Introduction to EEMS

Environmental Evaluation Modeling System v0.9e

An introduction to the principles of fuzzy logic, and an overview of the Environmental Evaluation Modeling System (EEMS) framework.

This material was adapted for the Utah_Colorado Plateau REA Stepdown Analysis from the Draft 2014 EEMS Manual v0.9e

James Strittholt, Tim Sheehan, Mike Gough, Rebecca Degagne
Conservation Biology Institute
8/1/2015

Table of Contents

I. Introduction

1.1 Purpose and Contents

This document is intended to help the reader understand how to use the EEMS Fuzzy Logic Modeling System.

How this document is organized

Section I provides a background on fuzzy logic, as well as a high level overview of the EEMS Modeling Framework. Section II describes the steps involved in developing and implementing an EEMS model. It is recommended that the user read through Sections I and II in order to obtain a conceptual understanding of the EEMS modeling process.

1.2 Software and Technical Requirements

Use of EEMS requires Microsoft Windows 7 or greater, ArcGIS 10.0 or 10.1, Spatial Analyst, and Python 2.6. While other versions of these software packages may work, they have not yet been tested.

The user should have formal GIS training, and be experienced with ArcGIS and ArcGIS Model Builder.

1.3 Overview of Fuzzy Logic

A logic model is a cognitive map (Jensen et al. 2009) that presents networks of various spatial data components and their logical relationships to explain the process used to evaluate a complex topic such as terrestrial intactness (Figure 1-1, Figure 1-2).

Logic models rely solely on spatial data layers that are arranged in a hierarchical fashion to answer a primary question that is located at the top of the diagram (Figure 1-1, Figure 1-2). In the example shown in this diagram, the evaluation pertains to the level of terrestrial landscape intactness. Data and analysis flows from the bottom up.

Unlike conventional GIS applications that use Boolean logic (1s and 0s) or scored input layers, logic models rely on fuzzy logic. Simply put, fuzzy logic allows the user to assign shades of gray to thoughts and ideas rather than being restricted to black (false) and white (true) determinations. All data inputs (regardless of the type—ordinal, nominal, or continuous) are assigned relative values between -1 (false) and +1 (true) up to six decimal places. There are many advantages of this modeling approach: (1) it is highly interactive and flexible; (2) it is easy to visualize thought processes; (3) the logic components are modular making it easy to include or exclude pieces of the logic design; (4) the logic can be managed using a number of different mechanisms; and (5) numerous, diverse topics can be included into a single integrated analysis. Raw spatial data source inputs (gold boxes) are populated by one or more GIS data layers (indicated by the stack of gray files). Moving up the diagram, these data are arranged and analyzed to form intermediate map products (purple boxes), which are then arranged and analyzed to generate the final results (green box). One way the user controls the logic of the information is the arrangement of the various data inputs and intermediate products—the higher up in the diagram, the greater the influence on the final result.

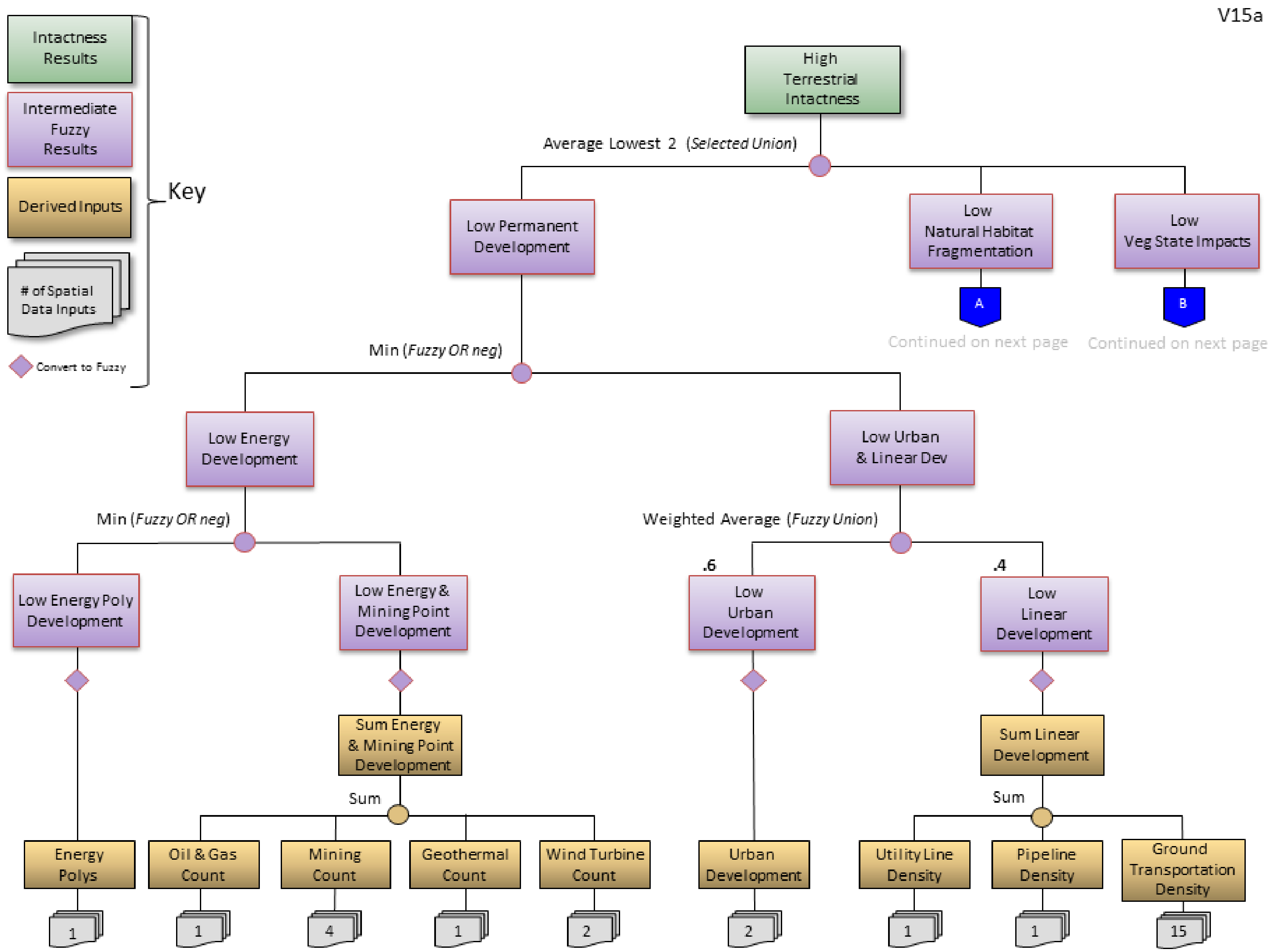


Figure 1-1. Logic model for terrestrial landscape intactness (Page 1 of 2)

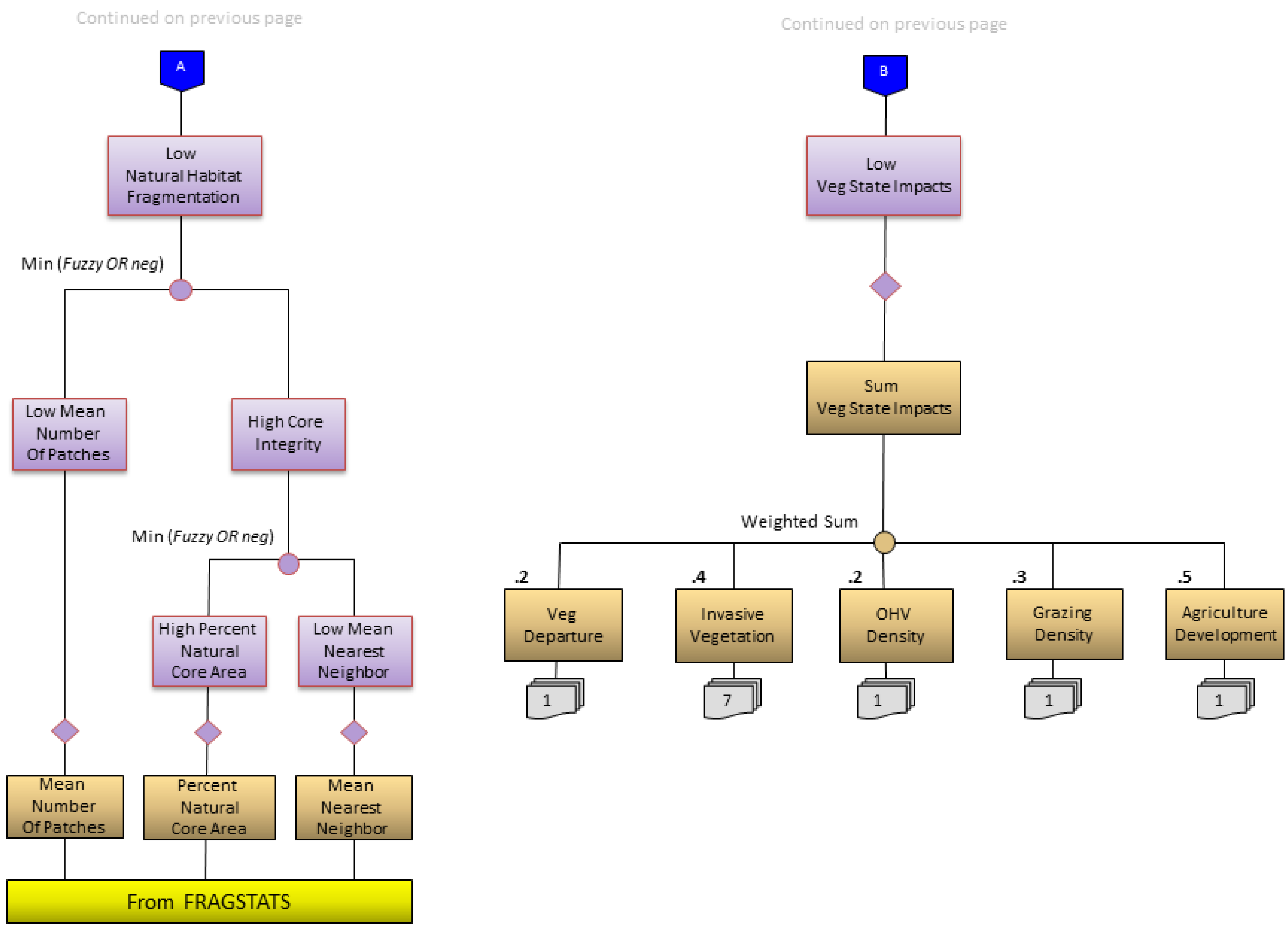


Figure 1-2. Logic model for terrestrial landscape intactness (Page 2 of 2)

Using fuzzy logic as the core modeling principle, logic model performance is achieved in several ways. For every spatial data input, the user determines how to assign the range of values along a truth continuum. When trying to determine and map the most suitable habitat from the standpoint of road density for wildlife—the greater the road density, the greater is the risk to wildlife through habitat degradation and direct mortality. In our example, road density ranges from 0 km/km2 to 24.5 km/km2. To assign a fuzzy logic continuum for this range of values, one could assign a -1 to the high value (this value is totally harmful for wildlife or false) and a +1 to the lowest value (this value is totally beneficial for wildlife, or true, red line in Figure 1-3). However, mountain lion research has shown that mountain lion populations have a low probability of persistence in areas with road densities > 0.6 km/km2 (Van Dyke et al. 1986). A more meaningful alternative then for setting fuzzy thresholds for this parameter would be that a road density of > 0.6 km/km2 is totally false (-1) and 0 remains totally true (+1, green line in Figure 1-3). Of course, not all wildlife species have the same sensitivity to roads, but this example illustrates how the logic in the model can be altered for known thresholds.

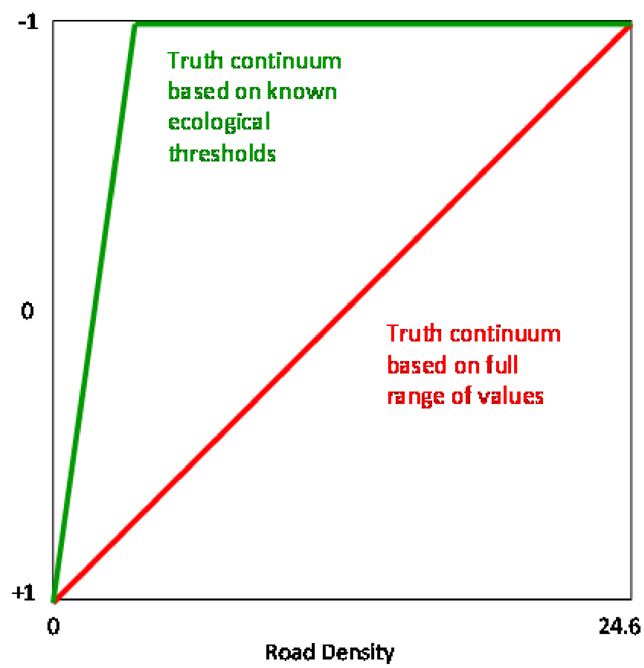


Figure 1-3. Diagram of two treatments of road density in fuzzy logic modeling illustrating important model control options, one based on a full range of values (red line) and the other based on a known threshold for road density (> 0.60 km/km^2 is totally false [-1], green line).

1.3 Overview of EEMS

The Environmental Evaluation Modeling System (EEMS) is a tree-based, fuzzy logic modeling system. With EEMS, data from different sources and different numerical domains can be combined to answer questions in areas such as current and future potential habitat value, ecological/development conflict, and landscape vulnerability to climate change.

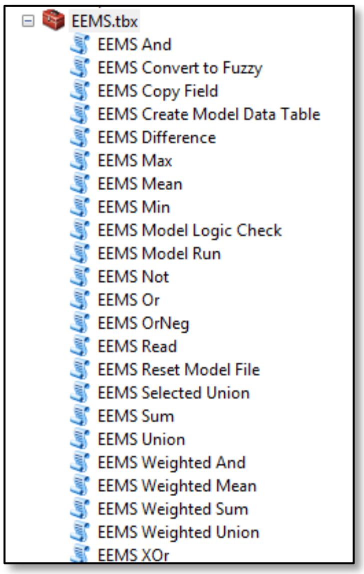
To use EEMS, a user builds a tree-based logic model in which the leaf nodes represent the initial data inputs. Initial data values are converted into fuzzy values (based on the premise that each input value can be represented by value ranging from -1 for fully false to +1 for fully true). Fuzzy logic operations (analogous to basic logic operations such as AND and OR) are used to combine nodes hierarchically until a final value representing the answer to a research question (e.g. What is the relative value of endangered species habitat across our study area?) is produced.

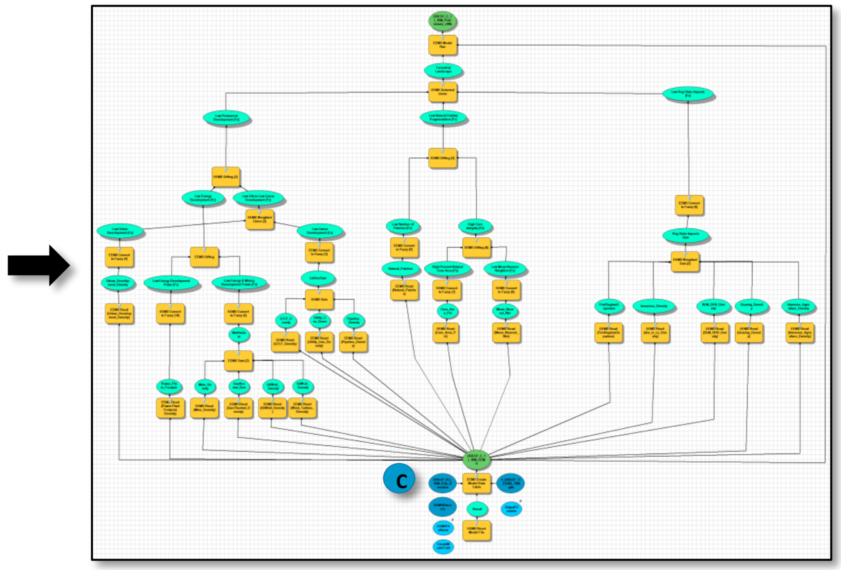
EEMS development is led by Tim Sheehan, a CBI ecological modeler, and has been applied by CBI scientists in a range of ecological evaluations. In the Tehachapis and Southern Sierra (see the hierarchical model below), a model incorporating data for habitat presence, habitat linkage, and disturbance was used to find areas of high ecological value and to provide guidance for reserve design to inform siting Wind Energy. For the Bureau of Land Management Rapid Ecological Assessments of the Sonoran Desert and Colorado Plateau ecoregions, several EEMS models were developed and used to evaluate a variety of current and projected ecological metrics.

1.4 Overview of the EEMS Fuzzy Logic Modeling Framework

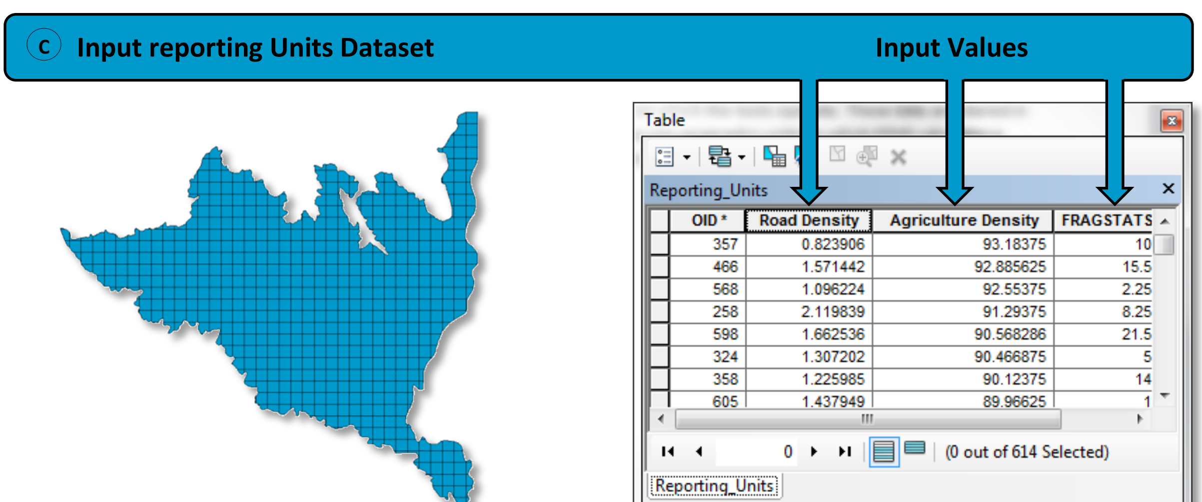
A **EEMS** is written in Python and deployed as a collection of script tools within an ArcGIS Toolbox.
B **EEMS models** are constructed by bringing these tools into an ArcGIS Model Builder model, where they are connected and arranged in a logical hierarchy.

A **EEMS** B **EEMS MODEL**





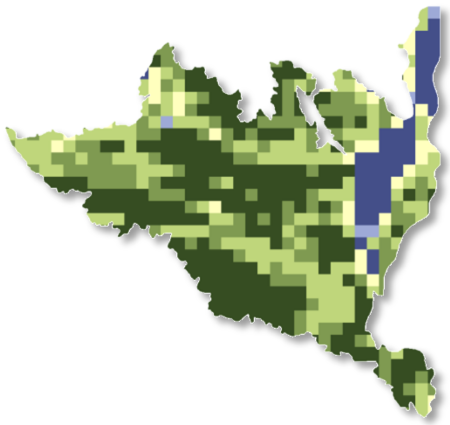
The third component of the EEMS modeling framework is the **Input Reporting Units Dataset** C on which the EEMS tools operate. All the values for the input variables (e.g., road density) need to be stored in a *single* vector dataset composed of reporting units. Reporting units are discrete geographic units containing quantitative measures of how much (or how little) of each variable is present within that reporting unit. Common measurements include densities, counts, and/or statistical measurements such as the mean or max. Reporting units may be regular geometric shapes (e.g., squares), or they may be in the form of irregular natural or political boundaries, such as watersheds or counties.



OID *	Road Density	Agriculture Density	FRAGSTATS
357	0.823906	93.18375	10
466	1.571442	92.885625	15.5
568	1.096224	92.55375	2.25
258	2.119839	91.29375	8.25
598	1.662536	90.568286	21.5
324	1.307202	90.466875	5
358	1.225985	90.12375	14
605	1.437949	89.96625	1

After the input reporting units dataset (C) has been created and the EEMS model (B) has been constructed, the model can be executed. When the model is executed, the EEMS tools (A) read in the values in the input reporting units dataset (C), the calculations are performed, and an **output reporting unit dataset** D is created which contains a final output value for each reporting unit.

D Output reporting Units Dataset **Output Values**



Table

EEMS_Output_Terrestrial_Intactness_4000m

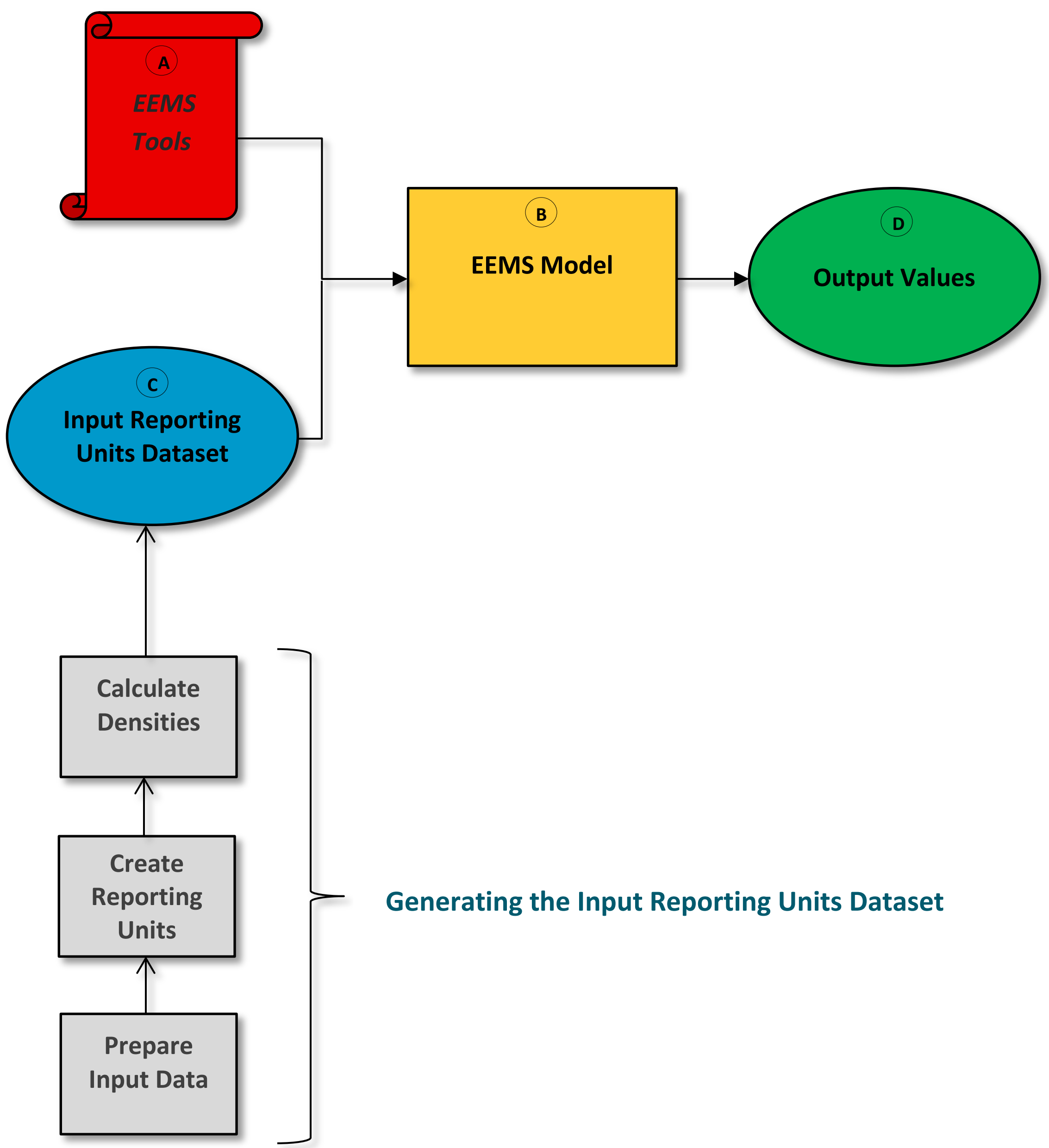
OID *	Road Density	Agriculture Density	FRAGSTATS	High Intactness
614	1.523997	38.059728	9	-0.522389
613	0.924152	0.143823	1	0.383899
612	1.294091	67.640625	1	-1
611	0.92066	1.86293	1	0.386226
610	1.53942	0	1	-0.02628
609	1.214107	81.781875	1	-1
608	1.337775	63.005625	1	-1
607	1.772449	16.689636	1	-0.181633
606	0	0	1	0.98

0 (0 out of 614 Selected)

EEMS_Output_Terrestrial_Intactness_4000m

THE FOUR MAIN COMPONENTS OF THE EEMS MODELING FRAMEWORK

The diagram below presents a high level overview of the four main components of the EEMS fuzzy logic modeling framework discussed above, and how these components are related. The gray boxes at the bottom of the diagram represent the steps that need to be performed in order to create the Input Reporting Units dataset. These steps are covered in the following section.



II. Implementation

2.1 Preliminary Steps

Before preparing your input data and constructing an EEMS model, it is important to perform the following preliminary steps:

1. Think about the question you are trying to answer.
2. Think about what information (raw data) you will need to acquire in order to answer that question.
3. Construct a conceptual model diagram describing how you might fit these pieces of information together in order to answer your question (Figure 2-1)

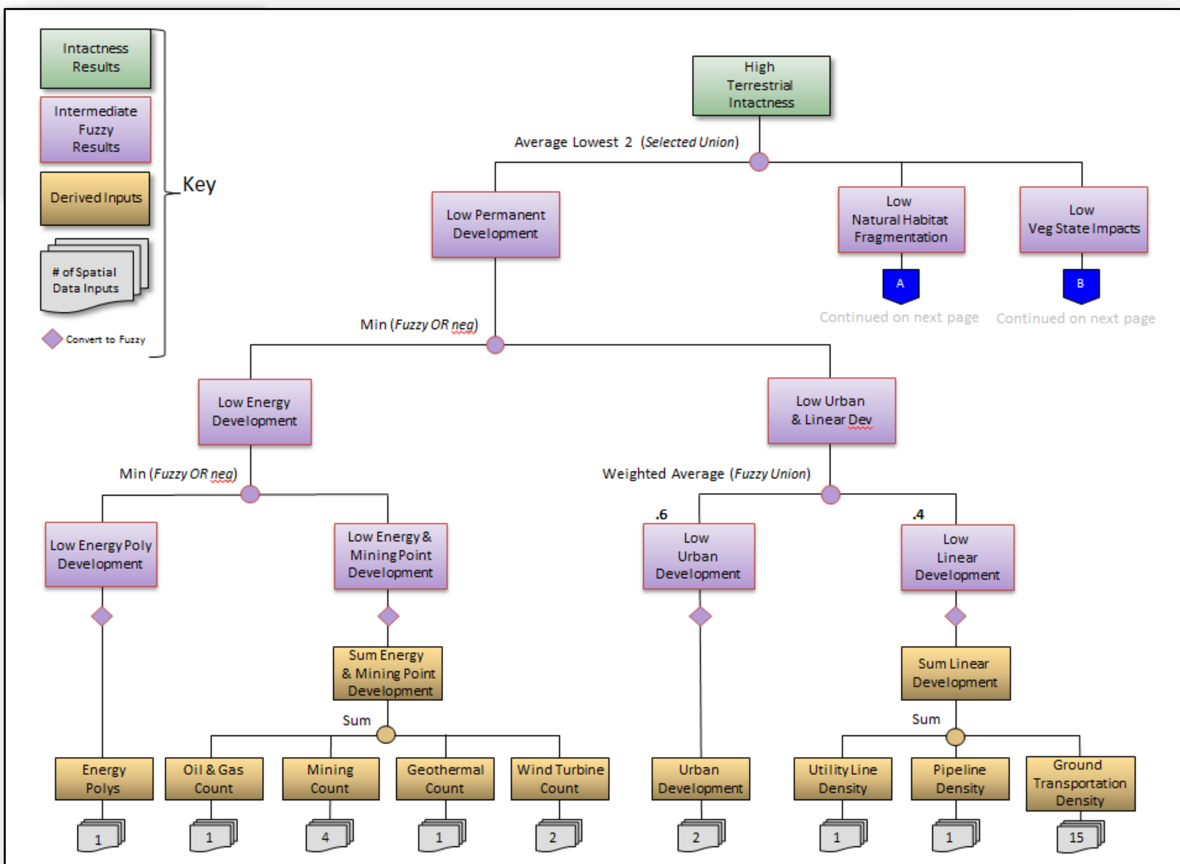


Figure 2-1: A conceptual model diagram (created in MS PowerPoint) describing the logic and operators to be used in constructing a Terrestrial Intactness logic model.

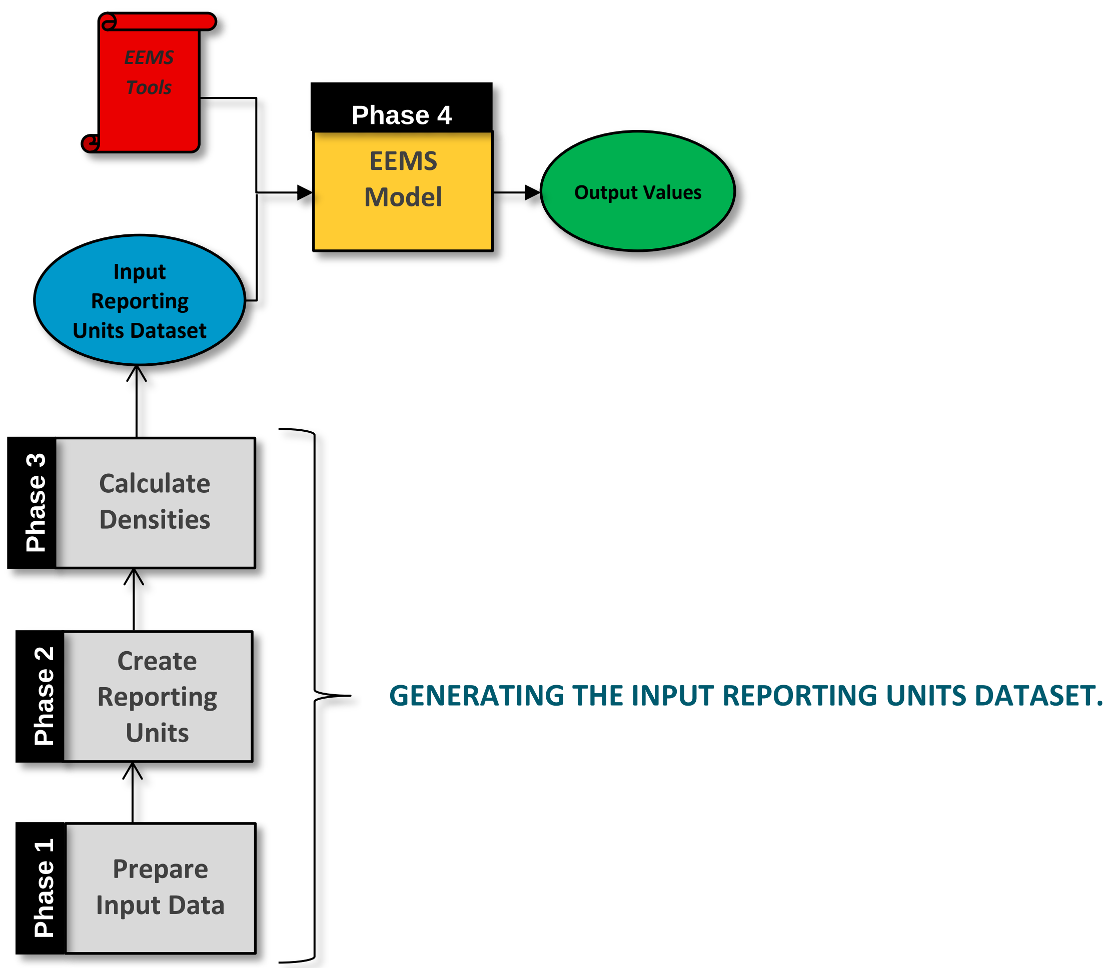
Once you've created a conceptual representation of your model, you can proceed with the EEMS modeling process.

2.2 The EEMS Modeling Process

There are typically four phases involved in the EEMS modeling process:

Phase 1: Preparing input data
Phase 2: Creating or defining reporting units
Phase 3: Calculating densities (Quantitative evaluation by reporting unit)
Phase 4: Constructing and executing an EEMS model

As discussed in section 1.4, EEMS operates on a single input dataset (shown in blue below). Each field in this dataset corresponds to an input variable, and each record corresponds to a reporting unit (discrete geographic units for which EEMS calculates a final output value). Consequently, in order to build and execute an EEMS model, one must first generate this master input reporting units dataset which contains all of the input data. This is accomplished in phases 1-3.



Implementation of phases 1-4 are generally carried out within the ArcGIS Model Builder environment using standard ArcGIS geoprocessing tools in conjunction with custom Python scripts.

The analyst has the option of constructing their own models and scripts to perform the tasks required of each phase, or, alternatively, he or she may use the EEMS support tools developed by CBI.

The following section describes the steps involved in the 4 phases of the EEMS modeling process.

PHASE 1 (PREPARING INPUT DATA):

During this phase of the analysis, you will need to define your study area and pre-processes your input data. The purpose of this phase is to ensure that all of the input data are ready to be quantitatively evaluated within each reporting unit.

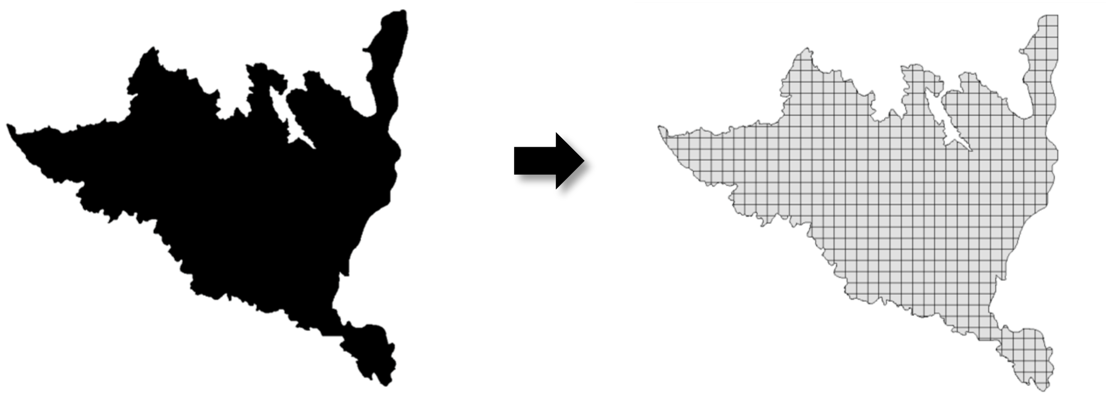
The process of preparing input data is unique to each analysis. Consequently, you will always need to construct your own models and scripts to conduct this phase of the process.

A common workflow for phase 1 is as follows:

1. Obtain or create a dataset which defines the study area.
2. Obtain the input data needed in order to answer the question under evaluation.
3. Project each dataset to a common projected coordinate system.
4. Select features of interest and generate new datasets as necessary.
5. Merge or combine existing datasets as necessary (e.g., roads from multiple counties).
6. Create points from x,y coordinates as necessary.
7. Dissolve input data as necessary.
8. Perform any necessary raster pre-processing
    a. **NOTE:** Raster datasets which will be used in in calculating percent cover by reporting unit should be classified to binary during this phase (1 for presence, 0 for absence).
9. Clip all of your input datasets to the study area.
10. Store all of your input data in a centralized location for input to phase 2.

PHASE 2 (CREATING OR DEFINING REPORTING UNITS):

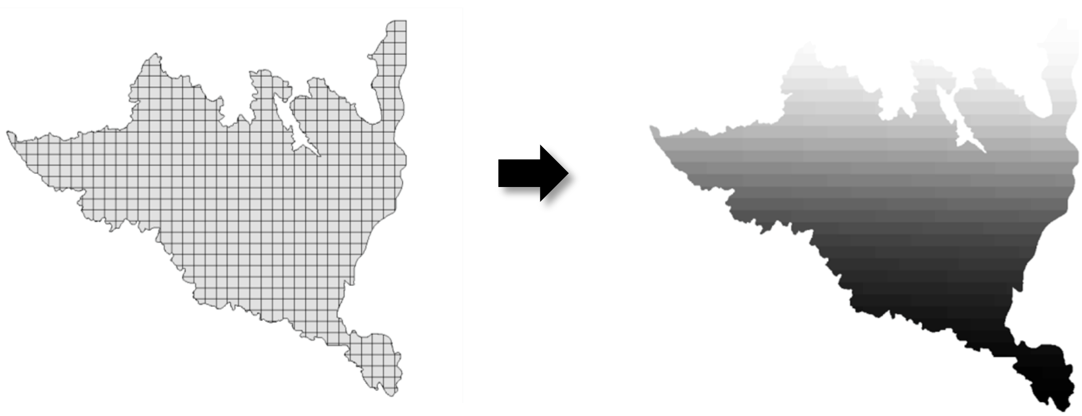
During phase 2, a reporting unit feature class is generated from the study area (which you defined in phase 1). The reporting units feature class is typically a vector based grid composed of regular square polygons (for example, $4km^2$ or $1km^2$). However, as mentioned previously, it may also be in the form of irregular natural or political boundaries, such as watersheds or counties.



The general procedure for creating the input reporting units dataset within ArcGIS is to:

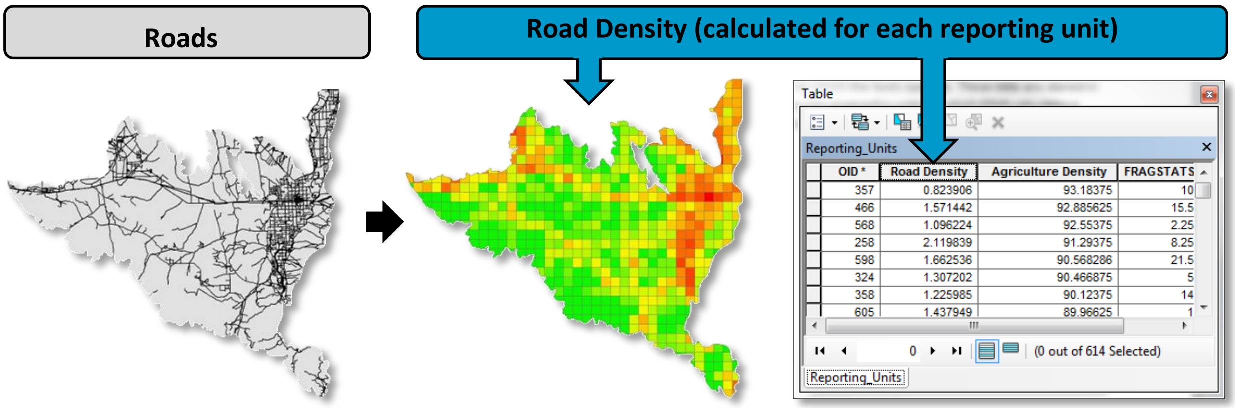
1. Use the **Fishnet** tool to create an empty reporting units dataset, setting the extent to match that of your study area boundary
2. **Clip** the fishnet to your study area boundary.

A corresponding raster version of the reporting units dataset should also generated for processing any raster data inputs. The raster version of the reporting units dataset is used for generating zonal statistics (e.g., mean vegetation departure within each reporting unit). The cell size of the raster reporting units dataset should match the resolution of your input raster datasets (although ESRI's zonal statistics tool will automatically perform a resampling if this is not the case). When the raster version of the reporting units dataset is created, the cell values generated must match the OBJECTIDs of the vector reporting units dataset within which they fall. The process of creating the raster version of the reporting units dataset is handled automatically if you use the CBI EEMS Model Template.



PHASE 3 (QUANTITATIVE EVALUATION, CALCULATING DENSITIES):

During this phase, the input datasets processed in phase 1 are quantitatively evaluated by reporting unit. That is, within each reporting unit, a value is calculated and stored which indicates *how much* (or how little) of each variable exists within that reporting unit. There are a number of steps involved in this phase. While there is no universal method of conducting this phase of the analysis, CBI has developed a set of tools, templates, and procedures to help automate many of the steps involved.



Common metrics calculated for vector data are as follows:

- **Points:** a count of the number of points within each reporting unit.
- **Lines:** the linear density within each reporting unit (e.g., km/km^2).
- **Polygons:** The percent of each reporting unit occupied by a polygon.

Common metrics calculated for raster data are as follows:

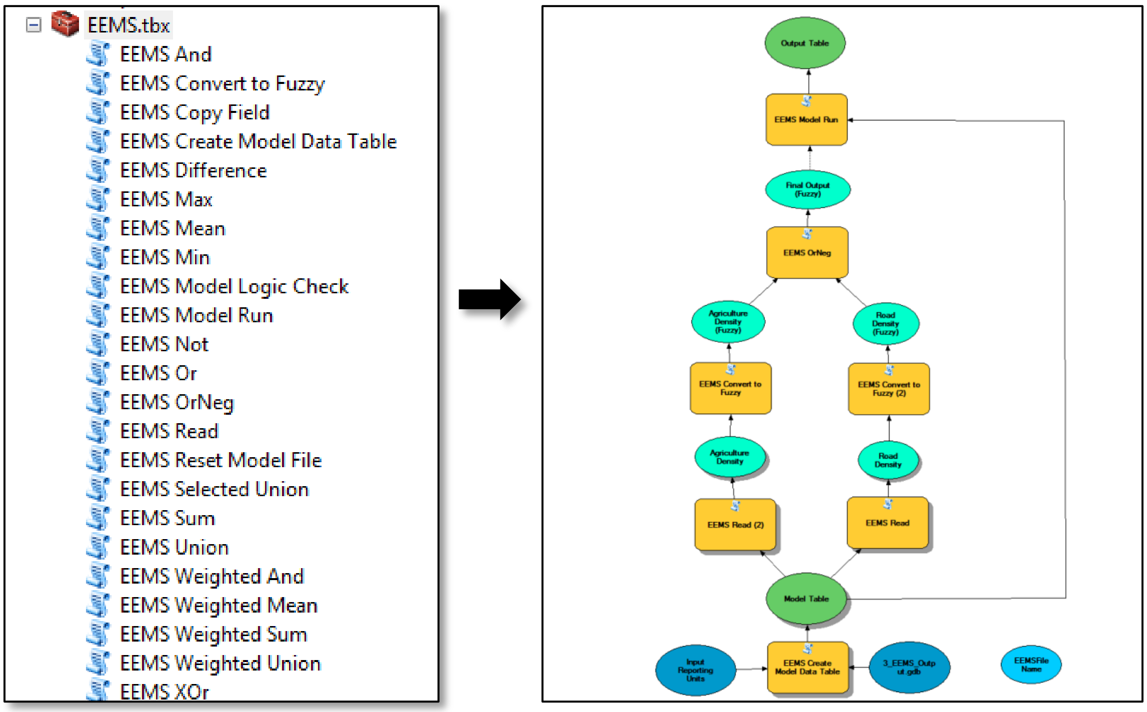
- Percent coverage within each reporting unit
- Statistical values (e.g., mean, max, min) within each reporting unit.

The general procedure for conducting this phase of the analysis is as follows:

1. Make a copy of the empty reporting units dataset created in phase 2.
2. Perform the necessary operations to calculate the desired metric within each reporting unit.
3. **NOTE:** This procedure can be conducted using the set of reporting unit tools developed by CBI. Once you have the desired metric (e.g., road density) stored in the copy of the reporting units dataset, join this dataset back to the master reporting units dataset based on the OBJECTID field.
4. Repeat steps 2 and 3 for each of your input datasets.

PHASE 4 (CONSTRUCTING AND EXECUTING AN EEMS MODEL):

The final phase of the process consists of constructing and executing your EEMS logic model. This is accomplished by dragging EEMS operators from the EEMS toolbox into your EEMS model and stringing them together in a logical hierarchy.



There are four core tools used in every EEMS model. These are described in the table below, and are typically connected and arranged as shown in the figure above:

OPERATOR	DESCRIPTION
EEMS CREATE MODEL DATA TABLE	Creates the output table with fields in sorted order. Input: The input Reporting Units dataset.
EEMS MODEL RUN	Runs a fuzzy model on a table. Inputs: Input table name Output table name Fuzzy model file name Output: Output table
EEMS READ	Reads a field.
EEMS CONVERT TO FUZZY	Finds the maximum for each row of the input fields.

Once the core EEMS tools have been incorporated into the EEMS model, the outputs can be combined using the logic operators listed and described in the table below. The operators used and the order in which they are executed will depend on the question under evaluation. Note that some of the operators are intended to operate on raw values while others are intended to operate on fuzzy values.

OPERATOR	INPUT DATA	DESCRIPTION
AND	Fuzzy	Finds the EMDS AND value of the inputs (maximum value). The formula is min + [(mean - min) * (min + 1) / 2]
CONVERT TO FUZZY	Raw	Converts a field's values into fuzzy values.
DIFFERENCE	Raw	Computes the difference sum for each row of the inputs.
MAX	Raw	Finds the maximum for each row of the input fields.
MEAN	Raw	Finds the mean for each row of the input fields.
MIN	Raw	Finds the minimum for each row of the input fields.
NOT	Fuzzy	Reverses the sign of values of the input field. TRUEness and FALSEness are swapped.
OR	Fuzzy	Returns the TRUEest of the inputs.
ORNEG (NEGATIVE OR)	Fuzzy	Returns the FALSEest of the inputs values.
SELECTED UNION	Fuzzy	Finds the union value (mean) of the specified number of TRUEest or FALSEest inputs.
SUM	Raw	Arithmetic addition of two or more inputs.
UNION	Fuzzy	Returns the mean of the inputs.
WEIGHTED AND	Fuzzy	Finds the weighted EMDS AND value of the inputs (maximum value). The formula is min + [(mean - min) * (min + 1) / 2] where the mean is weighted.
WEIGHTED MEAN	Fuzzy	Finds the weighted mean for each row of the input fields.
WEIGHTED SUM	Fuzzy	Finds the weighted sum for each row of the input fields. Multiplies each field by its weight before adding. Like a weighted mean without the division.
WEIGHTED UNION	Fuzzy	Finds the weighted union (mean) for each row of the input fields.
XOR	Fuzzy	Finds the fuzzy EXCLUSIVE OR value of the inputs by comparing the two truest values. If both are fully true or fully false, false is returned. Otherwise it applies the formula: (truest value - second truest value) / (full true - full false)

References Cited

Jensen, M., K. Reynolds, U. Langner, and M. Hart. 2009. Application of logic and decision models in sustainable ecosystem management. 2009. Proceedings of the 42nd Hawaii International Conference on Systems Sciences. Waikoloa, Hawaii. 5-8 January 2009.

Reynolds, K.M. 1999. NetWeaver for EMDS version 2.0 user guide: A knowledge base development system. U.S. Forest Service, General Technical Report PNW-GTR-471, U.S. Forest Service, Pacific Northwest Research Station, Portland, Oregon.