed.

Non-herbage component

The non-herbage component services the information not directly related to herbage, such as return of excreta components to surface organic matter, soilN and soilP, management events for buy, move and sell and data translation such as daylength.

When an *add_excreta* event is received from the stock module, the converter extracts the urea N and phosphate P from the *urine* component of the excreta and deposits a nominated fraction of the amounts into the top layers of soilN and soilP modules. Currently, the sulphate S and ash alkalinity amounts are ignored.
The converter extracts the faeces_om component and sends an *add_surfaceom* event along with the organic matter component. The SurfaceOM module receives this event and adds the DM etc to its pools.

Buystock, *sellstock* and *movestock* events received from the manager module are sent on to stock as *buy*, *sell* and *move* events respectively with reconstructed parameters.

Herbage component

The herbage component services information directly related to the actions of stock on APSIM plant and surface organic matter. The Herbage component exchanges data with the stock module and instantiates a sub-component (herbage model) for either plant or surfaceom (surface organic matter) interactions. This sub-component exchanges data with plant or surface organic matter modules and carries out the appropriate translations between stock and herbage science

When the Herbage component receives a request from stock for *plant2stock* data, it fills this property with data from the herbage model and sends it to stock.

When the Herbage component receives a *Remove_herbage* event, it gives the associated data about quantity of herbage removed to the herbage model.

Plant sub-component

The plant sub-component (herbage model) services the removal of herbage by the stock module from the plant module. It identifies the crop it interacts with and represents to stock by the *herbage_module_name* property.

When its parent Herbage component requests herbage data, it obtains the dry weights from the nominated crop, for green and senesced material, along with the associated N, P and digestibility. It then transforms the science to that of stock and gives this data to the Herbage component.

When its parent Herbage component passes the herbage removed by the stock, it transforms this data to plant science form and sends it as deltas with a *remove_crop_biomass* event to the crop it represents. A trampling effect is calculated and sent with a *detach_crop_biomass_rate* event to the crop for detachment of senesced leaf.

Conversion of plant science to stock science consists of

- distributing the plant part dry-weights over six digestibility class pools defined by the minimum, maximum and average digestibilites of each plant part. The distribution is calculated using a bell shaped polynomial, with the average digestibility determining the class with the highest proportion of dry-matter and the minimum and maximum digestibilities determining the highest and lowest digestibility classes (Freer *et al*, 1997; Freer *et al*, 2003).

  As plant parts are distributed into the stock digestibility class pools, a record of the proportions of each plant part put into each class pool is kept. The plant part dry-weights and these proportions are then used to calculate the contribution of each plant part to each digestibility class pool. These contributions are then used to distribute the herbage removed from the six digestibility class pools into the plant part deltas. The assumption is that all plant parts in a digestibility class pool are removed or grazed proportionally to their contribution to that pool.

  Seeds (meal and oil) are distributed into two digestibility classes (ripe/unripe) (Moore, 2005). The crop stage is used to determine the class, so at and after maturity the seeds are put into the ripe class, otherwise they are stored in the unripe class.

- For each digestibility class pool, calculating crude protein (Cp) concentration from the N content and a Cp:N ratio.
- For each digestibility class pool, calculating phosphorus (P) concentration if P is not available from the plant module, using a default N:P ratio.
- For each digestibility class pool, calculating sulphur concentration using a default N:S ratio.
- For each digestibility class pool, calculating protein degradability as dry-matter digestibility + 0.1 (Moore, pers. com.).
- For each digestibility class pool, calculating ash alkalinity, using default values if it is not available from the plant module.
- For each digestibility class pool, calculating the height ratio, an index of the bulk density of the herbage (Moore, pers. com.).

Surfaceom sub-component

The surfaceOM sub-component services the removal of residue by the stock module from surface organic matter module.

References

Freer M, Moore AD & Donnelly JR (1997). GRAZPLAN: decision support systems for Australian grazing enterprises. II. The animal biology model for feed intake, production and reproduction and the GrazFeed DSS. *Agricultural Systems* **54**, 77-126.

Freer M, Moore AD & Donnelly JR (2003). The GRAZPLAN animal biology model for sheep and cattle and the GrazFeed decision support tool. *CSIRO Plant Industry Technical Paper*, September 2003.

Moore AD, Donnelly JR & Freer M, (1997). GRAZPLAN: decision support systems for Australian grazing enterprises. III. Pasture Growth and Soil Moisture Submodels, and the GrassGro DSS. *Agricultural Systems* **54**, 77-126.

Moore AD (2005). Component Description – Stock. Internal document, March 2005.

Appendix 1

The components can be described in terms of properties and events. Properties and events can be described in terms of published and subscribed with properties additionally having initialisation and owned.

- Initialisation properties are parameters read at the start of the simulation.
- Owned properties are owned by the module. These properties are passively available to other modules.
- Published properties and events are owned by the module. These properties and events are actively sent to the system.
- Subscribed properties and events are those the module requires to start. These properties are actively sought from the system when needed while these events are passively received and acted upon.

StockScienceConverter

Properties - Initialisation

Conversion_model (nonherbage/herbage) – instantiates a nonherbage or herbage component.

Non-herbage

- **Properties**
  - **Initialisation**
    - *Debug* (on/off) - for producing detailed daily information in summary file.
    - *stock_module* – simulation name of stock module
    - *fraction_urine_added* – fraction of excreted urine added to soil. The remainder is assumed to be dropped at shade and watering points.
    - *fraction_faeces_added* - fraction of excreted faeces added to surface organic matter. The remainder is assumed to be dropped at shade and watering points.
  - **Owned**
    - *Daylength* (hr) – daylength including civil twilight for use by stock (Moore, 2005).
  - **Published**
    - *Dlt_urea* (kg/ha) – adds Urea N excreted in urine to soilN.
    - *Dlt_labile_p* (kg/ha) – adds phosphate P excreted in urine to soilP.
  - **Subscribed**
    - *Day_length* (hr) daylength including civil twilight
    - *Urea* – to obtain number of layers in urea profile for publishing dlt_urea
    - *Labile_p* - to obtain number of layers in P profile for publishing dlt_labile_p
- **Events**
  - **Published**
    - *Buy* – sends a buy event to stock (Moore, 2005).
    - *Sell* – sends a sell event to stock (Moore, 2005).
    - *Move* – sends a move event to stock (Moore, 2005).
    - *Add_surfaceom* – adds faeces drymatter with its N, P, S content and ash alkalinity to surface organic matter.
  - **Subscribed**
    - *Buystock* – event from the manager to buy stock. Parameters are the same as for the buy event.
    - *Sellstock* – event from manager to sell stock. Parameters are the same as for the sell event.
    - *Movestock* – event from manager to move stock. Parameters are the same as for the move event.
    - *Add_excreta* –event from stock describing excretion of faeces and urine (Moore, 2005).

Herbage

- **Properties**
  - **Initialisation**
    - *Debug* (on/off) - for producing detailed daily information in summary file.
    - *herbage_model* (plant/ surfaceom) – instantiates a plant or surfaceom sub-component.
- **Events**
  - **Subscribed**
    - *Remove_herbage* – event from stock describing removal of herbage and seeds (Moore, 2005).
- **Properties**
  - **Owned**
    - *plant2stock* – description of feed on offer for use by stock (Moore, 2005).
    - *herbage_trampling* – trampling factor by stock for reporting.
    - *dm_feed_on_offer* – drymatter array of feed on offer to stock in each of the 6 herbage digestibility classes and 2 seed digestibility classes, for reporting.
    - *dm_feed_removed* – drymatter array of feed removed by stock in each of the 6 herbage digestibility classes and 2 seed digestibility classes, for reporting.
    - *dmd_avg_feed_removed* – weighted average drymatter digestibility of feed removed by stock.
  - **Subscribed**
    - *Trampling* – obtained from stock as trampling effect on herbage (Moore, 2005; Moore *et al*, 1997). Currently receiving 0. NEEDS FIXING

Plant

- **Properties**
  - **Initialisation**
    - *Debug* (on/off) - for producing detailed daily information in summary file.
    - *herbage_module_name* (lablab/wheat/…) – identifies the crop this instance represents to stock. This allows multiple instances representing multiple crops in the system.
  - **Owned**
    - *dm_parts_removed* – array of drymatter grazed from each plant part, for reporting. In order of green (leaf, stem, pod, meal, oil) and senesced.
  - **Subscribed**
    - For each plant part (Leaf, Stem, Pod, Meal, Oil), green and senesced drymatter (*GreenWt*, *SenescedWt*) along with the N and P content (*GreenN*, *SenescedN*, *GreenP*, *SenescedP*) and digestibilities (*digestibility_max_dm_green/senesced*, *digestibility_avg_dm_green/senesced*, *digestibility_min_dm_green/senesced*)
    - *Stage* and *stage_name* – from plant, used to determine seed class (ripe/unripe)
    - *Height* – from plant, used to calculate height ratio for herbage description in plant2stock property (Moore, 2005).
- **Events**
  - **Published**
    - *remove_crop_biomass* – event to plant describing the drymatter of each plant component removed.
    - *detach_crop_biomass_rate* – event to plant describing the fraction of senesced material to detach from trampling (Moore *et al*, 1997).
  - **Subscribed**

Surfaceom

- **Properties**
  - **Initialisation**
  - **Published**
  - **Subscribed**
- **Events**
  - **Published**
  - **Subscribed**

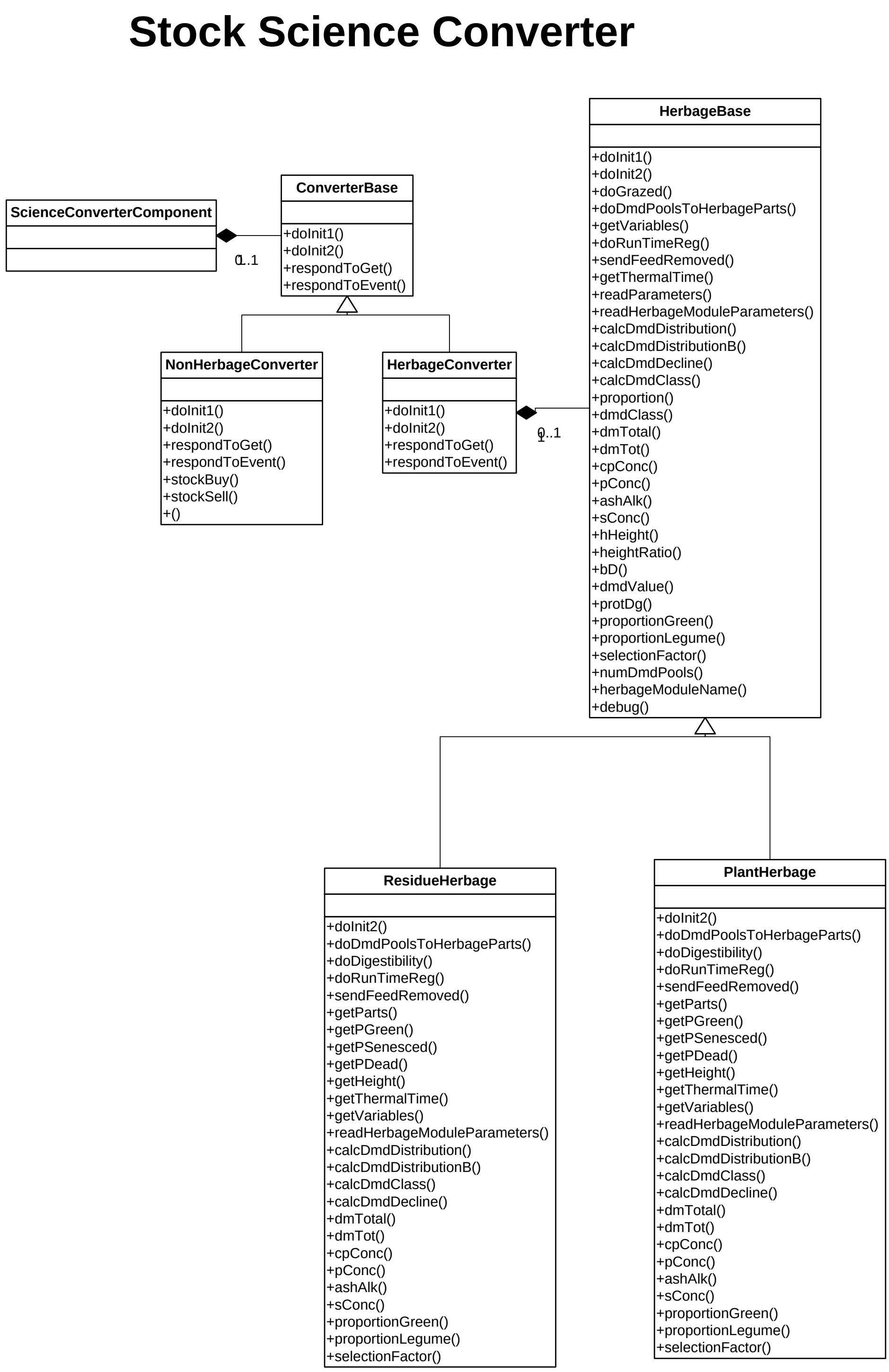
Appendix 2

The plant sub-component (herbage model) initialises with the following parameters located in the StockScienceConverter.ini file.

- *dmdValue*
  This is set to six digestibility classes into which the herbage is classified. Usually 0.8, 0.7, 0.6 0.5, 0.4, 0.3.
- *specific_detach_rate*
  Daily detachment rate of senesced herbage due to trampling by stock. Set to 0.000002 (ha/kg animal/day) (Moore *et al*, 1997)
- *proportion_legume*
  This is the proportion of legume in this herbage. Usually 0.0 or 1.0
- *dmd_seed*
  The digestibility of unripe and ripe seed. Usually 0.80, 0.80 (Moore, pers. com.).
- *seed_class*
  Equivalent digestibility class for concurrence with selective grazing of herbage classes - 1 = 0.8, 2 = 0.7, etc. Usually 1, 1. (Moore, pers. com.)

- *cp_n_ratio*
  The crude protein to Nitrogen ratio used to estimate the crude protein content of herbage. Set to 6.25(Moore, pers. com.).
- default P concentration for each plant part.
  Inactivated. This is read but currently over-ridden by n:P ratio. E.g. *p_conc_green_leaf_default*
- default Ash alkalinity for each plant part.
  E.g. *ash_alk_green_leaf_default*
- default N:S concentration ratio for each plant part. Used to estimate the Sulphur content of the herbage for its Nitrogen content. E.g. *ns_ratio_green_leaf_default*
- default N:P concentration ratio for each plant part. Used to estimate the Phosphorus content of the herbage from its Nitrogen content. E.g. *np_ratio_green_leaf_default*

Appendix 3. Static Class Diagram



Cattle Science Converter

APSIM Control:

Instantiate Stock as stock

Stock DLL representing the grazing animals

Reference: Stock Component Description.doc

Example:

component name="stock"

executable="C:\Program Files\FarmWise\stock.dll"

Instantiate StockScienceConverter as cattleconverter

Converts non-herbage data

Example:

component name="cattleconverter" executable="%apsuite\apsim\stock\lib\StockScienceConverter.dll"

Instantiate StockScienceConverter as cattle*cropmodulename*

Converts Herbage data

Where cropmodulename is the name of the crop being grazed.

Example:

component name="cattlelablab" executable="%apsuite\apsim\stock\lib\StockScienceConverter.dll"

APSIM Manager:

Buy stock

cattle buystock number = *n* (-), genotype = *gtext*, sex = *stext*, age = *m* (months), weight = *x* (kg)

Where:

n = number of stock

gtext = genotype of stock

stext = sex of stock

m = age of stock in months

x = weight of stock in Kg

Example:

cattle buystock number = 20 (-), genotype = angus, sex = steer, age = 12 (months), weight = 250.0 (kg)

Sell stock

cattleconverter sellstock number = *n* (-)

Where:

n = number of stock

Example:

cattleconverter sellstock number = 3 (-)

Initialisation:

Stock

Example:

genotypes

name = brahman

srw = 550.0

name = shorthorn

srw = 550.0

cattle

name = Brahman

number = 0

death_rate = 0.01

age = 365

weight = 250

sheep

Cattleconverter

Example:

debug = off

conversion_model = nonherbage - *non-herbage conversion*

Cattlecropmodulename

Example:

herbage_module_name = lablab - *module name to interface with*

debug =off

conversion_model = herbage - *herbage conversion*

herbage_model = plant - *plant or residue herbage type*

digestibility values of 6 digestibility classes

dmdValue = 0.8 0.7 0.6 0.5 0.4 0.3

Default values used if not available from specified plant module

p_conc_green_leaf_default = 0.004 - *P concentration* p_conc_green_stem_default = 0.003 *not used*

p_conc_senesced_leaf_default = 0.004

p_conc_senesced_stem_default = 0.003

p_conc_dead_leaf_default = 0.004

p_conc_dead_stem_default = 0.003

ash_alk_green_leaf_default = 254.0 - *Ash alkalinity*

ash_alk_green_stem_default = 96.0 - *cmol/kg*

ash_alk_senesced_leaf_default = 254.0

ash_alk_senesced_stem_default = 96.0

ash_alk_dead_leaf_default = 254.0

ash_alk_dead_stem_default = 96.0

ns_ratio_green_leaf_default	= 19.0	*- N:S ratio to calculate S*
ns_ratio_green_stem_default	= 11.0	*concentration from N*
ns_ratio_senesced_leaf_default	= 19.0	
ns_ratio_senesced_stem_default	= 11.0	
ns_ratio_dead_leaf_default	= 19.0	
ns_ratio_dead_stem_default	= 11.0	

np_ratio_green_leaf_default	= 8.0	*- N:P ratio to calculate P*
np_ratio_green_stem_default	= 8.0	*concentration from N*
np_ratio_senesced_leaf_default	= 8.0	
np_ratio_senesced_stem_default	= 8.0	
np_ratio_dead_leaf_default	= 8.0	
np_ratio_dead_stem_default	= 8.0	

Digestibility of specified plant parts – max, average, minimum

dmd_green_leaf	=	0.80	0.65	0.60
dmd_green_stem	=	0.60	0.55	0.50
dmd_senesced_leaf	=	0.60	0.50	0.40
dmd_senesced_stem =	0.50	0.35	0.30	
dmd_dead_leaf	=	0.40	0.40	0.40
dmd_dead_stem	=	0.30	0.30	0.30

cp_n_ratio = 6.25 *- crude protein : N ratio to calculate crude protein from N*

Subscribed Events

NonHerbage

Buystock

Sellstock

add_excreta

Herbage

remove_herbage

Plant

None

Residue

None

Published Events

NonHerbage

Buy

Sell

add_surfaceom

Herbage

None

Plant

None

Residue

None

Owned Properties

NonHerbage

dayLength

Herbage

dm_feed_removed

dm_feed_on_offer

plant2stock

Plant

None

Residue

None

Subscribed Properties

NonHerbage

day_length

trampling

urea

labile_p

Herbage

trampling

Plant

dm_green

dlt_dm_green

dlt_dm_green_retrans

n_green

p_green

dm_senesced

dlt_dm_senesced

dlt_dm_detached

n_senesced

p_senesced

dm_dead

dlt_dm_green_dead

dlt_dm_senesced_dead

dlt_dm_dead_detached

n_dead

p_dead

height

tt_tot

tt_tot(1-2)

Residue

None

Published Properties

NonHerbage

dlt_urea

dlt_labile_p

Herbage

None

Plant

None

Residue

None