**Build Land Unit module**

Read the events and events values

Read the land type, land use, management and input from the “land use events”

Sort the events by the date

**Land Use module**

Collect the soil type and climate zone

**For each time step:**

area = 1; F\_LU = -1, F\_MG = -1, F\_I = -1, R = 10;

Get EF\_1 value (if default, dry or wet)

Update F\_LU, F\_MG, F\_I if there are changes got in the change factor

SOC\_REF\_0 = SOC\_REF \* F\_LU \* F\_MG \* F\_I \* area;

delta\_C = (SOC\_REF\_0 - prevSOC) \* 30 / (20 \* 365)

if (prevSOC != -1) and (delta\_C > 0) soil to atmosphere have carbon movement of (delta\_C \* 1000 \* EF\_1\_value / R);

**Disturbance Event module**

**For all the timing steps:**

1. If planted, Get crop type and FCR\_table

yield = previous yield + FCR\_table["Growth\_Rate"]

decayRate = FCR\_table["Decay\_Rate"]

1. If harvested, debris transfer to atmosphere (0.5)

If there are disturbance events:

**NFertEvent:**

{

"date": {

"$date": "2001/05/01"

},

"id": 2,

"type": "agri.NFertEvent",

"name": "Organic fertilizer",

"quantity": 200,

"runtime": 5

}

1. get EF1 might be default, dry, wet or synth\_wet
2. soil to atmosphere have carbon movement of (fert.quantity \* EF\_1\_value) / fert.runtime

**EmissionEvent:**

{

"date": {

"$date": "2001/05/01"

},

"id": 2,

"type": "agri.EmessionEvent",

"name": "Organic fertilizer",

"quantity": 200,

"runtime": 5

}

1. get EF1 might be default, dry, wet or synth\_wet
2. soil to atmosphere have carbon movement of (fert.quantity \* EF\_1\_value) / fert.runtime

**HarvestEvent:**

{

"date": {

"$date": "2002/07/01"

},

"id": 3,

"type": "agri.HarvestEvent",

"name": "Winter Wheat",

"frac\_burnt": 0,

"frac\_renew": 0,

"frac\_remove": 0

}

1. get EF1 might be default, dry, and wet
2. Get DRY, R\_AG, R\_S, N\_AG, N\_BG, cf from FCR table
3. area = 1

dry\_crop\_yield = yield \* DRY

above\_ground\_dry\_residue = dry\_crop\_yield \* R\_AG

below\_ground\_residue = (dry\_crop\_yield + above\_ground\_dry\_residue) \* R\_S \* area \* harvest.frac\_renew;

F\_CR = above\_ground\_dry\_residue \* N\_AG \* (1.0 - harvest.frac\_remove - (harvest.frac\_burnt \* cf)) + below\_ground\_residue \* N\_BG;

1. Soil transfer to debris (F\_CR \* EF\_1\_value)
2. Harvested set true, crop type set to default, yield is 0, planted is false

**ManureManagementEvent:**

{

"date": {

"$date": "2003/08/01"

},

"id": 3,

"type": "agri.ManureManagementEvent",

"name": "Manure Management Event",

"N\_cdg": 5,

"N\_bedding": 5,

"animal\_type": ["Dairy cattle", "Buffalo", "Sheep"],

"no\_livestock": [10, 10, 5],

"productivity\_class": ["High", "High", "Low"],

"use": ["", "Dairy", "Dairy"],

"frac\_feed": 0,

"frac\_cnst": 0,

"frac\_fuel": 0

}

1. Read EF3 (from config files) might be other cattle default, cattle dry, and cattle wet and EF1 (from config files) might be categorized default, dry, and wet
2. Get AWMS\_manure\_management, region, animal type, N\_excretion\_rate (ex\_rate) , nitrogen Loss
3. Region 1 = region + productivity\_class(high/low/mean), region 2 = region, NMMS = 0, R\_N2 = 3
4. For each of the members of the AWMS manure (animal waste management system)
   1. N\_rate = ex\_rate[region\_1]

weight = Animal\_weights[region\_1]

N\_ex = N\_rate \* weight / 1000

MS = AWMS[region\_2]

Fprp = no\_livestock \* N\_ex \* MS

* 1. Soil transfer to atmosphere (Fprp \* EF\_3\_value)
  2. EF\_3\_system = nitrogenLoss["EF\_3"],

frac\_gas = nitrogenLoss[animal\_type + " Gas"]

frac\_leach = nitrogenLoss[animal\_type + " Leach"];

* 1. frac\_N2MS = R\_N2 \* EF\_3\_system

frac\_loss = frac\_gas + frac\_leach + frac\_N2MS + EF\_3\_system

NMMS += (no\_livestock \* N\_ex \* AWMS\_system + N\_cdg) \* (1 - frac\_loss) + no\_livestock \* AWMS\_system \* N\_bedding

1. Soil transfer to atmosphere (NMMS \* (1 - (frac\_feed + frac\_cnst + frac\_fuel)) \* EF\_1\_value)

**PlantEvent:**

{

"date": {

"$date": "2002/03/01"

},

"id": 3,

"type": "agri.PlantEvent",

"name": "Winter Wheat"

}

1. set planted to be true and set the crop\_type

**Spatial transform:**

Add a harvest and plant event,

Add two fertilizer events.

Add land use ("landtype", "landuse", "management", "input")

Add one manure management event.

**Time Series transform:**

Read the timeseries var of each time step

(change "vars": [ "landtype", "landuse", "management", "input"])