Ruminant Farm Systems Model: Focus on Enteric Emissions

September 2023

Many models are already out there

- Dairy contributions to climate change are widely discussed but difficult to measure.
- Companies and NGOs need tools to quantify dairy farm emissions and help suppliers achieve net zero emissions.
- Existing models do not capture the complex dynamics on dairy farms, so confusion and mistrust has arisen among dairy industry users.

TRUTERRA

CFT

DSSAT

AMTS

Integrated Farm System Model
Version 4.5
USDA / Agricultural Research Service
Pasture Systems and Watershed Management Research Unit
University Park, Pennsylvania

FARMERS ASSURING RESPONSIBLE MANAGEMENT FARM

FARM Environmental Stewardship
Version 2 Updates

SWAT Soil & Water Assessment Tool

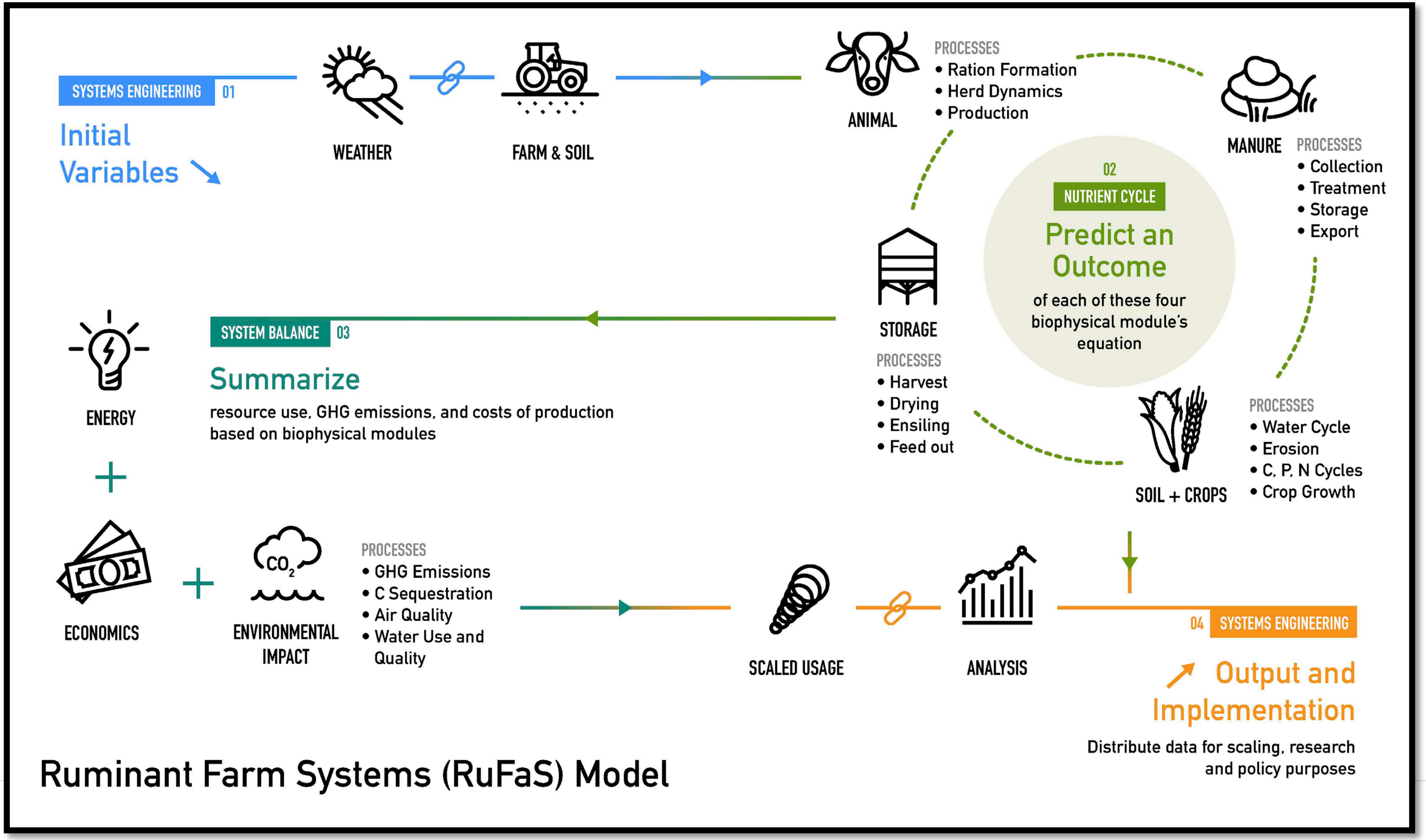
COMET Farm
USDA United States Department of Agriculture Natural Resources Conservation Service
Colorado State University
Whole Farm and Ranch Carbon and Greenhouse Gas Accounting System.

What is RuFaS?

A Next-Generation, Whole-Farm, Dairy Sustainability Simulation Model

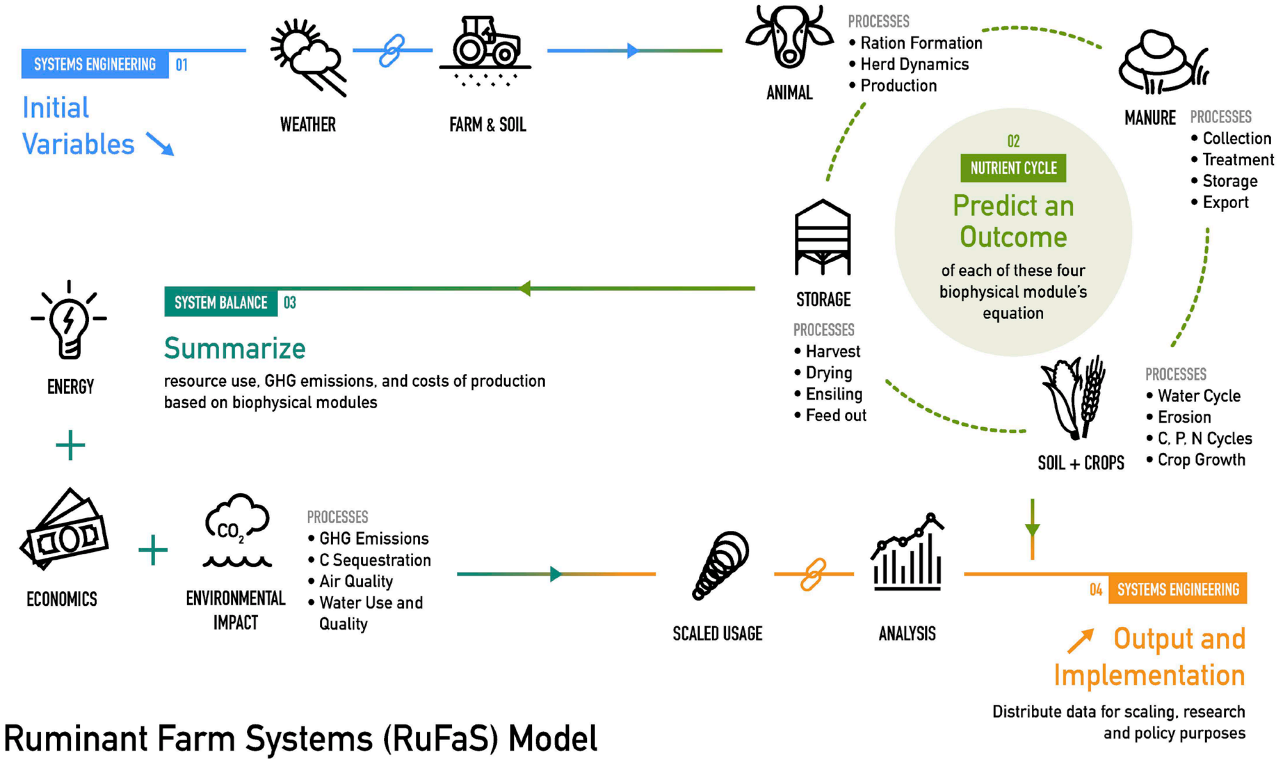
- Simulates dairy farm production and environmental impact
- Identifies ways to improve efficiency and sustainability
- Has a range of applications, from a research tool for scientists to a decision-aid tool for the dairy industry
- Coding emphasizes transparency and accessibility to ensure model flexibility, clarity, adaptability, and persistence

What is RuFaS?



The RuFaS Vision

To ***support research and sustainable decision-making*** in ruminant animal production through ***a state-of-art, open-source modeling environment*** that is continuously adapting as technology and scientific knowledge advance.



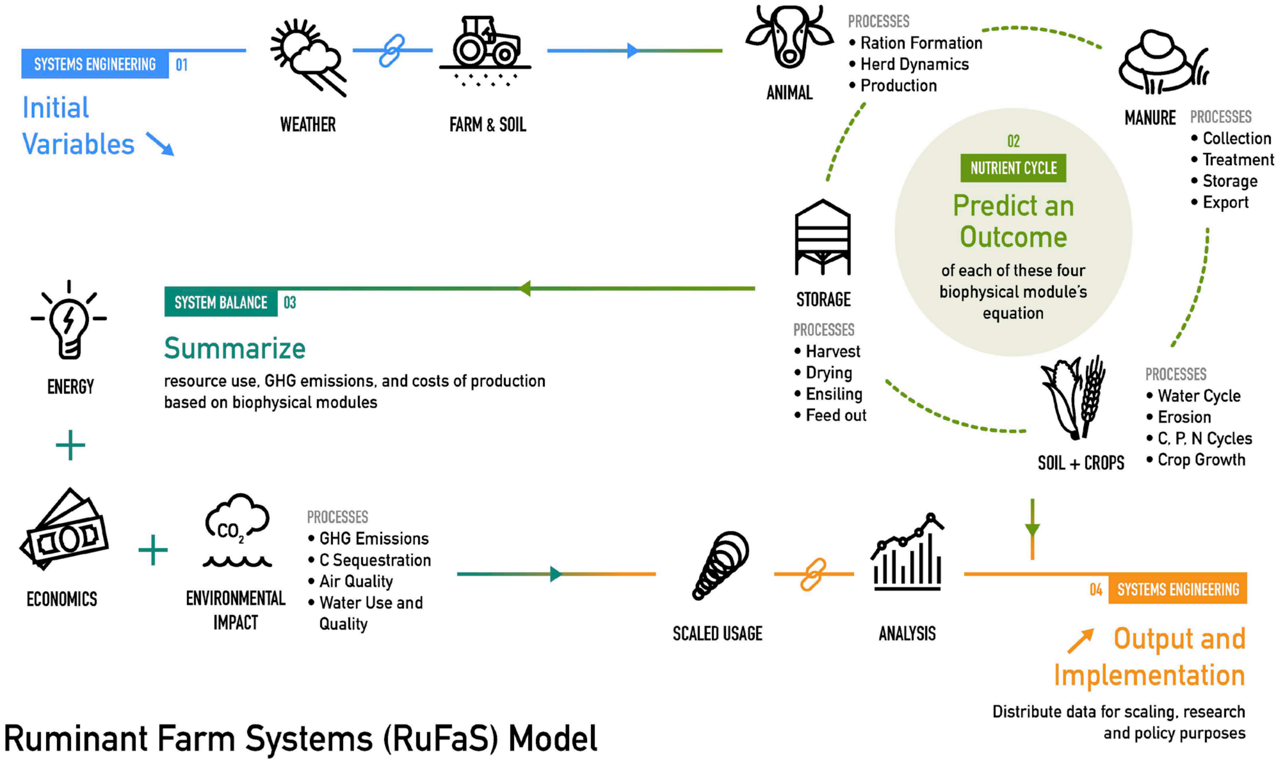
Ruminant Farm Systems (RuFaS) Model

The RuFaS Mission

To ***build an integrated, whole-farm model*** that simulates milk, meat, and crop production, greenhouse gas emissions, water quality impacts, soil health, and other ***sustainability outcomes*** of ruminant farms.

We strive to achieve the ***highest standards for prediction accuracy***, ***code structure*** and clarity, ***documentation***, and ***accessibility***.

Through ***continuous learning*** and improvement of our methods and algorithms, we are ***creating an open and inclusive platform*** for scientific collaboration.

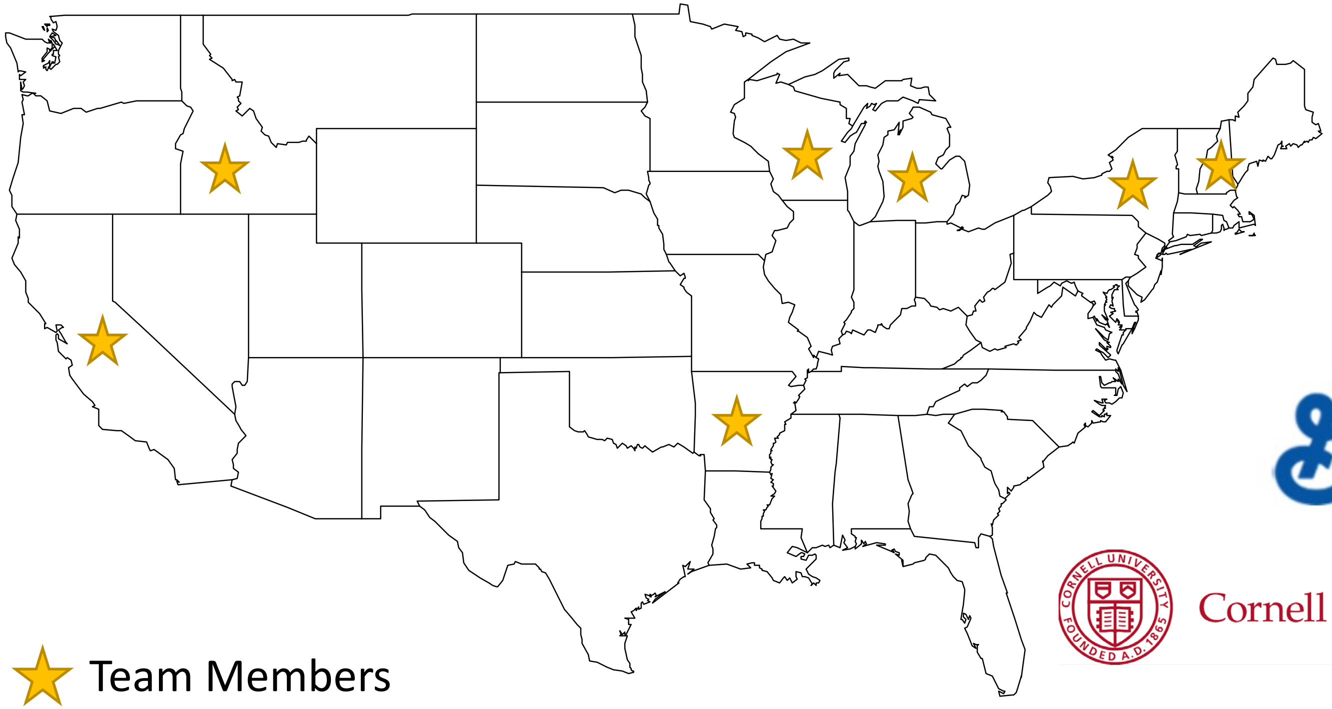


Ruminant Farm Systems (RuFaS) Model

RuFaS Goals

- **Interoperable**
- **Documented**
- **Open Source**
- **Sustainable**

RuFaS Team



★ Team Members

ANIMAL

MANURE

SOIL + CROPS

ENERGY

ECONOMICS

General Mills Making Food People Love

Cornell University

USDA

WISCONSIN UNIVERSITY OF WISCONSIN–MADISON

INNOVATION CENTER FOR U.S. DAIRY. HEALTHY PEOPLE • HEALTHY PRODUCTS • HEALTHY PLANET

The Nature Conservancy

UC DAVIS UNIVERSITY OF CALIFORNIA

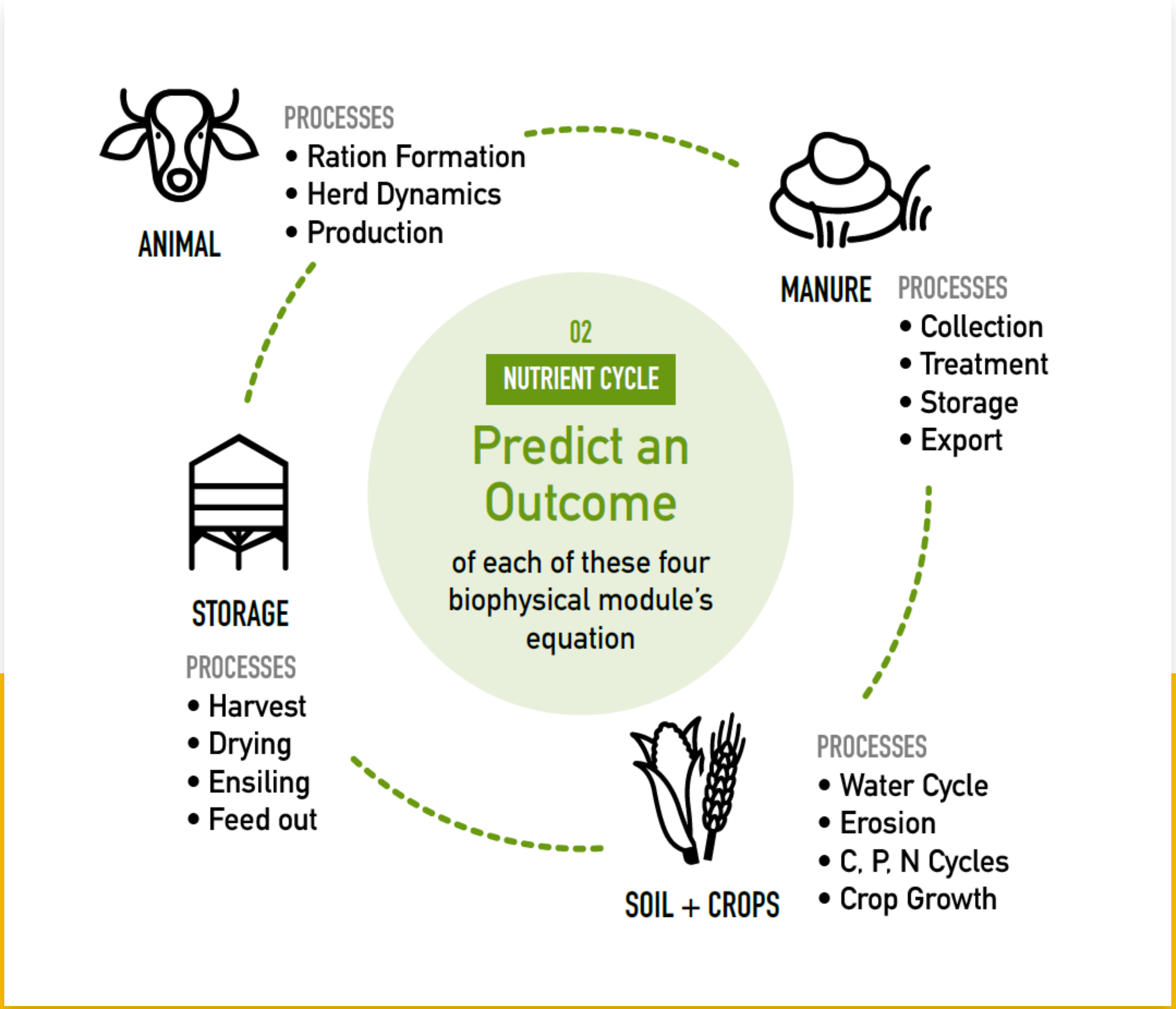
NEAFA Northeast Agribusiness & Feed Alliance Advocate · Educate · Collaborate

UNIVERSITY OF ARKANSAS

UNIVERSITY OF SOUTH DAKOTA

WWF

Biophysical Model



Soil & Crop: Version 1 Functionality

SOIL + CROPS

Management

- Tillage: from conventional to no-till
- Soil amendment: manure and fertilizer from broadcast to injection
- Flexible planting and harvest dates

Crops

- Corn (grain or silage)
- Alfalfa
- Grass
- Soybeans
- Wheat
- Rye
- Triticale
- Cover & Double Cropping

Outputs

- Emissions: N_2O, NH_3, CO_2
- Leaching and Runoff: N & P
- Water use
- Soil C stocks and changes
- Crop yields

Feed: Version 1 Functionality

STORAGE

Management

- Silage and Hay
- Storage separation by forage quality
- Inventory tracking
- Biomass loss during harvest, storage, and feedout
- Different feeds available to each animal group

Forage Composition Changes

- Dry matter content
- Total N and Non-protein N

Outputs

- CO_2 losses during harvest, storage and feedout
- Biomass left on field
- Spoilage/losses from storage and feedout
- Daily forage inventory

Animal: Version 1 Functionality

ANIMAL

Management

- Housing: Tie stall, freestall, drylot
- Reproduction: estrus detection, TAI, Re-synch
- Pens: variable number, size, stocking density and grouping methods
- Breeds: Holsteins and Jerseys

Diets

- Diets: 30-70% farm grown forage
- Purchased feed and by products
- Enteric methane mitigation supplements
- Least cost diet formulation OR User defined ration

Outputs

- Herd demographics
- Milk and meat production
- Enteric Methane production
- Embedded feed emissions
- Manure production and composition (N, P, VS,…)
- Energy, feed and water use

Manure: Version 1 Functionality

MANURE

Management Options

- Bedding: Sand or Organic
- Collection: Scraping or Flushing
- Processing: Solid-liquid separation, Anaerobic Digestion
- Storage: Lagoon, Composting, Bedded-pack, daily spread

Management Systems

- Allocated on a per pen basis
- Any combination of options (within reason)
- Milking parlor manure handled separately

Outputs

- Emissions: N_2O, CH_4, NH_3, CO_2
- Leaching and Runoff: N & P
- Water use
- Energy/Fossil Fuel use
- Soil C stocks and changes
- Crop yields

Energy Use: Version 1 Functionality

ENERGY

Management Options

- Manure collection methods
- Field operations
- Barn electricity use
  - Milk cooling
  - Heat abatement
- Water and manure pumps

Energy Production

- Anaerobic digestion CH_4 production
- (solar panel electricity production)

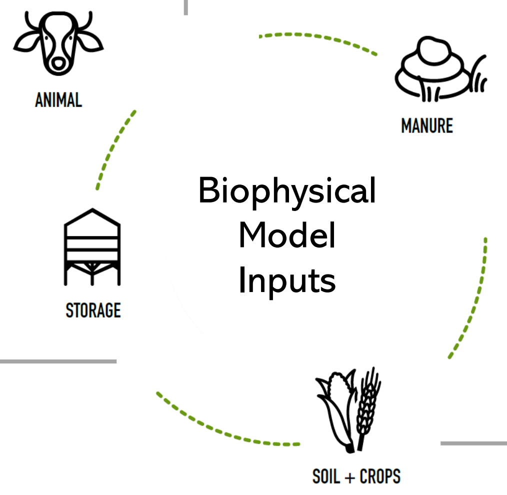
Outputs

- Gas and diesel use
- Electricity use
- Electricity or NG production

RuFaS Inputs

- Herd demographics
- Milk production
- Calf management
- Reproductive Program Management
- Avg. Birth weight and Mature Bodyweight
- Feeding Groups and Feeds or Diets

Farm Level Inputs
- Simulation length
- Weather
- Lat & Long
- Feed Composition Library



- Manure cleaning methods
- Bedding type
- Manure storage methods
    - Separation Methods
    - Treatment process
    - Long term storage methods
- Scenarios assigned to each pen separately

- Feeds used for different animal groups
- Feeds grown on farm
- Storage options used
- Purchased feeds

- Field Management
- Crops (and crop rotations)
- Field Fertilizer Practices
- Field Manure Practices
- Field Tillage Practices
- Field Soil Profiles

TWO DEVELOPMENT FOCI

ADDING + TESTING FUNCTIONALITY

Additional Management Practices

Input / output methods

Sensitivity Analyses

Precision & Accuracy Evaluation

IMPROVING CODE CLARITY + DOCUMENTATION

Automated testing

- Unit tests!

Refactoring large functions

Sphinx documentation

Progress in Model Documentation

Scientific Documentation

Coded in LateX or Rmarkdown – stored on designated repo folder and organized by module

```
### __Biomass allocation__

The `BiomassAllocation` class manages the crop biomass accumulation through the photosynthesis process and its partition between above and below ground organs during the growing season.

The central method, `allocate_biomass()`, calls on the **photosynthesize** and **partition biomass** methods to make daily updates on crop biomass allocation.

**Photosynthesize** converts the incoming solar radiation into plant biomass. First, potential plant growth is modeled by simulating **intercepted radiation** and
**maximum biomass growth**. Then, the latter is adjusted by plant stress to calculate the **biomass growth** on a given day and the **biomass** accumulated to date.

**Intercepted radiation** represents the amount of daily photosynthetically active radiation intercepted by the leaf area of the crop according to:

$$ R_{\text{int}} = 0.5\times R_{\text{inc}}\times (1-exp(-k_{\text{l}}\times A_{\text{leaf, i}}))
$$

where $R_{\text{int}}$ is the photosynthetically active radiation intercepted (`usable_light`), $R_{\text{inc}}$ is the total solar radiation available on a given
day (`incoming_solar_radiation`), $k_{\text{l}}$ is the light extinction coefficient (`light_extinction`), and $A_{\text{leaf, i}}$ is the leaf
area index on a given day (`leaf_area_index`).

**Maximum biomass growth** calculates the potential or upper-limit to total biomass increase on a given day that results from the **intercepted radiation** and the
crop-specific radiation-use efficiency, which is the amount of dry biomass produced per unit of intercepted solar radiation. It is calculated using the following
equation:
```

Biomass allocation

The `BiomassAllocation` class manages the crop biomass accumulation through the photosynthesis process and its partition between above and below ground organs during the growing season.

The central method, `allocate_biomass()`, calls on the **photosynthesize** and **partition biomass** methods to make daily updates on crop biomass allocation.

Photosynthesize converts the incoming solar radiation into plant biomass. First, potential plant growth is modeled by simulating **intercepted radiation** and **maximum biomass growth**. Then, the latter is adjusted by plant stress to calculate the **biomass growth** on a given day and the **biomass** accumulated to date.

Intercepted radiation represents the amount of daily photosynthetically active radiation intercepted by the leaf area of the crop according to:

$$R_{\text{int}} = 0.5 \times R_{\text{inc}} \times (1 - exp(-k_{\text{l}} \times A_{\text{leaf, i}}))$$

where R_{int} is the photosynthetically active radiation intercepted (`usable_light`), R_{inc} is the total solar radiation available on a given day (`incoming_solar_radiation`), k_{l} is the light extinction coefficient (`light_extinction`), and $A_{\text{leaf, i}}$ is the leaf area index on a given day (`leaf_area_index`).

Maximum biomass growth calculates the potential or upper-limit to total biomass increase on a given day that results from the **intercepted radiation** and the crop-specific radiation-use efficiency, which is the amount of dry biomass produced per unit of intercepted solar radiation. It is calculated using the following equation:

$$Growth_{\text{max}} = R_{\text{int}} \times Eff_{\text{light}}$$

In-line Documentation of Code Enables Future Coding Updates

```python
RUFAS: Ruminant Farm Systems Model
File name: manure_management.py

Author(s):  William Donovan, wmdonovan@wisc.edu
            Yunus Mohammed, ymm26@cornell.edu
            Sadman Chowdhury, skc86@cornell.edu
"""
import ...


class ManureManagement:
    """
    A class that sets up and manages different manure management components including manure handlers,
    reception pits, manure separators, and manure storage treatments. When the simulation engine performs
    a daily simulation, it invokes the update method on an instance of this class, thereby generating
    and storing daily output data.

    Notes:
        This class will replace the `ManureStorage` class.

    Attributes:
            manure_handlers: a dictionary that maps an animal pen's id to a ManureHandler object.
            reception_pits: a dictionary that maps an animal pen's id to a Recep
            manure_separators: a dictionary that maps an animal pen's id to a Man
            manure_treatments: a dictionary that maps an animal pen's id to a Tre

    """

    def __init__(self,
                 animal_management: AnimalManagement,
                 weather,
                 time,
                 manure_management_config):
        """Initializes a ManureManagement object by setting up the appropriate ma
        management components as specified by the data in the animal_management o

        Parameters
        ----------
        animal_management : AnimalManagement
            A reference to the AnimalManagement object that is one of the attribu
            of the simulation engine object.
        weather : Weather
            The Weather object used to initialize State variables.
        time : Time
            The Time object used to initialize State variables.
        manure_management_config : dict
            A dictionary that contains the configuration data for
            different manure management scenarios.

        """
        self.beddings: Dict[int, BaseBedding] = {}
        self.manure_handlers: Dict[int, BaseManureHandler] = {}
        self.reception_pits: Dict[int, ReceptionPit] = {}
        self.manure_separators: Dict[int, Optional[BaseManureSeparator]] = {}
        self.manure_treatments: Dict[int, BaseManureTreatment] = {}
        self.weather = weather
```

Ruminant Farm Simulation (RuFaS)

Search docs

CONTENTS:

- GitHub
- RUFAS package
  - Subpackages
    - RUFAS.output_handler package
    - RUFAS.routines package
  - Submodules
  - RUFAS.classes module
  - RUFAS.database_reader module
  - RUFAS.errors module
  - RUFAS.general_constants module
  - RUFAS.output_manager module
  - RUFAS.simulation_engine module
  - RUFAS.user_prompt module
  - RUFAS.util module
  - Module contents
- fileReader module
- main module
- setup module
- tests package

Submodules

RUFAS.routines.manure.manure_management module

RUFAS: Ruminant Farm Systems Model File name: manure_management.py

Description:

Author(s): William Donovan, wmdonovan@wisc.edu

Yunus Mohammed, ymm26@cornell.edu Sadman Chowdhury, skc86@cornell.edu

class `RUFAS.routines.manure.manure_management.ManureManagement`(*animal_management: AnimalManagement*)

Bases: `object`

A class that sets up and manages different manure management components including manure handlers, reception pits, manure separators, and manure storage treatments. When the simulation engine performs a daily simulation, it invokes the update method on an instance of this class, thereby generating and storing daily output data.

Notes:

This class will replace the *ManureStorage* class.

Attributes:

manure_handlers: a dictionary that maps an animal pen's id to a ManureHandler object.
reception_pits: a dictionary that maps an animal pen's id to a ReceptionPit object.
manure_separators: a dictionary that maps an animal pen's id to a ManureSeparator object.
manure_treatments: a dictionary that maps an animal pen's id to a Treatment object.

property `all_data`: *Dict[int, List[Tuple]]*

Returns all the data generated daily by different manure management components during the whole simulation.

Returns:

A dictionary that stores all the data generated daily by the four main manure management components. Its structure is as follows:

PROGRESS TOWARDS MODEL APPLICATIONS

DATA COLLECTION APP

Data Entry w/o Coding

Simple interface that will integrate directly with input manager

INPUT MANAGER

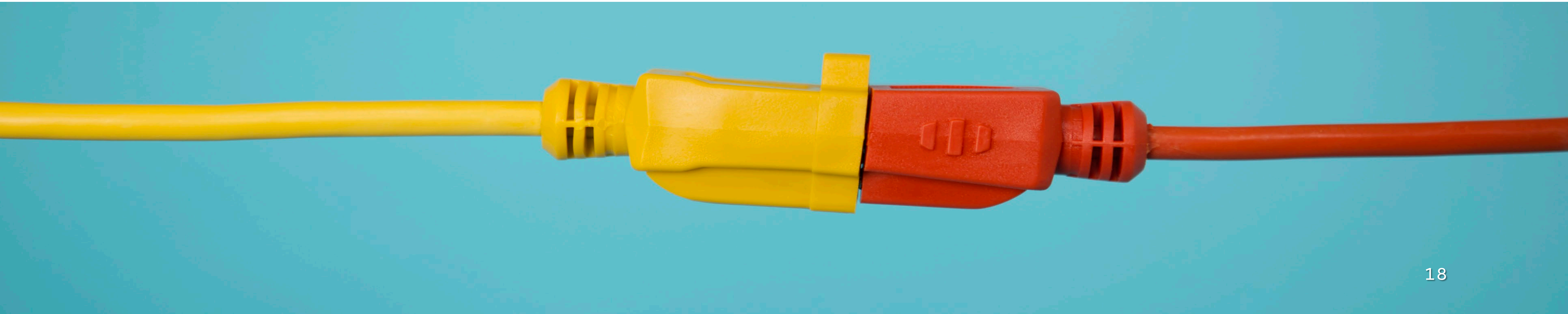
Single Source of Truth

All inputs flow through Input Manager. Checks and corrects inputs if needed.

OUTPUT MANAGER

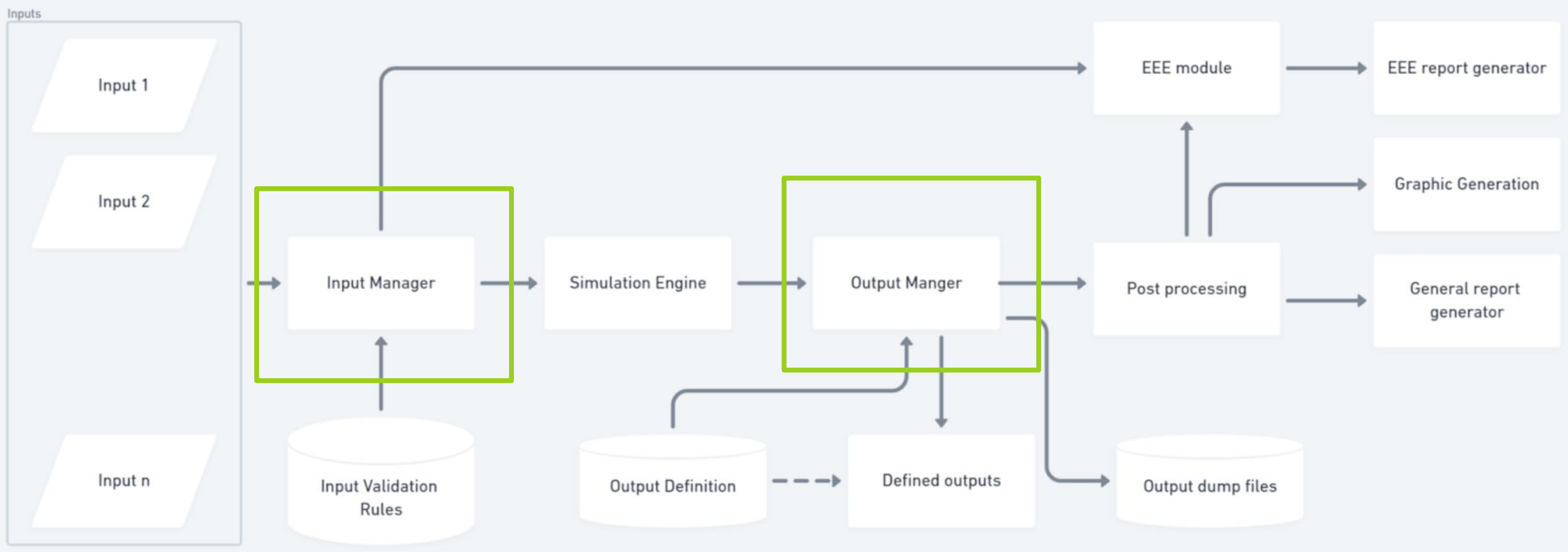
Complete Overhaul

New methods are flexible, testable, and **scalable**



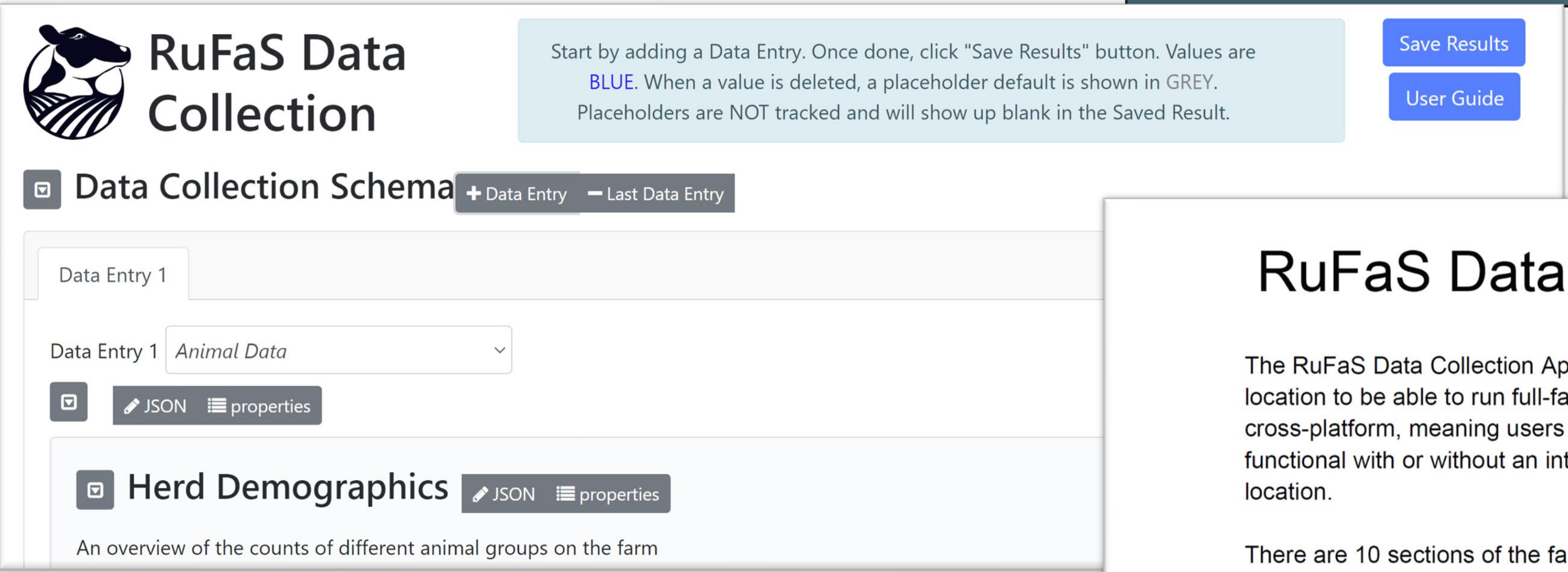
INPUT/OUTPUT MANAGERS UPDATE

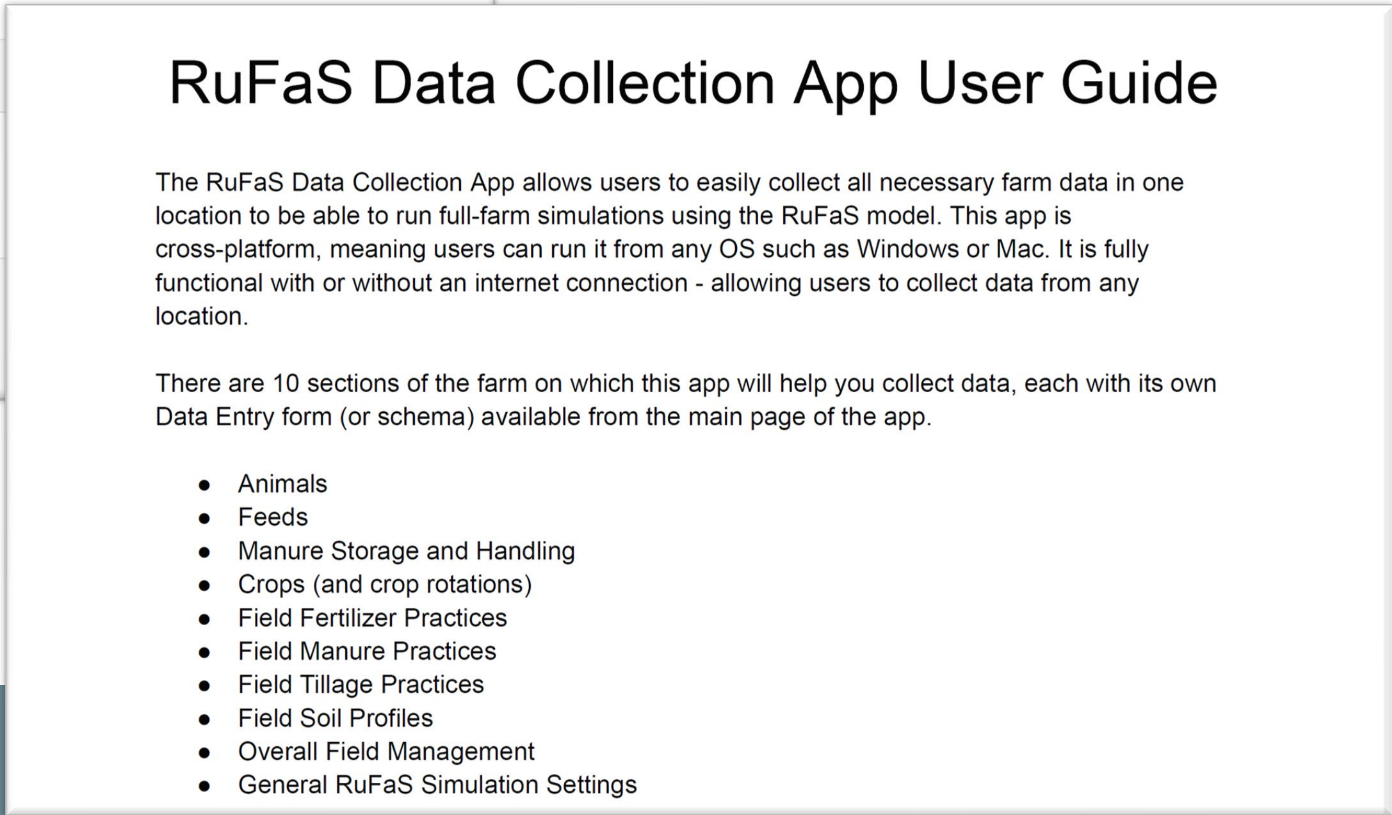
- All inputs flow to the simulation engine from a single source
- All outputs flow from the simulation engine in a pipeline



USER INPUTS TO MODEL INPUTS

- Data collection app provides user friendly way to input data, including documentation
- The input data and structure align with Input Manager metadata used for input validation





RuFaS Data Collection App User Guide

The RuFaS Data Collection App allows users to easily collect all necessary farm data in one location to be able to run full-farm simulations using the RuFaS model. This app is cross-platform, meaning users can run it from any OS such as Windows or Mac. It is fully functional with or without an internet connection - allowing users to collect data from any location.

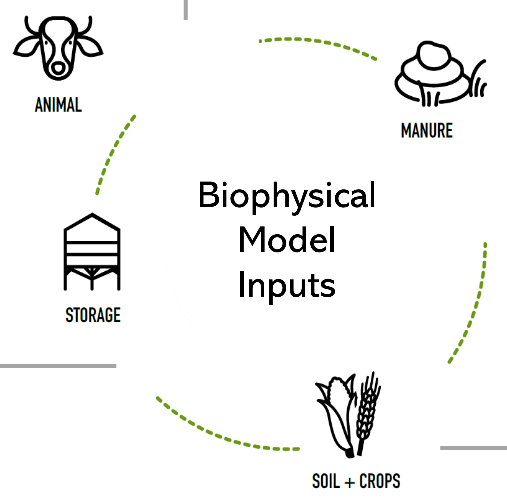
There are 10 sections of the farm on which this app will help you collect data, each with its own Data Entry form (or schema) available from the main page of the app.

- Animals
- Feeds
- Manure Storage and Handling
- Crops (and crop rotations)
- Field Fertilizer Practices
- Field Manure Practices
- Field Tillage Practices
- Field Soil Profiles
- Overall Field Management
- General RuFaS Simulation Settings

RuFaS Inputs

Farm Level Inputs
- Simulation length
- Weather
- Lat & Long
- Feed Composition Library

- Herd demographics
- Milk production
- Calf management
- Reproductive Program Management
- Avg. birth weight and Mature Bodyweight
- Feeding Groups and Feeds or Diets



- Manure cleaning methods
- Bedding type
- Manure storage methods
    - Separation Methods
    - Treatment process
    - Long term storage methods
- Scenarios assigned to each pen separately

- Feeds used for different animal groups
- Feeds grown on farm
- Storage options used
- Purchased feeds

- Field Management
- Crops (and crop rotations)
- Field Fertilizer Practices
- Field Manure Practices
- Field Tillage Practices
- Field Soil Profiles

RuFaS Evolution

2017

Formalized vision and established principles for RuFaS development

2018

Computer simulation structure established, and first lines of code written

Submitted first federal proposal

Expanded collaborators to include Cornell

2019

First paper published

Initiated documentation and version control methods

Submitted first successful Federal proposal

2020

Formalized Industry Advisory Council

Connected nutrient cycle in biophysical model

2021

Published 2 manuscripts on Animal Module

Hired first Professional Software Engineer

2022

Initiated contract with FARM-ES

2 additional manuscripts on Animal Module

2023

Announced Positions for 2 permanent USDA-ARS scientist positions for RuFaS

Preparing for V1 completion

Built Data Collection Application

Progress to Date

Funding Partners	
USDA Tech Transfer	$10,000
USDA-ARS / UVM Post Doc	$200,000
Jersey Association	$10,000
DMI	$1,200,000
General Mills	$300,000
NIFA IDEAS grant	$1,000,000
UW Madison Hatch Funding	$146,000
NEAFA	$100,000
Smith-Lever	$120,000
Atkinson Center	$180,000
USDA-ARS	$78,000
Zoetis	$50,000

Scholarship

Feature Article

A new modeling environment for integrated dairy system management

Ermias Kebreab,† Kristan F. Reed,‡ Victor E. Cabrera,§ Peter A. Vadas,# Greg Thoma,‖ and Juan M. Tricarico¶

animals MDPI

Article

The Ruminant Farm Systems Animal Module: A Biophysical Description of Dairy Cattle Management

Tayler L. Hansen [1], Manfei Li [2], Jinghui Li [3], C.J Vankerhove [1], M.A. Sotirova [1], Juan M. Tricarico [4], Victor E. Cabrera [2], Ermias Kebreab [3], and Kristan Reed [1,*]

J. Dairy Sci. 105:2180–2189
https://doi.org/10.3168/jds.2021-20817

The application of nonlinear programming on ration formulation for dairy cattle

J. Li,[1] E. Kebreab,[1] Fengqi You,[2] J. G. Fadel,[1] T. L. Hansen,[3] C. VanKerkhove,[4] and K. F. Reed[3*]
[1]Department of Animal Science, University of California, Davis 95616
[2]Robert Frederick Smith School of Chemical and Biomolecular Engineering, Cornell University, Ithaca, NY 14853
[3]Department of Animal Science, Cornell University, Ithaca, NY 14853
[4]School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853

15+ abstracts and conference proceedings

RuFaS Informs Decision-Makers

Extension Specialists

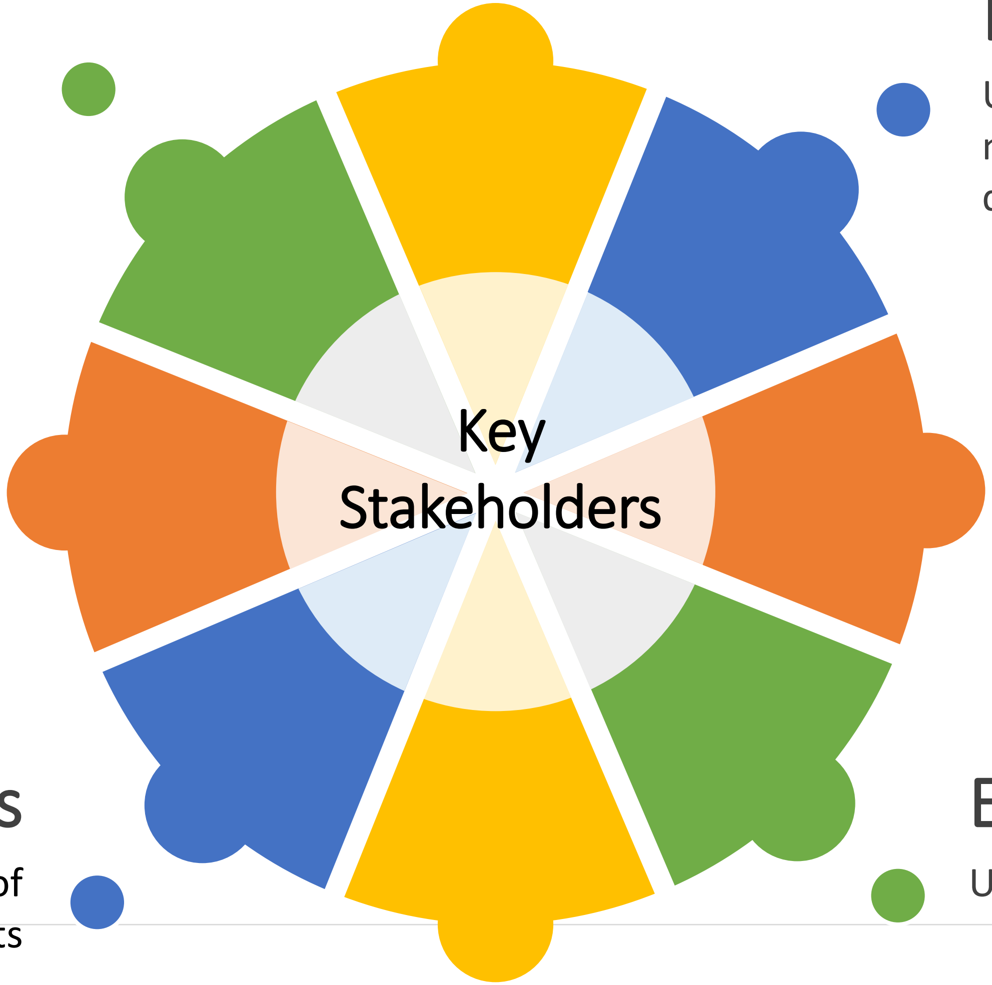
Use RuFaS to compare system impacts of proposed management practices before implementation

CAFO Planners

Use RuFaS to compare proposed management impacts on nutrient management plans before implementation

NGO Project Planners

Use RuFaS to compare system impacts of proposed projects



Farmers and Consultants

Use RuFaS to track progress of different management practices and inform future decisions

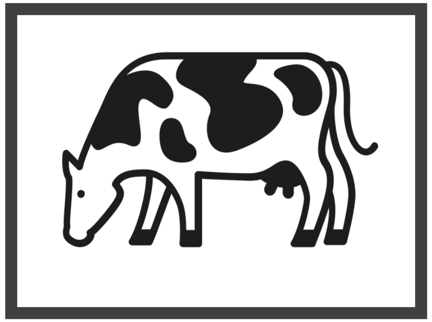
Dairy Processors

Use RuFaS to verify that claims meet company standards

Ecosystem Service Markets

Use RuFaS to quantify ecosystem services

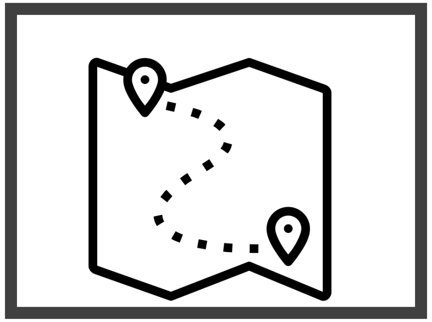
Vision of Success



Created by Rutmer Zijlstra
from Noun Project

Footprinting

Calculate baseline estimates of current farm outputs and environmental outcomes



Created by Aficons
from Noun Project

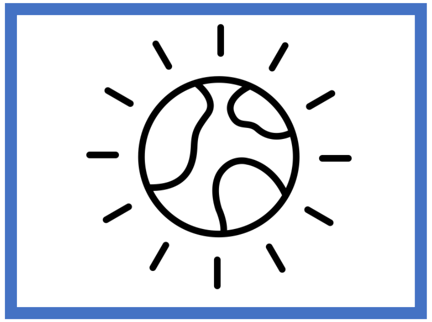
Planning

Identify management practices that will generate progress towards your sustainability goals

Created by mynamepong
from Noun Project

Implementation

Implement management plan, track progress, strive for continuous improvement



Created by Made x Made
from Noun Project

Impacts

Achieve industry-wide progress towards sustainable dairy production

NIFA AWARD # 2020-68014-31466

THANK YOU