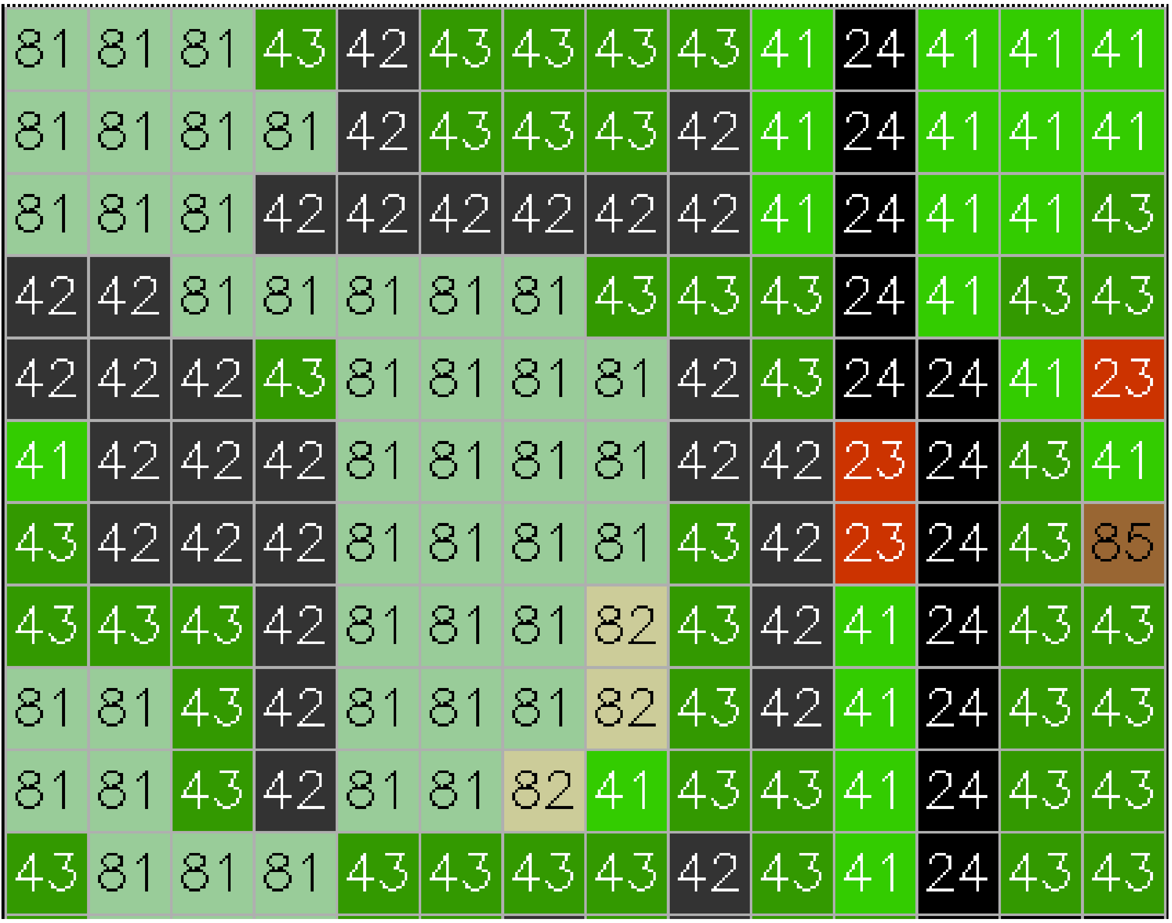
ERDC/CERL TR-04-24

US Army Corps
of Engineers®
Engineer Research and
Development Center

Development of Input Maps for the military Land use Evolution and Assessment Model (mLEAM) Land Use Change (LUC) Simulation Model

James D. Westervelt, Kyle Brock,
Woonsup Choi, and Yong Wook Kim

November 2004



81	81	81	43	42	43	43	43	43	41	24	41	41	41
81	81	81	81	42	43	43	43	42	41	24	41	41	41
81	81	81	42	42	42	42	42	42	41	24	41	41	43
42	42	81	81	81	81	81	43	43	43	24	41	43	43
42	42	42	43	81	81	81	81	42	43	24	24	41	23
41	42	42	42	81	81	81	81	42	42	23	24	43	41
43	42	42	42	81	81	81	81	43	42	23	24	43	85
43	43	43	42	81	81	81	82	43	42	41	24	43	43
81	81	43	42	81	81	81	82	43	42	41	24	43	43
81	81	43	42	81	81	82	41	43	43	41	24	43	43
43	81	81	81	43	43	43	43	42	43	41	24	43	43

Construction Engineering
Research Laboratory

ERDC/CERL TR-04-24
November 2004

Development of Input Maps for the military Land use Evolution and Assessment Model (mLEAM) Land Use Change (LUC) Simulation Model

James D. Westervelt

Construction Engineering Research Laboratory
PO Box 9005
Champaign, IL 61826-9005

Kyle Brock, Woonsup Choi, and Yong Wook Kim

University of Illinois
Urbana, IL

Final Report

Approved for public release; distribution is unlimited.

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Work Unit No. CS-1257

ABSTRACT: This document is intended to be used by GIS technicians in the development of input maps for the military Land use Evolution and Assessment Model (mLEAM) Land Use Change (LUC) simulation model. Military installations are facing growing challenges with adjacent incompatible land use patterns that limit the installation's ability to train soldiers and test equipment. Training and testing area associated with noise, dust, smokes, and radio transmissions that carry beyond an installations fenceline. Urban growth in these areas results in increasing potential for complaints from homeowners. The patterns of that urban growth are in direct response to past regional plans that involve specific investments and policies. Current plans and policies must be evaluated with respect to their direct, indirect, and cumulative impacts. The military version of the Land use Evolution and Assessment Model procedures described in this document provide an improved opportunity to evaluate proposed regional plans.

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.
DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

Contents

List of Figures and Tables

Figures

Tables

Conversion Factors

Non-SI[*] units of measurement used in this report can be converted to SI units as follows:

Multiply	By	To Obtain
acres	4,046.873	square meters
cubic feet	0.02831685	cubic meters
cubic inches	0.00001638706	cubic meters
degrees (angle)	0.01745329	radians
degrees Fahrenheit	(5/9) x (°F – 32)	degrees Celsius
degrees Fahrenheit	(5/9) x (°F – 32) + 273.15.	kelvins
feet	0.3048	meters
gallons (U.S. liquid)	0.003785412	cubic meters
horsepower (550 ft-lb force per second)	745.6999	watts
inches	0.0254	meters
kips per square foot	47.88026	kilopascals
kips per square inch	6.894757	megapascals
miles (U.S. statute)	1.609347	kilometers
pounds (force)	4.448222	newtons
pounds (force) per square inch	0.006894757	megapascals
pounds (mass)	0.4535924	kilograms
square feet	0.09290304	square meters
square miles	2,589,998	square meters
tons (force)	8,896.443	newtons
tons (2,000 pounds, mass)	907.1847	kilograms
yards	0.9144	meters

[*] *Système International d'Unités* ("International System of Measurement"), commonly known as the "metric system."

Preface

This study was conducted for the Strategic Environmental Research and Development Program (SERDP) Office under Work Unit CS-1257, “The Evolving Urban Community and Military Installations: A Dynamic Spatial Decision Support System for Sustainable Military Communities.” The technical monitor was Dr. Robert W. Holst, Compliance and Conservation Program Manager, SERDP. Mr. Bradley P. Smith is the Executive Director, SERDP.

The work was performed jointly by the Ecological Processes Branch (CN-N) of the Installations Division (CN), Construction Engineering Research Laboratory (CERL) and the University of Illinois’ Department of Urban and Regional Planning. The CERL Principal Investigator was Dr. James Westervelt. Kyle Brock, Woonsup Choi, and Yong Wook Kim are researchers associated with the University of Illinois, Urbana, IL. The UI Principal Investigator was Dr. Brian Deal. The technical editor was William J. Wolfe, Information Technology Laboratory (ITL), Champaign site, Stephen E. Hodapp is Chief, CEERD-CN-N, and Dr. John T. Bandy is Chief, CEERD-CN. The associated Technical Director is Dr. William D. Severinghaus. The Director of CERL is Dr. Alan W. Moore.

CERL is an element of the U.S. Army Engineer Research and Development Center (ERDC), U.S. Army Corps of Engineers. The Commander and Executive Director of ERDC is COL James R. Rowan, and the Director of ERDC is Dr. James R. Houston.

1 Introduction

Background

In the Department of Defense, encroachment refers to any activity that limits the ability to continue training and testing at military installations. Many encroachment challenges are related to the development of residences near military installations—a process called urbanization. Urbanization patterns develop in direct response to past regional investments in infrastructure (e.g., roads, sewer, electricity, and gas) and regional policies (e.g., zoning and tax incentives). The free-market system then allows people to compete for land and to develop it within the opportunities afforded by the local economy guided by the regional investments and policies. Therefore:

- The current opportunity to train and test is the result of past regional planning.
- Future opportunities to train and test will be the result of current and future investments and policies.

It is important that proposed investments and policies be evaluated with respect to the impact on future urban patterns and those patterns be evaluated with respect to changes in opportunities to conduct military training and testing.

The encroachment battleground involves debates over ill-defined property rights, a highly complex concept. Ownership of land is actually a set of property rights that contains many components—some of which are spelled out in legal documents and some of which are presumed. Perhaps the most visible component of land ownership is a fenceline on a legal boundary. However, some property rights may not extend to the fenceline while others can extend beyond it. Noise, dust, and smoke impacts from training can and do cross the fenceline legally because a landowner has the right to do so. For example, smoke from a resident's furnace or fireplace can leave the resident's property, as can noise or light. However, there are limits set in local, county, state, and Federal laws that must not be breached.

Objectives

The objective of this work was to provide installation planners with better tools to: (1) predict urban growth patterns that might occur in response to proposed regional plans involving certain investments and policies, and (2) predict the change in training and testing opportunity that might occur as a result of those patterns.

Approach

Prediction of urban patterns is accomplished using the Land Use Change (LUC) model, which is part of a suite of tools called the military Land use Evolution and Assessment Model (mLEAM). Researchers outlined the specific steps involved in preparing input maps for the mLEAM LUC model, with accompanying specific (case study) examples, and all data and supporting programming scripts.

Scope

This document provides information that will assist GIS technicians in the preparation of input maps for the mLEAM LUC model. GIS technicians will find general descriptions of the GIS processing steps that involve data acquisition to final development of mLEAM input maps and the description of the specific steps used to create maps for the multi-county area surrounding Fort Bragg, NC. All data and supporting scripts are provided with an accompanying CD. Appendix A contains a listing of the CD contents.

Mode of Technology Transfer

This report will be made accessible through the World Wide Web (WWW) at URL:

http://www.cecer.army.mil

2 Overview

Projecting Urban Growth to Predict Military Encroachment

Assume that you are a GIS technician assigned to predict urban patterns around a military installation over 20 to 50 years for the purpose of predicting future constraints on training and testing. The mLEAM model provides a set of tools and procedures for doing this in an optimal manner using available geographic information system (GIS) software and data. At the core of mLEAM is an urban growth model developed to efficiently and effectively predict urban footprints in a manner suitable for predicting future constraints to training and testing. There are many urban growth models; each is designed to answer a specific set of questions about that growth. It is useful to understand that the land use change (LUC) model at the core of mLEAM does not predict settlement patterns of different socioeconomic groups, predict the development of new roads and highways, or accurately predict the specific location of new growth. It does not take into account any information about the behavior of individual landowners, bankers, politicians, or voters.

The inputs to mLEAM are minimal. This way, the model provides general usefulness in a cost-effective way to a broad audience. The inputs are:

- raster GIS maps:
    - current land use
    - current roads and highways
    - no growth areas
    - employment centers (e.g., city centers) – location and size
- population
    - Current
    - Projections
- regional investments and policies
    - Changes in road network
    - New highways
    - New ramps to limited access highways
    - Change in no-growth areas.

Optional inputs involve a number of model calibration parameters.

The mLEAM LUC output is a time series showing changes in urban patterns. The LUC model predicts conversion of available undeveloped land to residential, commercial, and urban open space. Predicted urban patterns are then available for processing to identify changes in training and testing opportunities (which will be described in follow-up technical reports).

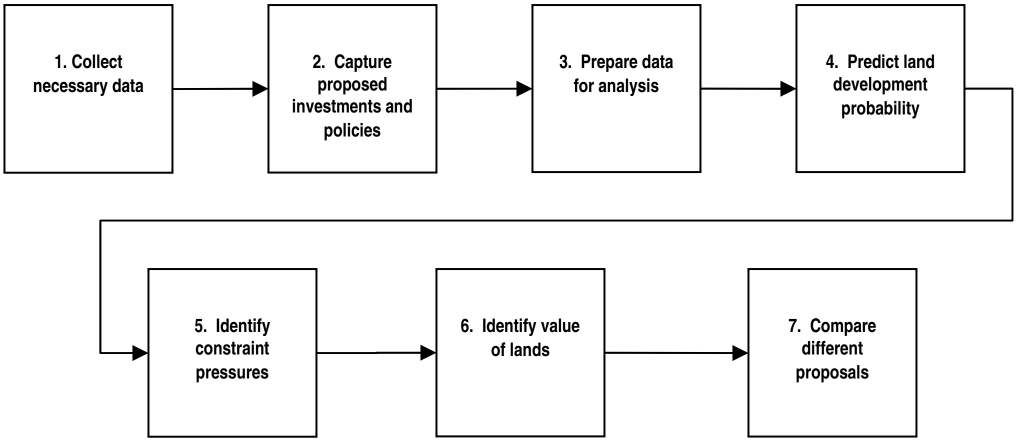

1. Collect necessary data
2. Capture proposed investments and policies
3. Prepare data for analysis
4. Predict land development probability
5. Identify constraint pressures
6. Identify value of lands
7. Compare different proposals

Figure 1. Conceptual diagram of complete mLEAM process.

The complete conceptual mLEAM approach (Figure 1) is comprised of seven steps:

1. Collect necessary data – maps and population projections.
2. Capture proposed investments and policies.
3. Prepare data for analysis.
4. Predict land development probability.
5. Identify constraint pressures on military training and testing.
6. Identify the value of land for training and testing (or carrying capacity).
7. Compare alternative regional policy/investment proposals (COAs).

Steps 1-4 are done mostly on a computer within an iterative framework described in Chapter 3. The result of step 4 will be one or more mLEAM graphical outputs. In steps 5-7, planners analyze and interpret the results of the mLEAM output(s) from step 4 to develop future plans for land in and around the military installation. Each of these seven steps is detailed in Chapter 3.

While conceptually simple, the procedures described in this document for conducting land use evolution studies are best implemented by GIS technicians and computer programmers.

The mLEAM LUC Algorithm

As discussed above, the LUC model urban growth projection takes a rather small set of user inputs and turns maps showing future urban growth patterns. The technical approach taken to process the inputs into the outputs is presented succinctly here and is intended primarily for managers and supervisors. The three gross steps are:

- acquire and prepare base maps (about 2 days)
- process maps using GIS into inputs for the LUC model (about 1 day)
- run the LUC model and visualize results (about 1 day per scenario).

The first step primarily involves acquiring nationally available maps from any of a number of specific government sources. mLEAM was designed to be generally useful and requiring only generally available information meets this goal, although locally available information can be used to provide more accurate inputs. Acquisition and preparation of the required maps can take less than a day; the sources and maps are presented in detail in later sections. This step also requires hands-on GIS processing to turn the acquired information into specifically formulated maps so that the automatic processing in the third step can be performed. Required processing involves putting all of the maps into a common equal-area projection, giving the maps specific names, and ensuring that the contents of the maps use specific category values. Acquisition and processing should take less than 2 solid working days with a modern GIS.

The second step involves running scripts that invoke specific GIS processing. Scripts are available for users of the ESRI Arc Info software on Windows-based PCs and for users of the GRASS GIS software on Unix/Linux machines. In this step, travel times are computed for each cell (location) to such things as the nearest small, medium, and large cities; water and forest; and roads, highways, and interstates. This step can take very little operator time although the computer processing of a multi-county area can involve several hours. Inspection of results is required and may result in the modification of input maps and parameters.

The third step involves running the LUC (land use change) model. In this step, the model, a configuration file, and the data prepared in the third step must be brought together. The model can take several hours to run a large area on a single processor computer. The model is written in C and is generally run on a Unix-based (e.g., Linux) computer. Output from the model can be taken back into a GIS for analysis and visualization.

The results of the processing can then be processed to understand the impacts of the predicted patterns on the opportunities for training and testing. At the time of this writing, procedures were being developed to automate this analysis and will appear in future reports.

The remainder of the information in this document is intended for GIS analysts. Instructions identify how to acquire data and software and how to process readily available data into urban growth projections for your area of interest.

3 mLEAM Procedures

While the conceptual steps given in Chapter 2 provide a theoretical framework by which to envision the mLEAM process, the actual GIS simulation deviates from a strict step-by-step method. In reality, the GIS (computer) portion of a mLEAM model is actually more of an iterative process than a step-by-step process, and this is consistent with the use of GIS as a practical tool. The computer portion of mLEAM can be seen as a separate but intertwined framework that runs on top of the conceptual framework given in Chapter 2.

The following procedures for generating urban growth predictions are to be followed by individuals with significant experience in the operation of geographic information systems. Today, GIS software makes different GIS functionality available to everyone from novices to experts. These procedures involve the ability to:

- Download maps from various web sites.
- Convert these to maps with specific projections, names, and categories.
- Read, write, and execute involved scripts.

Organizing and running the computer portion of a mLEAM model consists of a number of steps. While each of these steps is not inherently difficult, they do assume the user is familiar with a number of computer programs. Instructions are provided that assist in the use of ESRI's suite of software and the Unix/Linux based GRASS software. Users can choose to use one system or the other, but access to the strengths of both provides more flexibility. Detailed instructions for running the computer portion of mLEAM are included in the Appendices to this report.

The mLEAM computer framework, depicted in Figure 2, consists of manipulating raw data inputs into an overall land use projection, via two main transformation stages. The "manipulation" basically entails selecting germane information from a file and saving it in a readable computer format. First, raw data files are collected from various sources such as the Internet. Next, a GIS program such as Arc/Info or GRASS is used to assemble the raw data files into useable base files; this is done by hand. Then an automated GIS program takes the base files and manipulates them into Land Use Change (LUC) input files.

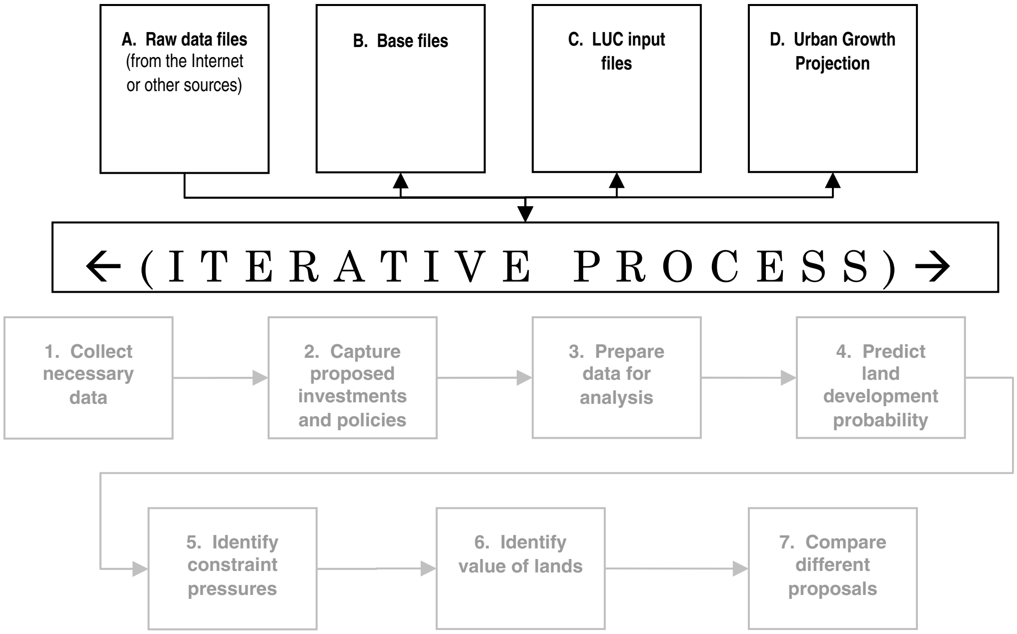

A. Raw data files (from the Internet or other sources)
B. Base files
C. LUC input files
D. Urban Growth Projection
← (ITERATIVE PROCESS) →
1. Collect necessary data
2. Capture proposed investments and policies
3. Prepare data for analysis
4. Predict land development probability
5. Identify constraint pressures
6. Identify value of lands
7. Compare different proposals

Figure 2. mLEAM computer framework.

Finally, the automated LEAM scripts convert the LUC input files into an overall land use projection. Due to the iterative nature of the GIS process, some of the intermediate or final maps may be reused as starter or intermediate maps for subsequent LUC input files.

Again, the computer steps A through D shown above are a way to envision the iterative nature of the GIS portion of mLEAM, and these steps are the realistic way of performing steps 1 through 4 of the conceptual framework outlined in Chapter 2. For example, a user may download a raw data file from the Web, manipulate it into a base file and then a LUC input file to get it ready for an urban growth projection. The user may then take this file and use it as one of the inputs for a different base file. There are more than a dozen related LUC input files to create, so there are many instances of re-using a common file to create new base files.

Still, the 7-step process holds for showing the mLEAM procedure in a conceptual, sequential manner. Figure 3 shows these steps.

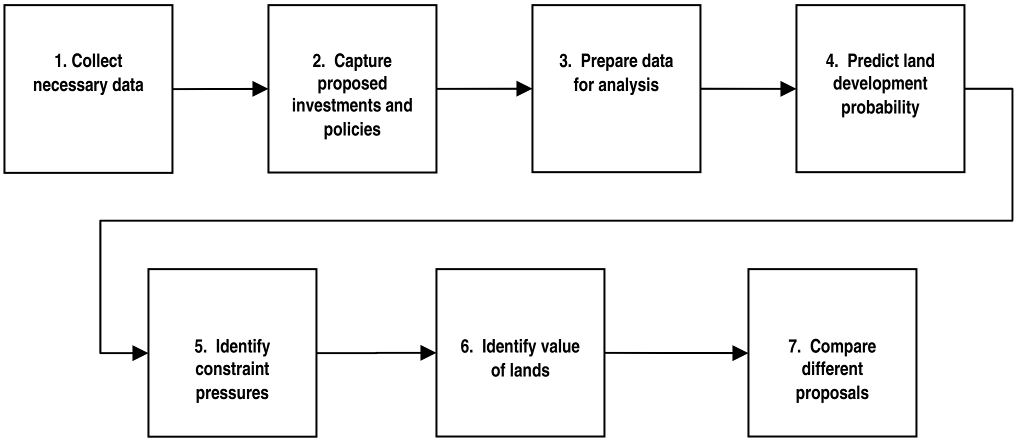

1. Collect necessary data
2. Capture proposed investments and policies
3. Prepare data for analysis
4. Predict land development probability
5. Identify constraint pressures
6. Identify value of lands
7. Compare different proposals

Figure 3. Complete mLEAM analysis steps.

1. Collect Necessary Data

The user will gather maps and other data from various sources (usually the Internet) for input into a GIS.

2. Capture Proposed Investments and Policies

The user will gather proposed investments and policies for the area, such as new highways or new zoning ordinances—usually available only through contacts in local or state governing boards. Since these things often affect the landscape, the user will create additional base maps that incorporate the proposed investments and policies. These additional maps will act as the basis for secondary, tertiary, etc. scenarios, so as to answer questions like "If this ___________ is built, how will the new development scenario compare to the 'business as usual' scenario?"

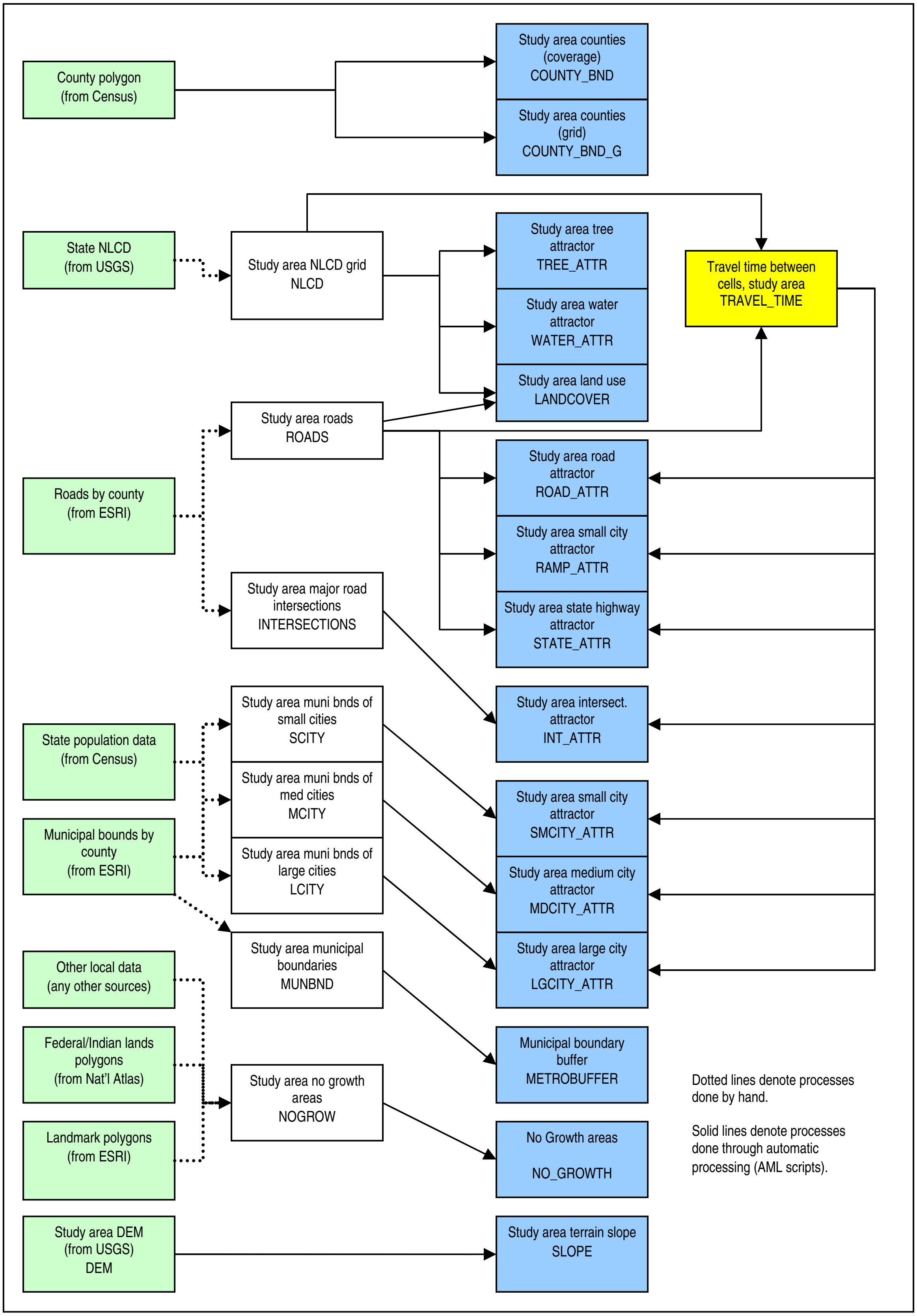

County polygon (from Census)
Study area counties (coverage) COUNTY_BND
Study area counties (grid) COUNTY_BND_G
State NLCD (from USGS)
Study area NLCD grid NLCD
Study area tree attractor TREE_ATTR
Study area water attractor WATER_ATTR
Study area land use LANDCOVER
Travel time between cells, study area TRAVEL_TIME
Study area roads ROADS
Roads by county (from ESRI)
Study area road attractor ROAD_ATTR
Study area small city attractor RAMP_ATTR
Study area state highway attractor STATE_ATTR
Study area major road intersections INTERSECTIONS
Study area intersect. attractor INT_ATTR
State population data (from Census)
Municipal bounds by county (from ESRI)
Study area muni bnds of small cities SCITY
Study area muni bnds of med cities MCITY
Study area muni bnds of large cities LCITY
Study area small city attractor SMCITY_ATTR
Study area medium city attractor MDCITY_ATTR
Study area large city attractor LGCITY_ATTR
Study area municipal boundaries MUNBND
Municipal boundary buffer METROBUFFER
Other local data (any other sources)
Federal/Indian lands polygons (from Nat'l Atlas)
Landmark polygons (from ESRI)
Study area no growth areas NOGROW
No Growth areas NO_GROWTH
Study area DEM (from USGS) DEM
Study area terrain slope SLOPE
Dotted lines denote processes done by hand.
Solid lines denote processes done through automatic processing (AML scripts).

Figure 4. Converting raw maps to mLEAM input maps.

3. Prepare Data for Analysis

As Figure 4 shows, the user will select pertinent information from the raw data and put it into a computer format useable by the mLEAM software.

There are two methods for creating the mLEAM input maps: (1) using ESRI ArcGIS exclusively, or (2) using ESRI ArcGIS and GRASS GIS. The advantage of using ESRI ArcGIS exclusively is that the user can perform all operations with one program on Windows. The advantage of using ESRI ArcGIS in conjunction with GRASS is that GRASS has more capabilities for handling raster data. This method requires the use of ESRI ArcGIS very briefly in the initial stages of the mapmaking process, since the downloaded files usually come in ESRI Shapefile format. This method also requires access to both a Windows machine and a UNIX machine.

Create Land Use Change Input Maps

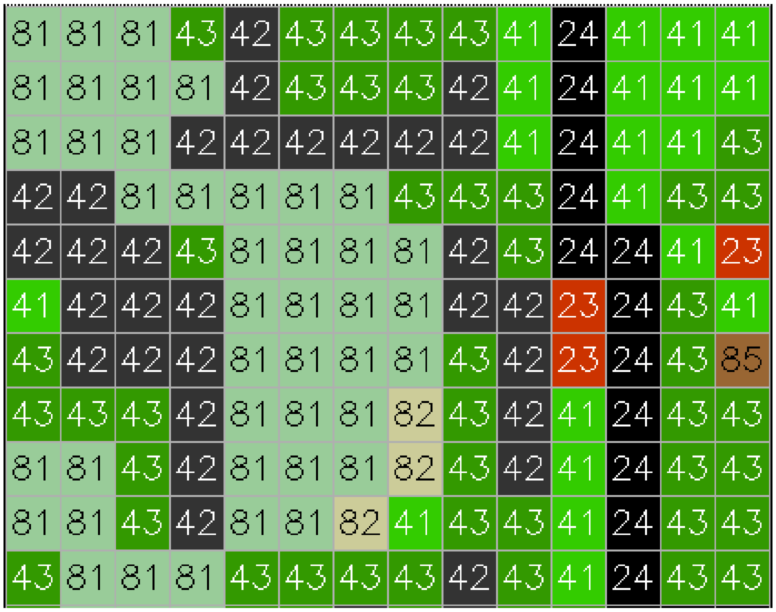
The goal is to create several specific Land Use Change input maps (Table 1), which are described in the following sections.

Table 1. Land use change input maps.

Map	Description
Boundary	Boundary of study region (usually a group of counties)
Land Cover	Land cover of study region (NLCD data)
Metrobuffer	Buffer around municipal boundaries in study region (typically 1.5 miles)
No Growth	Areas in study region in which development is not possible or highly unlikely
Slope	Slope of terrain in study region
Forest Attractor	Distance to nearest forest
Water Attractor	Distance to nearest water body
Highway Attractor	Travel time to nearest highway
Intersection Attractor	Travel time to nearest major road intersection
Ramp Attractor	Travel time to nearest highway ramp
Road Attractor	Travel time to nearest road
Cities Attractor (large, medium, small)	Travel time to nearest city (divided into three groups—small, medium, and large)

Boundary. This map simply indicates the boundaries of LEAM simulation runs. Each grid cell in which the model will be run is indicated with the number 1 and all other cells are given the value of zero.

Land Cover. The National Land Cover Data (NLCD) classification system is used. Figure 5 shows the raster map showing color-coded categories below use the NLCD category definitions.



81	81	81	43	42	43	43	43	43	41	24	41	41	41
81	81	81	81	42	43	43	43	42	41	24	41	41	41
81	81	81	42	42	42	42	42	42	41	24	41	41	43
42	42	81	81	81	81	81	43	43	43	24	41	43	43
42	42	42	43	81	81	81	81	42	43	24	24	41	23
41	42	42	42	81	81	81	81	42	42	23	24	43	41
43	42	42	42	81	81	81	81	43	42	23	24	43	85
43	43	43	42	81	81	81	82	43	42	41	24	43	43
81	81	43	42	81	81	81	82	43	42	41	24	43	43
81	81	43	42	81	81	82	41	43	43	41	24	43	43
43	81	81	81	43	43	43	43	42	43	41	24	43	43

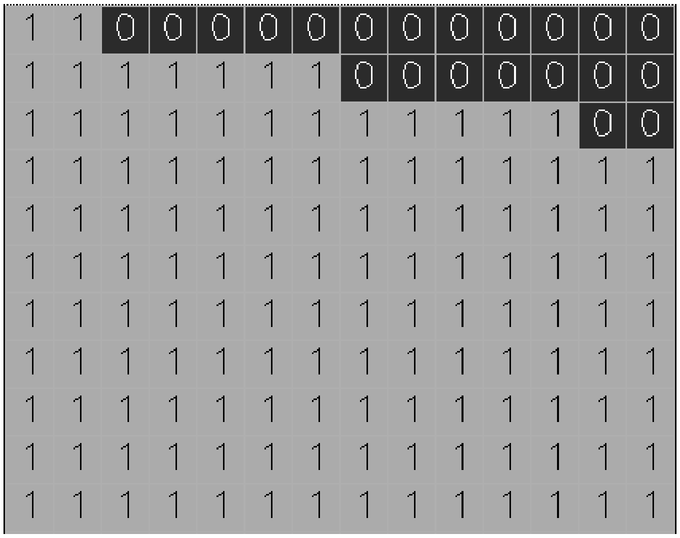
Figure 5. Sample land cover map.

The NLCD category numbers represent land cover as follows:

11. Open Water
12. Perennial Ice/Snow
21. Low Intensity Residential
22. High Intensity Residential
23. Commercial/Industrial/Transportation
31. Bare Rock/Sand/Clay
32. Quarries/Strip Mines/Gravel Pits
33. Transitional
41. Deciduous Forest
42. Evergreen Forest
43. Mixed Forest
51. Shrub land
61. Orchards/Vineyards/Other
71. Grasslands/Herbaceous
81. Pasture/Hay
82. Row Crops
83. Small Grains
84. Fallow
85. Urban/Recreational Grasses

Further details on these definitions can be found at URL:
http://landcover.usgs.gov/classes.html

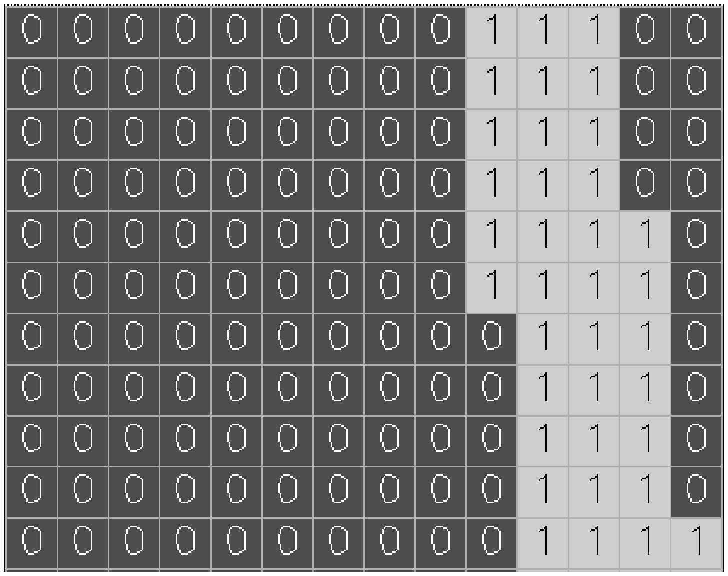
Metrobuffer. Cells defined as within a growth sphere of influence are indicated with a 1, others are zero (Figure 6). This map can capture the extent of the anticipated legal boundaries of cities over the course of a LEAM simulation.



1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 6. Sample metro buffer map.

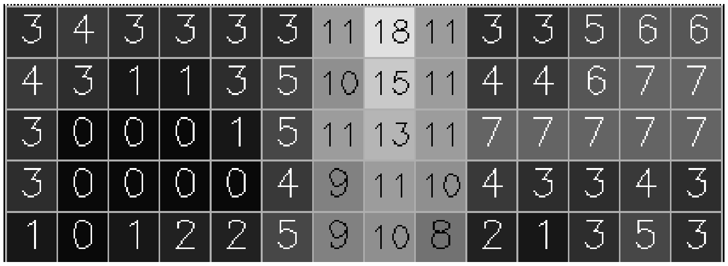
Nogrowth. The nogrowth map is also a one-zero map. Cells that cannot grow are indicated with the value of 1. In the sample map below, the cells coded with a "1" are associated with interstate highway rights of way. Other examples of no-growth cells are those associated with water, no-growth zones, parks, forests, nature preserves, etc. This map (Figure 7) is used to capture many alternative land management policies (e.g., zoning) and investments (e.g., purchase of property and/or property rights.)



0	0	0	0	0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1

Figure 7. Sample no growth map.

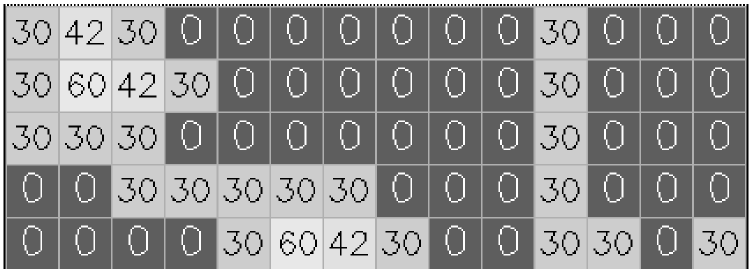
Slope. The slope map (Figure 8) indicates the slope of the cell in degrees. The slope map is used in the LEAM process because the likelihood of development is generally higher in areas with gentle slopes and generally lower in areas with steep slopes.



3	4	3	3	3	3	11	18	11	3	3	5	6	6
4	3	1	1	3	5	10	15	11	4	4	6	7	7
3	0	0	0	1	5	11	13	11	7	7	7	7	7
3	0	0	0	0	4	9	11	10	4	3	3	4	3
1	0	1	2	2	5	9	10	8	2	1	3	5	3

Figure 8. Sample slope map.

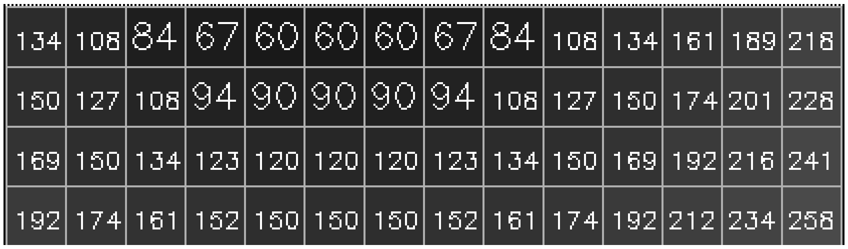
Forest_attractor. The forest attractor is simply the straight-line distance to the nearest cell identified as having forest in the land cover map. Cells containing forest are coded as "0" (zero distance to the nearest forest). The cell size the map shown in Figure 9 is 30-meters and the distances to nearby cells containing forest is obviously recognizing this cell size. The forest attractor map is used in LEAM because the existence of trees makes development more attractive.



30	42	30	0	0	0	0	0	0	0	30	0	0	0
30	60	42	30	0	0	0	0	0	0	30	0	0	0
30	30	30	0	0	0	0	0	0	0	30	0	0	0
0	0	30	30	30	30	30	0	0	0	30	0	0	0
0	0	0	0	30	60	42	30	0	0	30	30	0	30

Figure 9. Sample forest attractor map.

Water_attractor. The water attractor map is identical in concept to the forest attractor. The sample map portion shown in Figure 10 indicates a body of water above the area just to the left of center. The water attractor map is used in LEAM because the existence of water makes development more attractive.



134	108	84	67	60	60	60	67	84	108	134	161	189	218
150	127	108	94	90	90	90	94	108	127	150	174	201	228
169	150	134	123	120	120	120	123	134	150	169	192	216	241
192	174	161	152	150	150	150	152	161	174	192	212	234	258

Figure 10. Sample water attractor map.

Highway_attractor. This and the following three maps identify the driving time in minutes to the nearest state highway, intersections of open roads, limited-access highway ramps, and main roads. The driving time is calculated with an algorithm that takes into account speed limits for roads, as defined by the user. The creation of these maps relies on the development of a cell-traversal travel time map. Then, GIS analyses that identify the minimum travel time to the respective resources are conducted to create these four maps. In the images below, black represents zero travel time and the whiter the cell, the greater the travel time. The pure white areas indicate no-growth locations (water, Federal lands, etc.) upon which travel is near impossible; hence, travel times are arbitrarily set extraordinarily high. The highway attractor map (Figure 11) is used in LEAM because development is more likely near a highway.

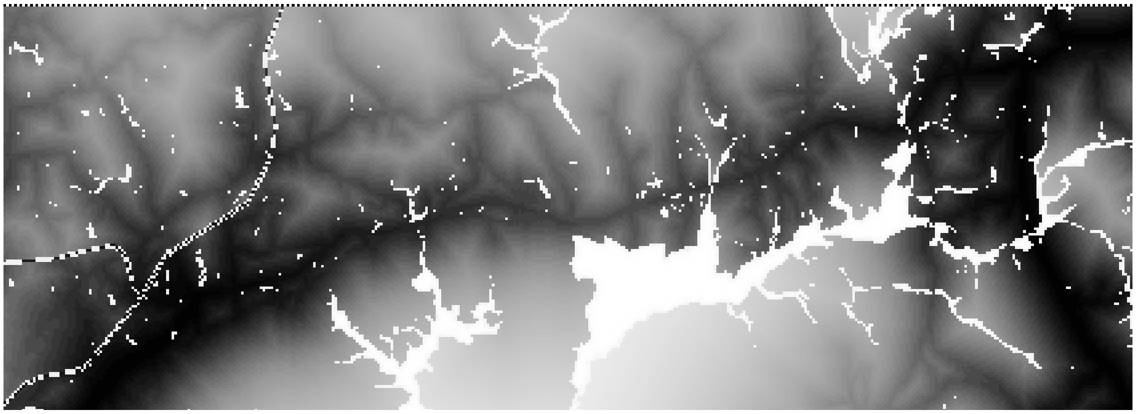


Figure 11. Sample highway attractor map.

Intersection_attractor. The map shown in Figure 12 depicts the travel time to the nearest highway intersection. The intersection attractor map is used in LEAM because development is more likely near highway intersections.

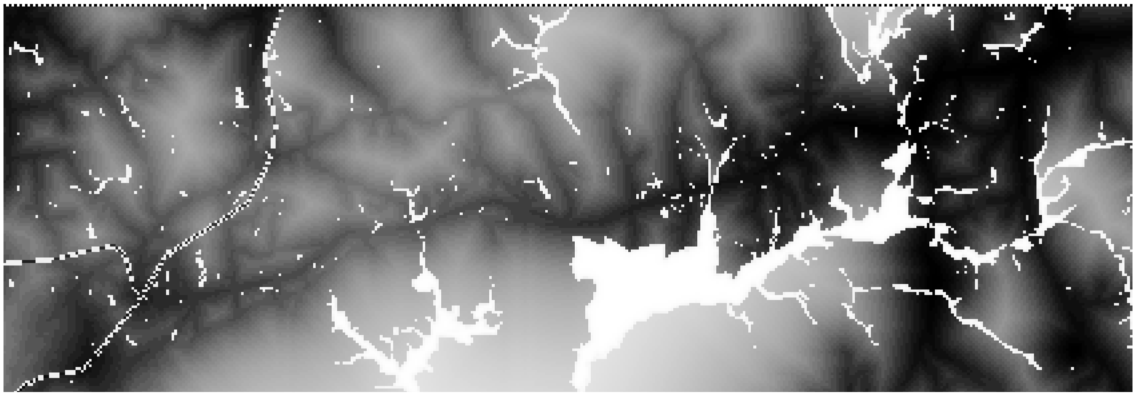


Figure 12. Sample intersection attractor map.

Ramp_attractor. The map shown in Figure 13 depicts the travel time to the nearest limited access highway ramp (e.g., U.S. Interstates). The intersection attractor map is used in LEAM because development is more likely near ramps.

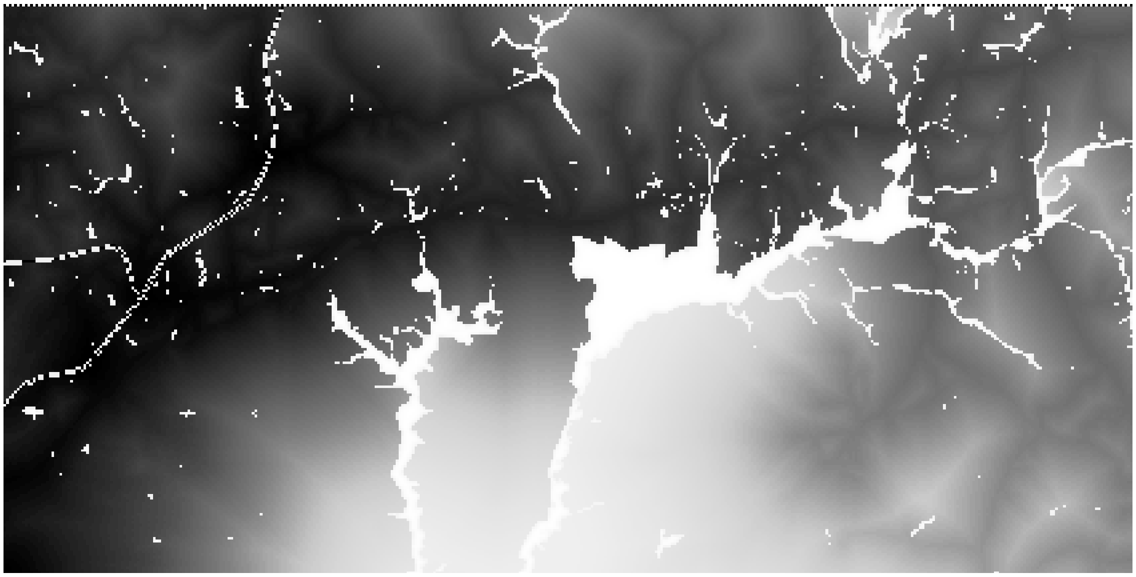


Figure 13. Sample limited access highway ramp attractor map.

Road attractor. The map shown in Figure 14 depicts the travel time to the nearest road. The road attractor map is used in LEAM because the existence of a road increases the likelihood of development occurring.

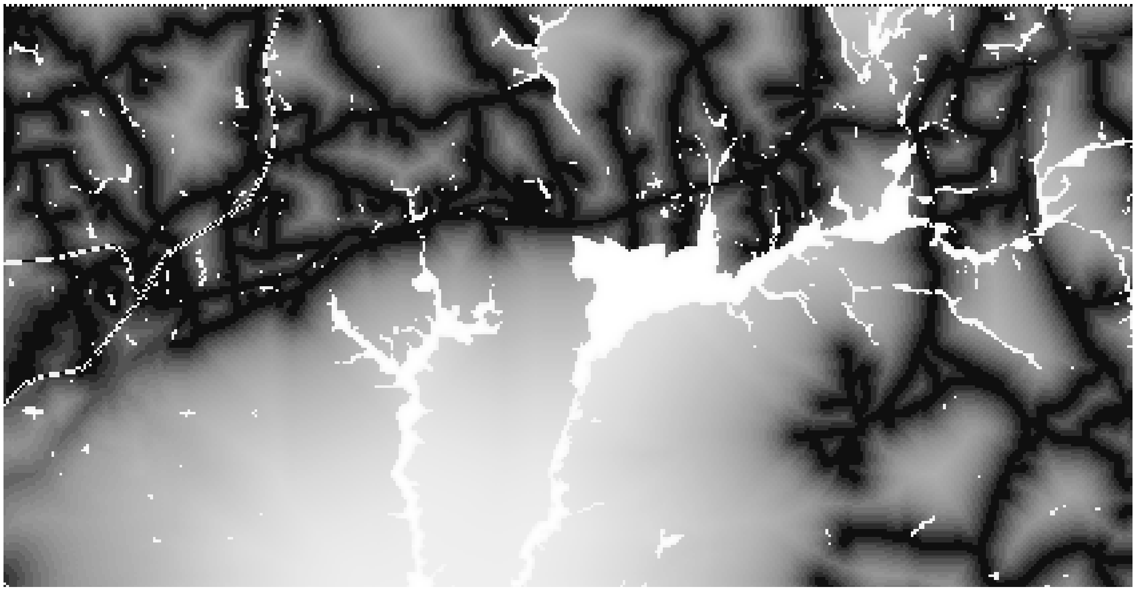


Figure 14. Sample county road attractor map.

Cities_attractor. The cities attractor map (Figure 15) denotes driving time to the nearest employment centers (municipalities). Because there are many municipalities in a study area, they are grouped into a few categories by population (e.g., large, medium and small cities). Attractor map is created for each group of municipalities by the same way as road attractor maps. Attractor map for each group is given different weights.

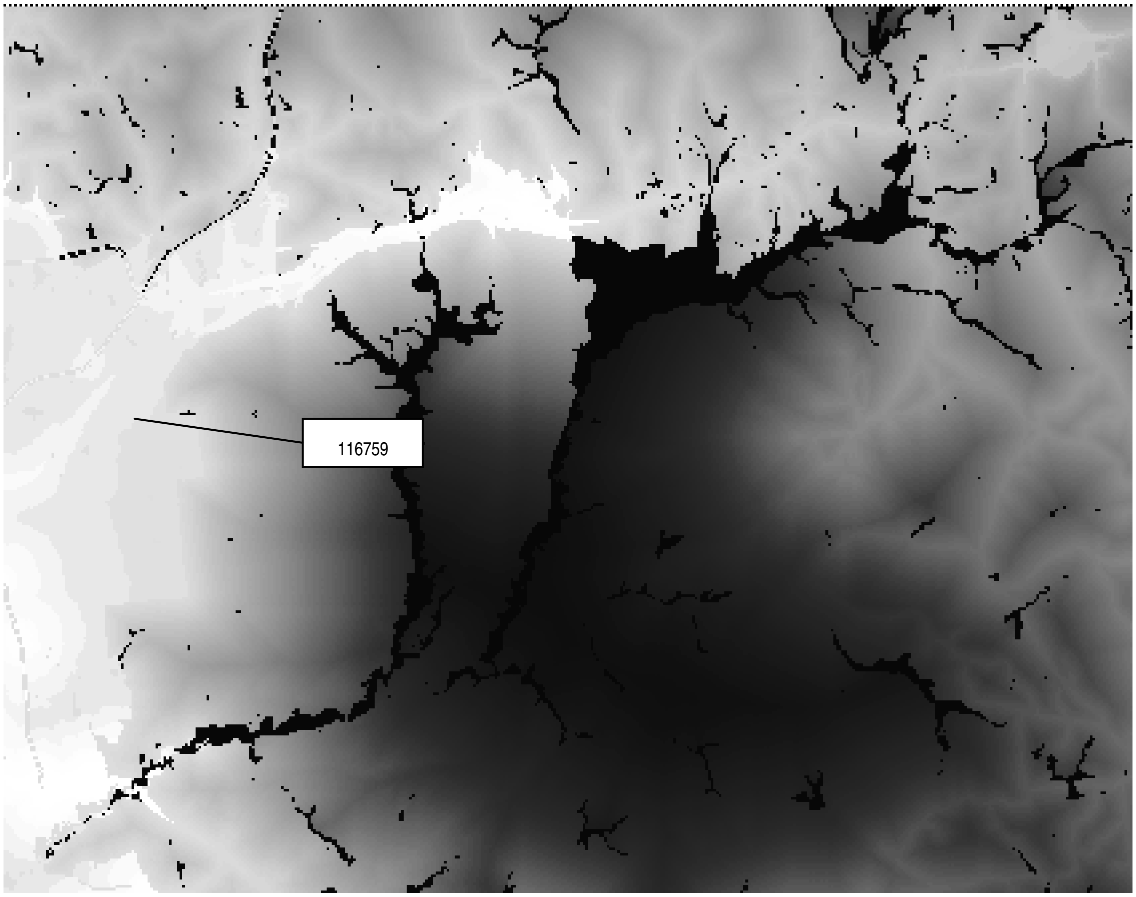


Figure 15. Sample cities attractor map.

General Approach

The mLEAM modeler must develop the input maps (shown above) for their area of interest. There are many approaches that can be used and users are encouraged to explore options and opportunities. However, the following figure suggests the maps that might be used to create the mLEAM input maps (identified along the right side). The grey-background maps along the left side are locally specific user maps that capture plans that mLEAM will test with respect to anticipated urban growth. Those without the grey background represent the generic nationally available maps. Two temporary maps (bottom center) are suggested as important steps in the process of creating attractiveness maps that consider driving time on the current and planned road/highway network. One is simply a description of the driving time required to cross each 30-meter-square cell and is used to generate several needed images. The other indicates the fundamental economic draw of cities and their associated job and commercial centers.

Most of the map processing is straightforward. Creation of the slope map requires running standard slope-aspect analysis models available in most raster GIS packages. The boundary map is simply an overlay identifying the counties involved in the simulation process. Similarly, the Metrobuffer map is a 1-0 map identifying the extent of anticipated edges of cities through the course of a simulation. These and the no-growth map are most challenging with respect to extracting agreement from the user community. Forest and water attractiveness are simply proximity analyses that give distance from each cell to the closest cell containing forest or water.

The road, highway, ramp, and intersection attractiveness maps are based on the travel time to drive to these features. A travel time map is first developed, which identifies the time (in minutes) required to traverse each cell. Artificially high values are assigned to such areas as water, permanent easements for limited access roads/highways, and railroads, to essentially indicate that passage is impossible. Using a raster GIS cumulative cost function, this map can be used to generate the various travel-time based attractiveness maps.

The most challenging, and perhaps the most important, input map to generate is the cities_attractor map. This map captures the notion of travel time to jobs and shopping. In the current release a value is assigned to each cell with respect to the most attractive center (rather than a cumulative summing of the attractiveness to all nearby centers). "Attractiveness" in this case is defined as the city center closest to a grid cell with respect to shortest travel time.

4a. Predict Land Development Probability, or 4b. Predict a Number of Alternative Futures

The user will run the mLEAM simulation(s) with the mLEAM input maps created in step 2, and one or more mLEAM output scenarios will result. The output map will show the likelihood of urban development for a given region. If more than one input scenarios are run, the user will be able to compare the outputs for multiple scenarios, thus allowing the prediction of a number of alternative development futures.

5. Identify Constraint Pressures on Military Training and Testing

Since the mLEAM outputs are purely graphical, planners can easily glean trends from the mLEAM models and see which areas of an installation might be affected by urban encroachment.

6. Identify the Value of Land for Training and Testing (or Carrying Capacity)

Military planners must decide the comparative values of installations' lands with respect to training and testing. In many cases, training/testing operations on certain areas within an installation have been suspended due to massive urban encroachment and the noise and dust complaints and lawsuits that follow. In these cases, it has simply become too costly for the military to use those grounds for training and testing. It is incumbent upon the military planners to recognize which lands are of top priority for saving and which are relatively expendable.

7. Compare Alternative Regional Policy/Investment Proposals (COAs)

The final goal of the mLEAM approach is for planners to take the insights learned from the growth models and prepare regional policies that will benefit the region the most. The graphical models can help show the predicted end result of a particular investment proposal or policy, and the opportunity to model multiple scenarios allows for easy comparison of alternative plans. If one of the models predicts growth in a pattern that is preferable to the others, planners might use that regional policy (or aspects of it) as a starting point for shaping urban growth in their area.

4 An mLEAM Example: Fort Bragg, NC

Challenge

Fort Bragg, a large Army base in south-central North Carolina, has been experiencing urban encroachment on its borders from neighboring Fayetteville, resulting in more noise and dust complaints from the new residents. This caused the installation to reconsider its use of some border lands for training and testing. The loss of this land for training and testing has both economic and national security implications. Due to the construction of a new four-lane limited-access highway running between Fort Bragg and Fayetteville, urban encroachment was foreseen as only getting more intrusive. The challenge was for a model to accurately and precisely predict where the new growth would occur, both with the new highway and without it.

Collecting Necessary Data and Preparing for Analysis

This section will describe the process by which the Fort Bragg area data was collected and converted into useable base files. This corresponds with the steps A, B, and C of the mLEAM computer approach, described in the first section of Chapter 3. Again, since GIS is an iterative process, there will not be a strict adherence to following a sequence from A to B to C. This section will show both methods for producing the mLEAM input maps: (1) using ESRI Arc/Info exclusively, and (2) using ESRI ArcGIS alongside GRASS.

Following the section describing the steps taken in creating the input maps, a set of schematic diagrams is included. These schematic diagrams depict the order of data acquisition and manipulation for each map layer. There are four diagrams:

1. Scenario without new highway, done with ESRI ArcGIS exclusively.
2. Scenario without new highway, done with ESRI and GRASS.
3. Scenario with new highway, done with ESRI ArcGIS exclusively.
4. Scenario with new highway, done with ESRI and GRASS.

The term “BR” is generally used as an abbreviation for “Bragg” (to shorten the filenames).

CD of Files Used to Create Fort Bragg mLEAM Scenario

A CD was burned containing the files used in the creation of the Fort Bragg mLEAM scenario. In most cases, the CD filenames correspond with the suggested filenames given in the tutorial shown on the following pages. A complete list of the files on the CD is included in the Appendix A.

ESRI ArcGIS Only Version

Initial Setup

1. Create directory on hard drive (e.g., C:\Fort_Bragg) to hold data. There should not be any spaces in the path names.
2. Obtain Arc/Info scripts and support files from LEAM lab, and save them to the directory created in step 1. These files are:

   1bnd_cover
   1bnd_grid
   2landuse
   3munbuf
   4travel
   6roadatr
   7flood
   8cityatr
   8growth
   9nogrow
   10rampatr
   11rdintatr
   12slope
   13stateatr
   14treeatr
   15wateratr
   16facility
   astrgrdp
   astrgrdp2
   boundremap
   boundremap2
   cityremap
   facilityremap
   fldremap
   geog_alb
   growremap
   intersection_point
   munbndremap
   nogrowremap
   roadremap
   travelremap1
   travelremap2
   travelremap3
   travelremap4
   travelremap5

3. Create workspace in Arc/Info. The command will be “w” followed by the path to the directory created in step 1 (e.g., “Arc: w c:\fort_Bragg”). This workspace will be the place where files are accessed and saved. This workspace must be opened each time Arc/Info is used.
4. Notes: Some of the following maps can be created by running AML scripts.[*] These scripts are collections of commands that run automatically in Arc/Info. If the scripts are used, some of the intermediate steps described will not be necessary, because the scripts already contain those commands. The references to the scripts are shown in italics.

It is imperative that the AML scripts be saved to the same directory as the maps being worked on; otherwise, Arc/Info will not recognize them. When running an

[*] Appendix B includes a listing of AML scripts.

AML script from an "Arc:" prompt, type "&run <pathname>.<AMLscript> <arguments (if necessary)>" without the quotes.

Creating Maps

1. County Boundary Map. This will describe taking a raw vector data file of county boundaries from a data source and manipulating it into a LEAM input map file. The result will be two files: (1) a vector coverage of the county boundaries in the Bragg area, and (2) a grid of the same area.

Source(s)

http://www.census.gov/geo/www/cob/co2000.html

Procedure

1. Go to website above, select North Carolina, download and unzip the file → co37_d00.e00.
2. In Arc/Info, Import the cover by Arc IMPORT command (e.g., "Arc: import cover co37_d00.e00 nc_counties").
3. Project NC_COUNTIES using geog_alb.txt projection file (e.g., "Arc: project cover nc_counties nc_counties_al geog_alb.txt").
4. Build polygons by CLEAN command (e.g., "Arc: clean nc_counties_al").
5. To limit the area to just the Fort Bragg area, select necessary counties using the RESELECT command (Counties in Fort Bragg region: Cumberland, Harnett, Hoke, Lee, Moore, Richmond, and Scotland). ("Arc: reselect nc_counties_al br_county poly" → " >: RES county lk '051' or county lk '085' or county lk '093' or county lk '105' or county lk '125' or county lk '153' or county lk '165'" | type 'n's to the questions that follow.

   AML Script Opportunity: Procedures (2) thru (5) can be done by running the boundary cover AML script (e.g., "Arc: &run C:\Fort_Bragg\1bnd_cover.aml"):

   - *For the exchange file name, type "CO37_D00".*
   - *For the resulting cover name, type "BR_COUNTY".*
   - *For the logical expressions, type the commands/arguments used in step (5) at the >: prompt.*

6. Convert BR_COUNTY to a grid (e.g., BR_COUNTY_G) and reclassify so that all cells have the value of 1 (e.g., BR_GRID_1).
7. Clip BR_GRID_1 with the cover BR_COUNTY as BR_GRID using GRIDCLIP command in GRID module (e.g., "Grid: gridclip br_grid_1 br_grid cover br_county").
8. Export BR_COUNTY as an e00 file so that it can be used for other applications (e.g., "Arc: export cover br_county br_county.e00").

9. Export BR_GRID_G_1 as ASCII (e.g., "Arc: gridascii br_grid_g_1 br_county.asc").

   AML Script Opportunity: Procedures (6) through (9) can be done by running the boundary grid AML script: "Arc: &run C:\Fort_Bragg\1bnd_grid.aml":

   - *For boundary cover name, type "BR_COUNTY".*
   - *For resulting grid name, type "BR_GRID".*

2. Landuse Map. This will describe taking a raw NLCD raster data file from a website and manipulating it into a LEAM input map file. The result will be a raster map showing land use in the Bragg area by grid cell (Figure 16).

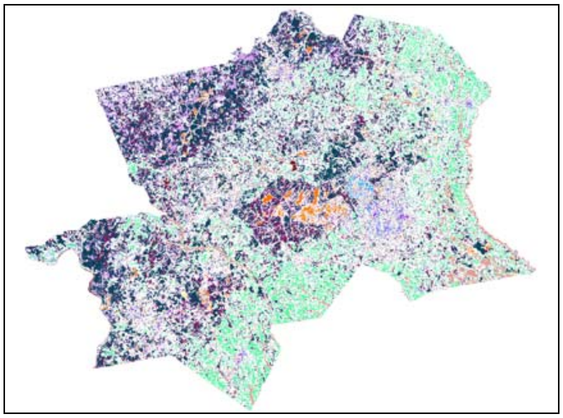


Figure 16. Land use, Fort Bragg region.

Source(s)

USGS National Land Cover Data (NLCD) from:

http://edcsgs9.cr.usgs.gov/pub/data/landcover/states/

Road network data from Census:

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

Municipal boundary data from Census:

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

Procedure

1. Go to USGS NLCD website, Choose North Carolina, Download a NLCD TIFF file of North Carolina from NLCD site, unzip and save as grid → NC_NLCD.
2. Download road shapefiles and municipal boundary shapefiles of the same counties (Cumberland, Harnett, Hoke, Lee, Moore, Richmond, Scotland) and merge into one file for each (e.g., BR_ROAD.SHP and BR_MUNBND.SHP).
3. To do this: From ESRI website: choose North Carolina → Select by layer "Line Features—Roads" → Pick counties → Download files and unzip → Open layers in ArcMap and merge into one road shapefile
   Repeat for municipal boundaries, except choose "Designated Places 2000" from the Select by Layer list.
4. In Arc/Info, clip the NC_NLCD grid with the boundary cover BR_COUNTY → save as BR_NLCD (e.g., "Arc: grid" → "Grid: gridclip nc_nlcd br_nlcd cover br_county").

5. Convert BR_ROAD.SHP to a cover using SHAPEARC and ROUTEARC commands in ARC (e.g., "Arc: shapearc br_road.shp br_road1 subclass" → "Arc: routearc br_road1 subclass br_road2") then kill BR_ROAD1 ("Arc: kill br_road1 all").
6. Project the cover BR_ROAD2 to Albers as BR_ROAD1. (This step may be unnecessary if the road file is already projected.)
7. Convert BR_MUNBND.SHP to a polygon cover using SHAPEARC and CLEAN commands in ARC and project to Albers as BR_MUNBND1 ("Arc: shapearc br_munbnd.shp br_munbnd" → "Arc: clean br_munbnd" → "Arc: projectdefine → Project.
8. Select CFCC2 A1, A2, A3 and A6 class roads from BR_ROAD1 and save as BR_ROAD2 (e.g., "Arc: RESELECT br_road1 br_road2 LINE" then type logical expressions like "res cfcc2 = 'A1' OR cfcc2 = 'A2' OR cfcc2 = 'A3' OR cfcc2 = 'A6'") then kill BR_ROAD1. The 'A1' and 'A2', etc. part is case sensitive.
   This selection is done to select the major roads (Interstates, Highways, Ramps).
9. Convert BR_ROAD2 to a grid BR_ROAD1 and reclassify as 24 (e.g., "Grid: br_road24 = (br_road1 * 0) + 24") then kill BR_ROAD1.
10. Create 70m buffer along BR_ROAD2 as BR_R_BUFF and kill BR_ROAD2.
11. Clip the buffer cover BR_R_BUFF using the municipal boundary cover BR_MUNBND1 as BR_R_BUFF1 (e.g., "Arc: ERASE br_r_buff br_munbnd1 br_r_buff1") and clean BR_R_BUFF1.
12. Convert BR_R_BUFF1 to a grid BR_R_BUFF2 with INSIDE field as cell values and kill BR_R_BUFF1.
13. Select cell value 100 from BR_R_BUFF2 and save as another grid BR_R_BUFF1.
14. Select cell values 21 and 22 from BR_NLCD
    (21 = Low Intensity Residential, 22 = High Intensity Residential).

AML Script Opportunity: Procedures (4) through (13) can be done using the landuse AML script: "Arc: &r C:\Fort_Bragg\2landuse_new.aml br_road br_nlcd br_munbnd br_nlcd_r"

> *Caution: There are four arguments used in this AML script: The first argument is the name of road shape file, the second one is the name of land cover grid, the third one is the name of municipal boundary shapefile and the last one is the output grid name → no extensions (.SHP).*
>
> *Shapefiles should be in Geographic Coordinates.*
>
> *Remap files ASTRGRDRP.TXT and ASTRGRDRP2.TXT must be located in the workspace.*

3. Buffer Map. This will describe taking a raw vector data file of municipal boundaries from a data source, combining it with a county boundary file, and manipulating into a LEAM input map file. The result will be a raster map showing a 1.5 mile buffer around all municipal boundaries in the Bragg area (Figure 17).

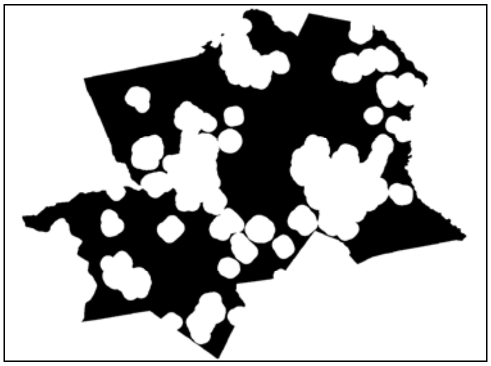


Figure 17. Fort Bragg municipal boundary buffer (1.5 Mi).

Sources

Municipal boundary data from http://arcdata.esri.com/data/tiger2000/tiger_download.cfm
County boundary data from #1 (e.g., br_county.e00)

Procedure

1. Go to the web page above and select "North Carolina."
2. Download designated places 2000 for counties of interest (Cumberland, Harnett, Hoke, Lee, Moore, Richmond, Scotland).
3. Unzip the file, open in ArcMap and merge into one shape file (e.g., BR_MUNBND.SHP).
4. Import the boundary cover from the boundary data (e.g., "Arc: import cover br_county.e00 br_bnd") and convert to a grid (e.g., "Arc: polygrid br_bnd br_bnd_g" and type '30' for cell size and 'y' for converting the entire cover).
5. Convert the shapefile to a coverage (e.g., "Arc: shapearc br_munbnd.shp br_mun subclass" → "Arc: clean br_mun br_mun_poly").

   Note: The name SUBCLASS is the name of info file where the attributes are saved. It does not really matter how you name it.
6. Project the BR_MUN_POLY from GCS to Albers, name as BR_MUN, and build (or clean) BR_MUN.

   Note: Depending on the input file coordinate system, it may be necessary to use PROJECTDEFINE before projecting.
7. Create 1.5 mile buffers outside the boundary of BR_MUN.
8. Create a new field in the table and put 1 to all the records (as integer) ArcView→Attribute Table→Editing→Add Field.

9. Convert to a grid with the new field as values BR_MUNBUFF.
10. Merge BR_MUNBUFF onto BR_BND_G and save as BR_MB.
11. Kill BR_MUNBUFF.
12. Clip the grid with the boundary cover imported from br_county.e00 (e.g., "Grid: latticeclip br_mb br_bnd br_munbuff") → BR_MUNBUFF grid is the final result.

AML Script Opportunity: Procedures (4) thru (12) can be done using the municipal buffer AML script: "Arc: &r C:\Fort_Bragg\3munbuf.aml.

Files 'boundremap2' file and 'munbndremap' must be in the workspace.

For the municipal boundary shapefile without extension, type "BR_MUNBND".

For the boundary GRID, type "BR_GRID".

For the BOUNDARY cover, type "BR_COUNTY".

4. Travel Minute Map. This will describe combining the road network file with the land use file from previous steps. The result will be a raster file showing travel time across each grid cell in minutes, for the Bragg area (Figure 18). There is a maximum travel time of approximately 37 minutes across a grid cell due to a programmer-defined mathematical limitation. This map will be an intermediate map—it will be used in conjunction with other landscape maps to make attractor (travel time) maps, but will not be used as a land use change input map per se.

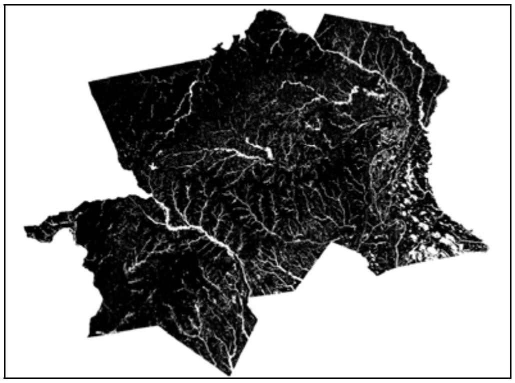


Figure 18. Travel time across grid cells, Fort Bragg region.

Source

BR_NLCD map: Road network data from:

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

Procedure

1. Download road data from the website above for North Carolina counties of interest (Cumberland, Harnett, Hoke, Lee, Moore, Richmond, Scotland).
2. Open these files in ArcMap and merge the shapefiles into one file (e.g., BR_ROAD.SHP). It is acceptable to use the same file from the LANDUSE map #2.
3. Select roads whose CFCC2 is A1, A2, A3, A4, A5, A6 or A7.
4. Add a new numeric field named ‘class’ and assign 1 for A1 records, 2 for A2 records, 3 for A3 records, etc.

   Editor→Add field called “Newfield”→Calculate as [Cfcc2].Right(1).

   Then, Add another field called “class,” making sure that it is categorized as a number→Calculate as [Newfield].AsNumber.

   Then delete the field “newfield”.

   Export this to a .e00 file.
5. Assigning travel speed to road data, Add a new field and assign values (travel speed in 0.001mph) as follows:

A1: 70000

A2: 50000

A3: 40000

A4: 30000

A5: 20000

A6: 30000

A7: 10000.

6. Convert to a grid (e.g., BR_ROAD) with the values above as cell values.
7. Travel speed grid.

   Assign travel speed (0.001mph) values to different land cover cells (name it temporarily XXX).

   NLCD codes 21, 22 or 23: 10000 (codes represent urban areas: 21=Low Intensity Residential, 22=High Intensity Residential, 23=Comm/Ind Transportation).

   Non-urban land cover: 500

   Limited-access hwy buffer cells: 1

   Open water and wetlands cells: 1
8. Merge BR_ROAD onto the XXX grid and save as BR_TRAVELSPEED.
9. Travel minute grid.

   A travel minute grid is made using the equation under ArcGrid (e.g., “Grid: TRAVELMIN = 1000 / (26.8224 * BR_SPEEDLIMIT)”).

   Example: when speed limit is 60mph, travel minute = 1000 (26.8224 * 60000) = 0.000621min/m = 0.621min/km = 1min/mile.

   26.8224 is a conversion factor.

AML Script Opportunity: Procedures (5) thru (9) can be done using the travel map AML script, “Arc: &r C:\Fort_Bragg\4travel.aml”.

Note: the ROAD COVER file should be a .e00 file. It should have the “Class” field from step (1). It should be in Geog Coord System (like the original shapefile was in), not in a Projected System like Albers. If the file is in a projected system, it will result in an error.

For the value item field name, type “CLASS”.

For limited-access hwy class, type “1”.

For ramp class, type “6”.

Files ‘travelremap1’, ‘travelremap2’, ‘travelremap3’, ‘travelremap4’, and ‘travelremap5’ should be in the same workspace

10. Convert BR_ROAD.SHP as a cover and export as an e00 file for future use, e.g.,:

    “Arc: shapearc br_road.shp br_road subclass”

    “Arc: routearc br_road subclass br_road1”

    “Arc: export cover br_road1 br_road.e00”

5. City Attractor Maps (a. Large, b. Medium, c. Small). This will describe taking a raw vector data file of municipal boundaries in the Bragg area from a data source, adding population characteristics from the U.S. Census Bureau, selecting certain polygons based on population, doing a cost calculation based on the travel time map made previously, and manipulating it into three Land Use Change input map files. The result will be three files: raster files showing travel time from each grid cell to: (1) large cities, (2) medium cities, and (3) small cities (Figure 19).

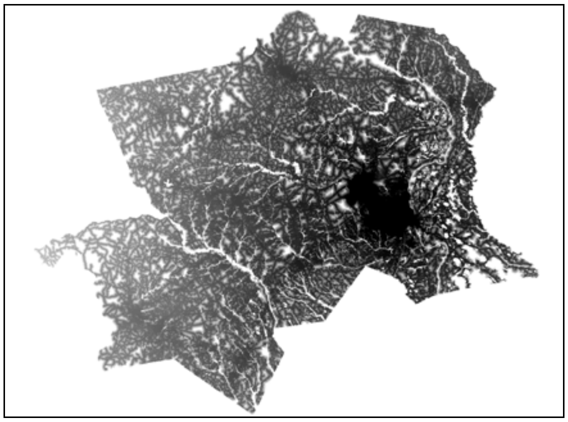


Figure 19. Travel time to large city, Fort Bragg region.

Source

Municipal boundary (Census population data of incorporated places) from:

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

Procedure

1. Download municipal boundary data or use file from #3.
2. Open the table of the shapefile, add a column "population" and input the values from Census.
3. Divide the cities into necessary groups (e.g., large, medium and small) based on population distribution. This is a user-defined arbitrary classification.
4. Generate shapefiles of large cities, medium cities and small cities (e.g., BR_LCITY.SHP, BR_MCITY.SHP and BR_SCITY.SHP).
5. Convert each shapefile to grids (e.g., BR_LCITY, etc.).
6. Run COSTDISTANCE command with BR_LCITY as source and TRAVELMIN as cost with the maximum value 120 (e.g., "Grid: br_lcity_att = int (costdistance(br_lcity,travelmin))").
7. Merge BR_LCITY_ATT onto a boundary grid whose cell values are all 120.
8. Repeat for medium and small cities → city attractor maps are produced for each group of cities.

AML Script Opportunity: Procedures (5) thru (7) can be done using the city attractor script: "Arc: &r C:\Fort_Bragg\8cityatr.aml"

> *For the city shape file name, type the name of the city file (e.g., BR_LCITY.SHP)*
>
> *Repeat for other size cities (there should be a Large, Medium, and Small size city attractiveness file when complete)*

6. Road Attractor Map. This will describe taking a raw vector data file of roads in the Bragg area from a data source, doing a cost calculation based on the travel time map made previously, and manipulating it into a LEAM input map file. The result will be a raster file showing travel time from each grid cell to the nearest city road (Figure 20).

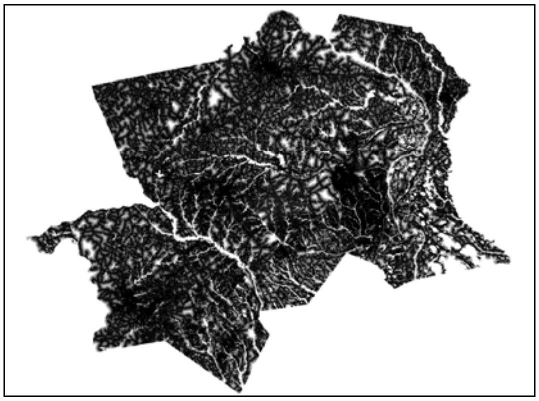


Figure 20. Travel time to nearest road, Fort Bragg region.

Source

Road network data from:

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

Travel minute grid from 4 (e.g., TRAVELMIN)

Procedure

1. Import the exchange file (e00) from 4 to a cover (e.g., BR_ROAD).
2. Select CFCC2 = “A4” AND Fetype = “RD” and convert to a grid (e.g., BR_A4RD).

   Note: “A4” is selected to highlight the city roads (class A4) only
3. Project the grid.
4. Run COSTDISTANCE command with BR_A4RD as source and TRAVELMIN as cost and set the values to integer.

AML Script Opportunity: Procedures (1) thru (4) can be done using the road attractor AML script: “Arc: &r C:\Fort_Bragg\6roadatr.aml”

For the value item field name, type “CLASS”.

For roads and streets, type “4”.

7. No Growth Zone Map. This will describe taking raw vector data files of no growth areas in the Bragg area from a data source, and manipulating it into a LEAM input map file. These areas consist of Federal lands, Indian lands, water bodies, roads, landmark polygons, etc. The result will be a raster file showing all places within the Bragg area on which development cannot occur (Figure 21).

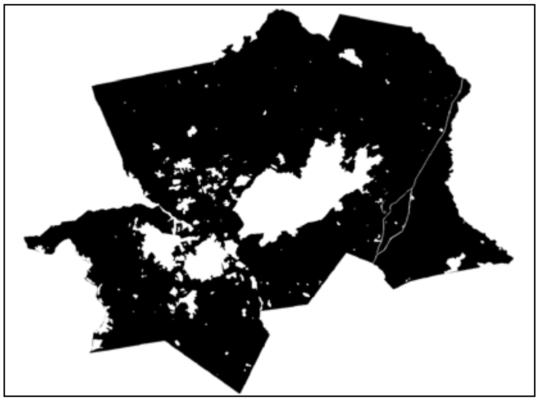


Figure 21. No growth areas, Fort Bragg region.

<u>**Source**</u>

1. National Atlas for Federal and Indian land: http://www.nationalatlas.gov
2. Other local data about ownership (if possible)
3. Landmark polygon data from: http://arcdata.esri.com/data/tiger2000/tiger_download.cfm
4. Road network data exchange file (e00)
5. Boundary cover (e.g., BR_COUNTY) and boundary grid (e.g., BR_GRID)

<u>**Procedure**</u>

1. Merge all shapefiles except roads and boundary into one shapefile (e.g., BR_NOGROW).
2. Convert the shapefile to a grid and save as: BR_NOGROW.
3. Select limited-access highways (A1 class) from the roads cover and make 50m buffer because developments right beside limited-access highways are regarded as unlikely → convert to a grid: BR_I_50MBUFF.
4. Merge BR_NOGROW and BR_I_50MBUFF.
5. Clip the merged grid with the boundary grid.
6. Reclassify the grid so that within the area: 1 and others: 0.
   AML Script Opportunity: Procedures (2) thru (6) can be done by running the No Growth AML script: "Arc: &r C:\Fort_Bragg\9nogrow.aml"
   - *For the input shape file name, type "BR_NOGROW.SHP".*
   - *For the road exchange file name, type "BR_ROAD".*

- *For the value item field, type "CLASS".*
- *For the limited-access highways, type "1".*
- *For the input boundary cover, type "BR_COUNTY".*
- *For the boundary grid, type "BR_GRID".*

"Nogrowremap" and "facilityremap" files must be in the workspace

8. Ramp Attractor Map. This will describe taking a raw vector data file of roads in the Bragg area from a data source, doing a cost calculation based on the travel time map made previously, and manipulating it into a LEAM input map file. The result will be a raster file showing travel time from each grid cell to the nearest highway ramp (Figure 22).

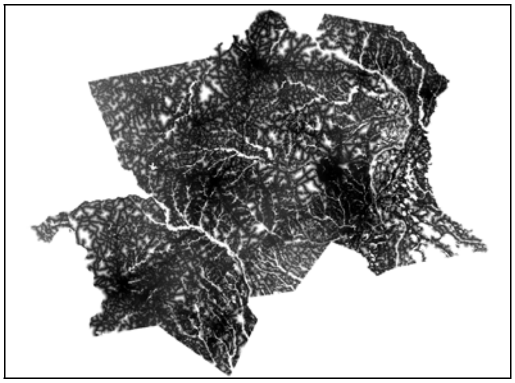


Figure 22. Travel time to nearest highway ramp, Fort Bragg region.

Source

Road network data from:

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

Travel minute grid from 4 (e.g., TRAVELMIN)

Procedure

1. Select ramps from the roads cover and converted to a grid: BR_RAMP.
2. Run COSTDISTANCE command with BR_RAMP as source and TRAVELMIN as cost and set the values to integer.

AML Script Opportunity: This entire procedure can be done using the ramp attractor AML script:"Arc: &r C:\Fort_Bragg\10rampatr.aml".

9. Road Intersection Map. This will describe taking a raw vector data file of roads in the Bragg area from a data source, extracting intersection nodes from the file, and manipulating it into a LEAM input map file. The result will be a raster file showing travel time to major intersections (Figure 23).

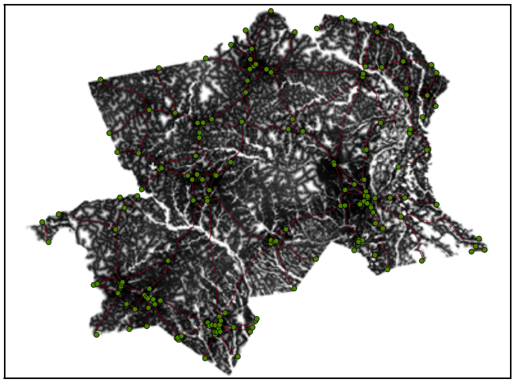


Figure 23. Travel time to major intersections, Fort Bragg region.

Source

Road network data from #2

Procedure

1. A2 and A3 classes are selected from the road data and saved as a shapefile.
2. Generate a point theme where the points are intersections of the selected roads: an avenue script was made by Yongwook Kim.
3. Use the project file intersection_point.apr.
4. Add the necessary road shapefile.
5. Press “C” icon to consolidate arc lines → a new shapefile is made.
6. Press “N” icon to generate a node shapefile from the new shapefile.
7. Pay attention to the pop-up windows.
8. Convert the point theme to a grid: BR_RO_NODE.
9. Run COSTDISTANCE command with BR_RO_NODE as source and TRAVELMIN as cost and set the values to integer.

AML Script Opportunity: Procedures (3) thru (4) can be done using the intersection AML script: “Arc: &r C:\Fort_Bragg\11rdintatr.aml”

Must have the point shapefile in the workspace

10. Slope Map. This will describe taking a digital elevation model (DEM) of the Bragg area from a data source, deriving the slope of the land, and manipulating it into a LEAM input map file. The result will be a raster file showing physical slope for each grid cell in the Bragg area (Figure 24).

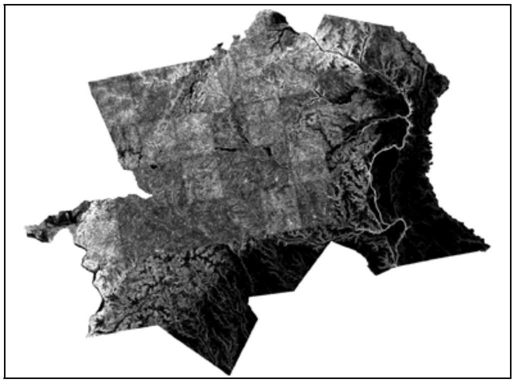


Figure 24. Physical slope, Fort Bragg region.

Source

USGS 1:24K DEM (the second one is recommended):

http://edc.usgs.gov/geodata/
http://seamless.usgs.gov/

Procedure

1. Go to http://seamless.usgs.gov/, use download tool on National Map, extract DEM as a zip file → Download file to appropriate folder (there may be more than one file to download) → If there is more than one DEM file, you will need to merge them into one file; this can be done in ArcInfo

   (e.g., "Arc: grid → Grid: br_demmerge = merge(c:\fort_Bragg\XXXXXXXX, c:\fort_Bragg\YYYYYYYY)"), where XXXXXXXX and YYYYYYYY are 8-digit numbers representing the downloaded DEM section files.

   Note: Due to the download process, the 8-digit DEM files may be held in a subfolder with the same name as the 8-digit filename. In this case, the Arc command would be "Arc: grid → Grid: br_demmerge = merge(c:\fort_Bragg\XXXXXXXX\XXXXXXXX, c:\fort_Bragg\YYYYYYYY\YYYYYYYY)"

   Note: to get out of Grid mode, enter "q" or "quit"
2. Project merged grid to Albers projection (this will take a long time in ArcInfo).
3. Clip the grid with the boundary grid.

4. Derive slope using ArcView derive slope menu.

   AML Script Opportunity: Procedures (2) thru (4) can be done using the slope AML script:

   "Arc: &r C:\Fort_Bragg\12slope.aml"

11. State Road Attractor Map. This will describe taking a raw vector data file of roads in the Bragg area from a data source, doing a cost calculation based on the travel time map made previously, and manipulating it into a LEAM input map file. The result will be a raster file showing travel time from each grid cell to the nearest state highway.

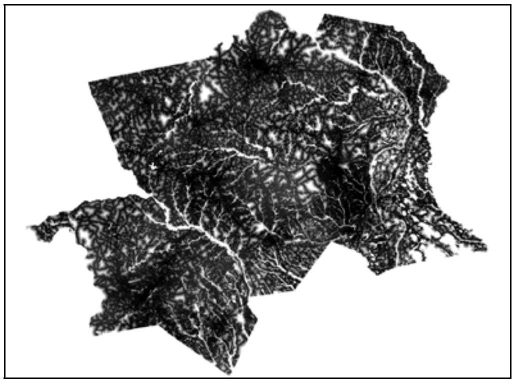


Figure 25. Travel time to state highways, Fort Bragg region.

Source

Road network data as an exchange file from #4
Travel minute grid from 4 (e.g., TRAVELMIN)

Procedure

1. Select state highways from the roads cover and convert to a grid: BR_STHWYS.
2. Run COSTDISTANCE command with BR_STHWYS as source and TRAVELMIN as cost and set the values to integer.

AML Script Opportunity: This entire procedure can be done using the state highway attractor AML script:"Arc: &r C:\Fort_Bragg\13stateatr.aml"

Tree Attractor Map. This will describe taking information from the Bragg area NLCD map, calculating distance to the nearest forest, and manipulating it into a LEAM input map file. The result will be a raster file showing Euclidean distance from each grid cell to the nearest forest grid cell (Figure 26).

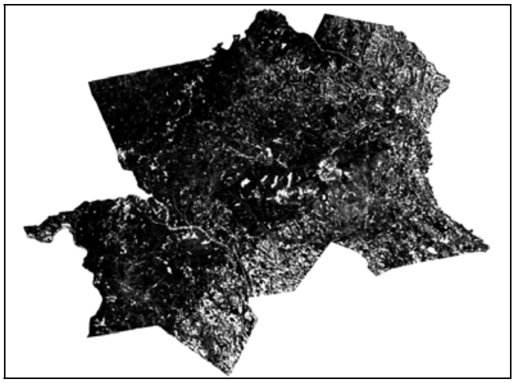


Figure 26. Distance to nearest forest, Fort Bragg region.

Source

USGS NLCD

Procedure

1. Select forest cover cells (41, 42 and 43) from the NLCD grid: BR_FOREST.
2. Use FIND DISTANCE menu to find distance to forest cells and set the values to integer.

AML Script Opportunity: This entire procedure can be done using the tree attractor AML script:

> *"Arc: &r C:\Fort_Bragg\14treeatr.aml".*

12. Water Attractor Map. This will describe taking information from the Bragg area NLCD map, calculating distance to the nearest water body, and manipulating it into a LEAM input map file. The result will be a raster file showing Euclidean distance from each grid cell to the nearest water or wetland grid cell (Figure 27).

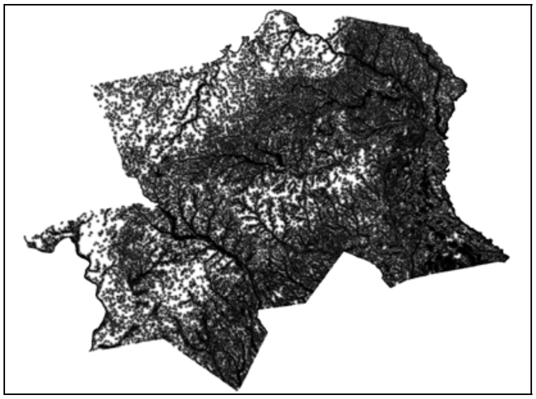


Figure 27. Distance to nearest water, Fort Bragg region.

Source

USGS NLCD

Procedure

1. Select open water and wetland cells (11, 91, and 92) from the NLCD grid: BR_WATER.
2. Use FIND DISTANCE menu to find distance to water cells and set the values to integer.

AML Script Opportunity: This entire procedure can be done using the water attractor AML script: "Arc: &r C:\Fort_Bragg\15wateratr.aml"

****

All maps created in this tutorial are mLEAM input maps and are now ready for use with the LEAM software.

****

ESRI ArcGIS with GRASS Version

Initial Setup

1. On Windows machine, create directory on hard drive (e.g., C:\Fort_Bragg) to hold data. There should not be any spaces in the path names.
2. On Windows machine, install the program WinSCP—this will allow the user to move files back and forth between the Windows environment and the UNIX environment. WinSCP is available for download on the internet.
3. On Windows machine, ensure that ESRI ArcGIS is installed.
4. On UNIX computer (or an X-Windows interface to a remote UNIX computer) , ensure that GRASS is installed.
5. Before the manipulated data is brought into GRASS, it is necessary to create two different LOCATIONS in GRASS. One LOCATION should have a Latitude-Longitude Geographic coordinate system and should be named as such. The other LOCATION should have an Albers Equal Area projection system and should be named as such. These two LOCATIONS will make it possible to import most of the manipulated data files, as these files typically are projected in either Lat/Long or Albers. The r.proj command can take a file from one LOCATION and reproject it into the current LOCATION's projection.

Creating Base Maps

Map 1: County boundary map (BR_COUNT). The GRASS version is a raster file showing the counties of interest.

Creating Map 1 from scratch using ArcToolbox and ArcMap 8.3:

Source: County boundary data from URL:
http://www.census.gov/geo/www/cob/co2000.html

1. Select North Carolina, download and unzip the .e00 file to the directory of choice.
2. Import .e00 file using ArcToolbox: Conversion Tools→Import to coverage→Import from interchange file.

   Input file=<co37_d00.e00>

   Output dataset=<br_county> (Ensure the output file path is the same as the input file path.)
3. Project file to Albers using Projections→Project wizard.
4. Convert to shapefile using ArcToolbox: Conversion Tools→Export from Coverage→Coverage to Shapefile.
5. Select necessary counties (Cumberland, Harnett, Hoke, Lee, Moore, Richmond, Scotland) using ArcMap, then save the selection as a new file.

6. Copy this file to UNIX environment using WinSCP.
7. The resulting map should be a shapefile of the counties of interest, and this can be imported into GRASS using r.in.shape. The GRASS location should be the one with the Albers Equal Area projection defined.

   (Example: r.in.shape in=counties.shp out=counties)

Map 2: NLCD land use data (BR_NLCD). Creating Map 2 from scratch:

Source: USGS National Land Cover Data (NLCD) from URL:
http://seamless.usgs.gov

1. Download a NLCD TIFF file of North Carolina and unzip.
2. Save as grid (nc_nlcd.tif) and copy to UNIX environment using WinSCP.
3. This file can be imported into GRASS using r.in.gdal.

   (Example: r.in.gdal –o input=nc_nlcd.tif output=nc_nlcd)

Map 3: Roads (BR_ROADS). GRASS version is raster file showing roads on counties of interest. The roads are classified according to speed limits (A1 through A7).

Creating Map 3 from scratch:

Source: Road network data from Census:
http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

1. Download road shapefiles of counties of interest (Cumberland, Harnett, Hoke, Lee, Moore, Richmond, Scotland) and merge into one file in ArcMap 8.3 (e.g., ROAD.SHP).
2. Open ArcView 3.2 and make a new field for the roads:

   Open ROADS theme and open the attribute table.

   Create two new fields, one titled “Class” and one titled “New.” “Class” should be a numeric field, and “New” should be a text field.

   Start editing.

   Select all entries in the attribute table.

   Calculate values for “New” with the expression =[Cfcc2].Right(1).

   The switch for String will have to be checked.

   Calculate values for “Class” with the expression =[New].AsNumber.

   The result should be that “Class” shows the numerical values for “Cfcc2,” without the “A” character in front of the number.

   Save edits.
3. Copy this file to UNIX environment using WinSCP.

4. This file can be imported into GRASS using r.in.shape. It is important to include the additional parameter "cat=class" so that the roads are classified according to their speed limit classes. (Example: r.in.shape in=road.shp out=road cat=class).

Map 4: Municipal boundaries map (BR_MUNBND). GRASS version is raster file showing municipal boundaries of all towns within the counties of interest.

Creating Map 4 from scratch:

Source:

Municipal boundary data from Census

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

Census population data of incorporated places: Summary File 1→Geographic Comparison Tables→State→Place (GCT-PH1)

1. Download municipal boundary shapefiles of counties of interest (Cumberland, Harnett, Hoke, Lee, Moore, Richmond, Scotland) and merge into one file in ArcMap 8.3 (e.g., MUNBND.SHP).
2. Open the table of the shapefile, add a field "population" with long integer, start editing, and input the values from Census. Save edits.
3. Copy this file to UNIX environment using WinSCP.
4. This file can be imported into GRASS using r.in.shape. To create maps 5, 6, 7, it is also important to include the additional parameter "cat=Population." This will probably need to be imported into the Albers LOCATION, since the merged municipal boundary layers usually are projected with Albers by default.

   (Example: r.in.shape in=munbnd.shp out=munbnd cat=Population)

Maps 5, 6, 7. (Small city boundaries map (BR_SMCITY); Medium city boundaries map (BR_MEDCITY); Large city boundaries map (BR_LGCITY). GRASS version is three distinct raster files showing municipal boundaries of small, medium, and large cities within the counties of interest. These files were created from the BR_MUNBND map (map 4) above. The cities were arbitrarily classified into three groups with population data from the U.S. Census.

Creating Maps 5, 6, 7 from scratch:

If the BR_MUNBND map (Map 4) was imported with the category "Population," maps 5, 6, and 7 can be created in GRASS.

1. In GRASS, use r.reclass to categorize the city polygons into three classes based on their population. The result will be a new GRASS map.

   (Example: r.reclass munbnd out=class_munbnd

1 thru 1000 = 1 small city

1001 thru 24000 = 2 med city

24001 thru 150000 = 3 large city)

2. In GRASS, use r.reclass again (this time, on the new map created in step 2) to make a map of small city polygons only.

   (Example: r.reclass class_munbnd out=small_city

   1 = 1 small city

   2 thru 3 = NULL)

3. Repeat step 3 to make maps for medium city polygons and large city polygons.
4. The result will be three different maps, each showing polygons of small, medium, or large cities (maps 5, 6, 7).

Map 8. No growth map (BR_NOGROWTH). GRASS version is raster file showing areas where growth will not occur. These areas include U.S. military installations, Government lands, landmarks, water bodies, Indian lands, and the like.

Creating Map 8 from scratch:

Source (use some or all of these sources)

National Atlas for Federal and Indian land:

http://www.nationalatlas.gov

Other local data about ownership

Landmark polygon data from URL:
http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

ROADS map (Map #3)

Boundary cover shapefile

1. In ArcMap 8.3, gather the appropriate maps for the counties of interest from sources i through iii above and merge into one “no growth” shapefile. You will need to clip the Federal/Indian map with the county boundary shapefile before merging with the other maps.
2. Copy this file to UNIX environment using WinSCP.
3. This file can be imported into GRASS using r.in.shape.
4. (Example: r.in.shape in=no_grow out=no_grow).
5. In GRASS, use r.reclass to make a new road map with just class A1 roads.

   (Example: r.reclass roads out=interstate

   1 = 1 interstate

   2 thru 7 = NULL)

6. Use r.buffer to create a 50m buffer around the A1 roads, because growth closer than 50m to an interstate is highly unlikely.

   (Example: r.buffer interstate out=interstate_buffer dist=50 unit=meters)

7. In GRASS, merge the no growth shapefile and the interstate buffer file using r.mapcalc.

   (Example: r.mapcalc
   'nogrowth=if(isnull(interstate_buffer),if(no_grow),if(interstate_buffer))'

8. This will make a new map showing areas of EITHER 50m interstate buffer OR Federal/Indian/water lands.

Map 9. Digital elevation model (BR_DEM) GRASS version is raster file showing elevation of the area of interest. The map will contain the counties of interest but will also have extraneous land because of the way the DEM data was extracted (the file will not have the "cookie cutter" shape of the county boundary map, it will be a rectangular grid containing the counties of interest).

Creating Map 9 from scratch:

Source: USGS 1:24K DEM. The second URL (the National Map) is recommended:

http://edc.usgs.gov/geodata/

http://seamless.usgs.gov/

1. To obtain the DEM from The National Map,.

   Go to http://seamless.usgs.gov/ , zoom in to the area of interest

   Click on "NED" checkbox in upper right hand menu under Download Layers. Uncheck all other boxes.

   Use rectangular download tool (the icon is on the middle left hand side) and draw around the area of interest.

   The website should automatically begin downloading and might take awhile to finish.

2. Unzip the downloaded file and save to the appropriate directory.
3. Copy the finished grid file to UNIX environment using WinSCP.
4. This grid can be imported into GRASS using r.in.gdal or r.in.arc.

   (Example: r.in.gdal in=dem out=dem)

   Warning: the grid file may have an eight digit name assigned to it due to the National Map conventions.

Creating mLEAM Input Maps via GRASS Makefile:

From the nine base maps created above, the GRASS makefile can automatically produce the mLEAM input maps. The code for the GRASS makefile is shown in Appendix C.

Table 2. mLEAM input files made by GRASS makefile.

Map	Description
Boundary	Boundary of study region (usually a group of counties)
Land Cover	Land cover of study region (NLCD data)
Metrobuffer	Buffer around municipal boundaries in study region (typ 1.5 miles)
No Growth	Areas in study region in which development is not possible or highly unlikely
Slope	Slope of terrain in study region
Forest Attractor	Distance to nearest forest
Water Attractor	Distance to nearest water body
Highway Attractor	Travel time to nearest highway
Intersection Attractor	Travel time to nearest major road intersection
Ramp Attractor	Travel time to nearest highway ramp
Road Attractor	Travel time to nearest road
Cities Attractor (large, medium, small)	Travel time to nearest city (divided into three groups—small, medium, and large)

Create Alternative Scenarios

A new loop highway, running from NW Fayetteville along the edge of Fort Bragg and connecting with I-95 again in SW Fayetteville, was scheduled to be built in the near future. The CAD file for this highway was provided by Fort Bragg, and then digitized into GIS using ArcMap 8.3. This road was then added to the original road network map, which resulted in a secondary road network map; this acted as a base map for a different mLEAM growth scenario. Two different possibilities for the area emerged: (1) without the new highway, or (2) with the new highway.

Table 3. Additional mLEAM input files made by GRASS makefile due to new highway.

Map	Description
Cities Attractor (large, medium, small)	Travel time to nearest city (divided into three groups—small, medium, and large)
Highway Attractor 2	Travel time to nearest highway
Intersection Attractor 2	Travel time to nearest major road intersection
Ramp Attractor 2	Travel time to nearest highway ramp
Road Attractor 2	Travel time to nearest road

Run mLEAM

The resulting mLEAM input maps can be put into the LEAM software to produce the final mLEAM output models. The results of running this model are shown in Figures 28 to 31. The two resultant maps showed growth for the two different road scenarios: (1) without the new highway, and (2) with the new highway.

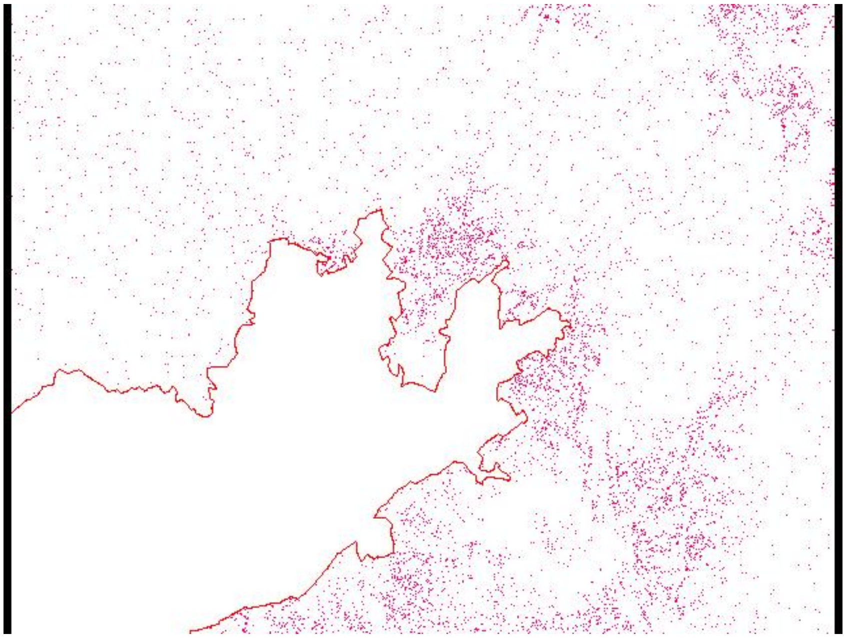


Figure 28. futureLanduse@scenarioBase.

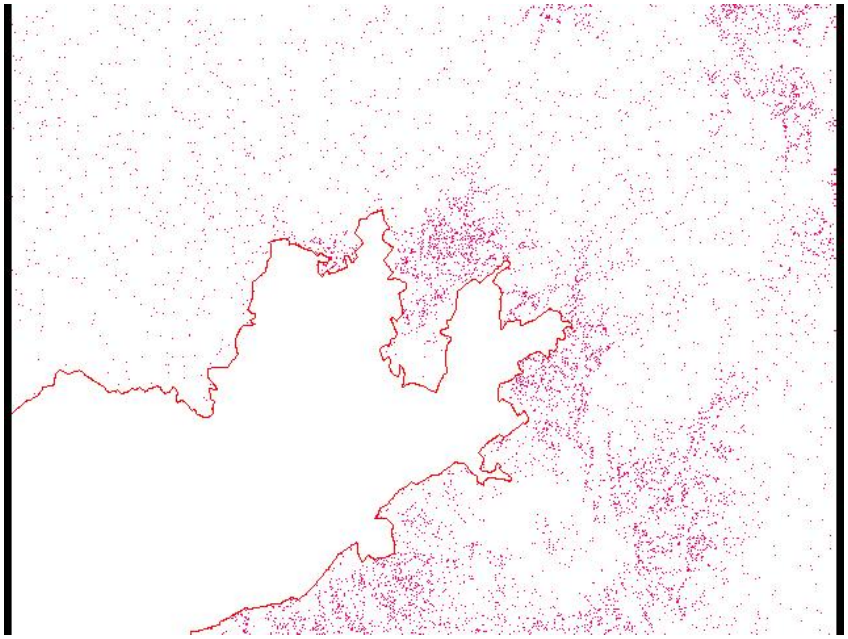


Figure 29. futureLanduse@scenarioRoad.

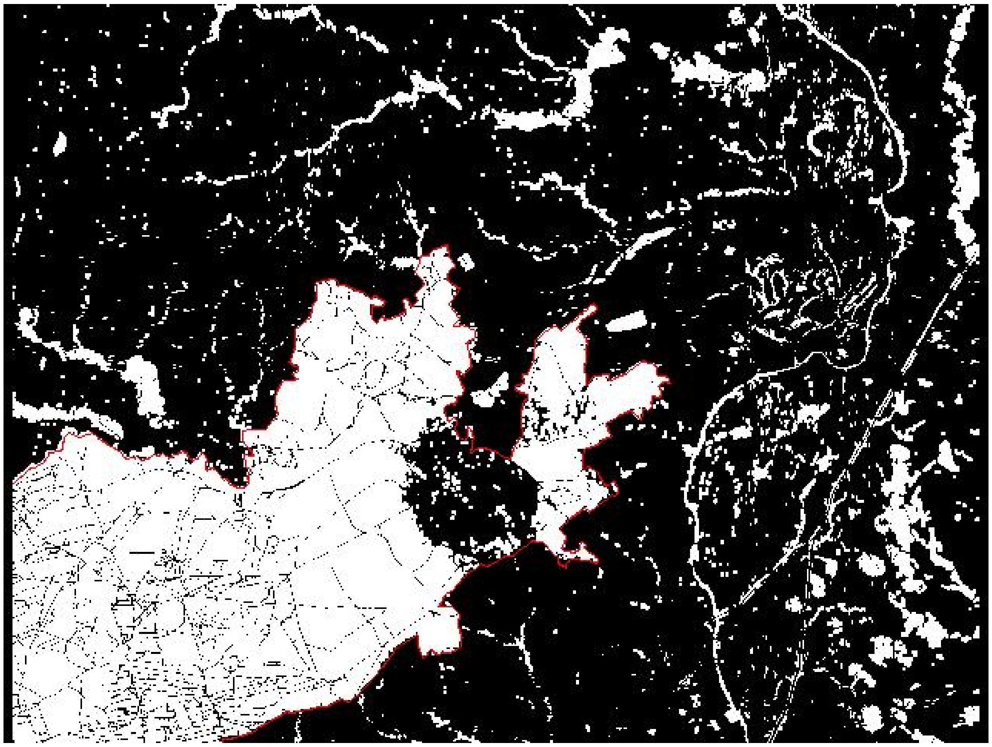


Figure 30. ATTRACTOR_RES@scenarioBase.

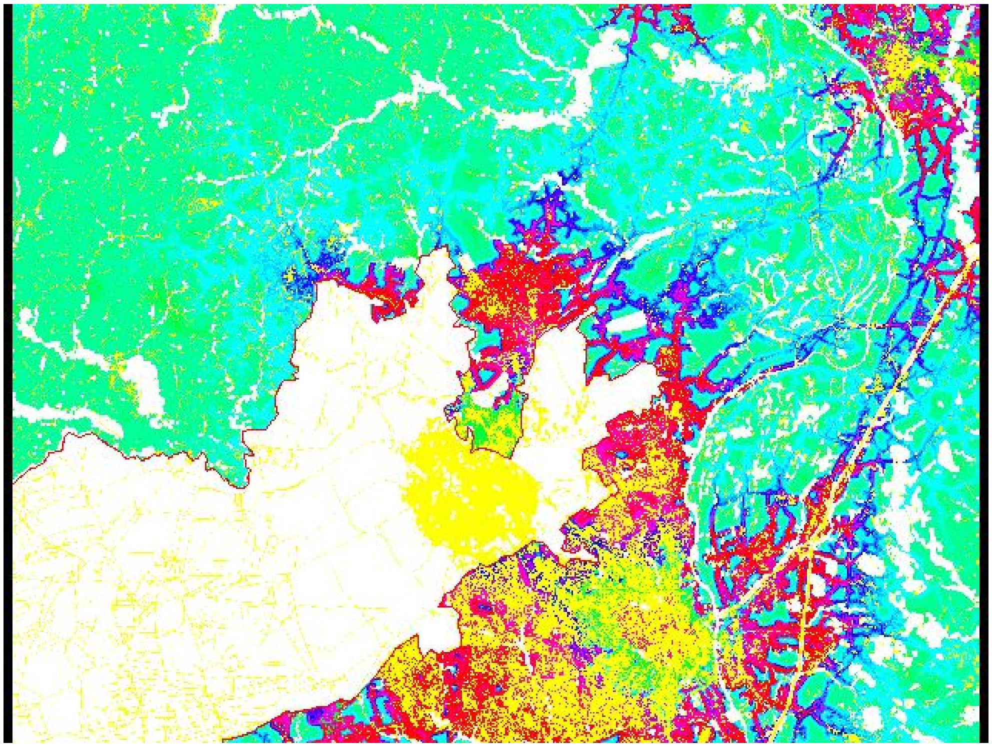


Figure 31. ATTRACTOR_RES@scenarioRoad.

Review and Compare Results

Each of the mLEAM outputs showed considerable growth of Fayetteville toward Fort Bragg, with development touching the border of the installation in many cases. Large growth was also predicted on the western edge of Fort Bragg, centered on the cities of Pinehurst and Southern Pines.

The growth patterns around the Fayetteville region, however, differed between the two scenarios. The scenario without the new highway showed a more centralized growth pattern situated primarily along the main arteries going in and out of Fayetteville. The scenario with the new highway predicted more growth along this highway, resulting in much more growth closer to Fort Bragg and to the north of Fayetteville. This is expected, since residents usually desire to live close to roads that have quick access to and from their places of work. A map was made (Figure 32) highlighting the difference in the growth patterns between the two scenarios (this map was not made in the mLEAM program—it was done manually in a GIS). Figures 33 to 36 show the schematics for converting the map data.

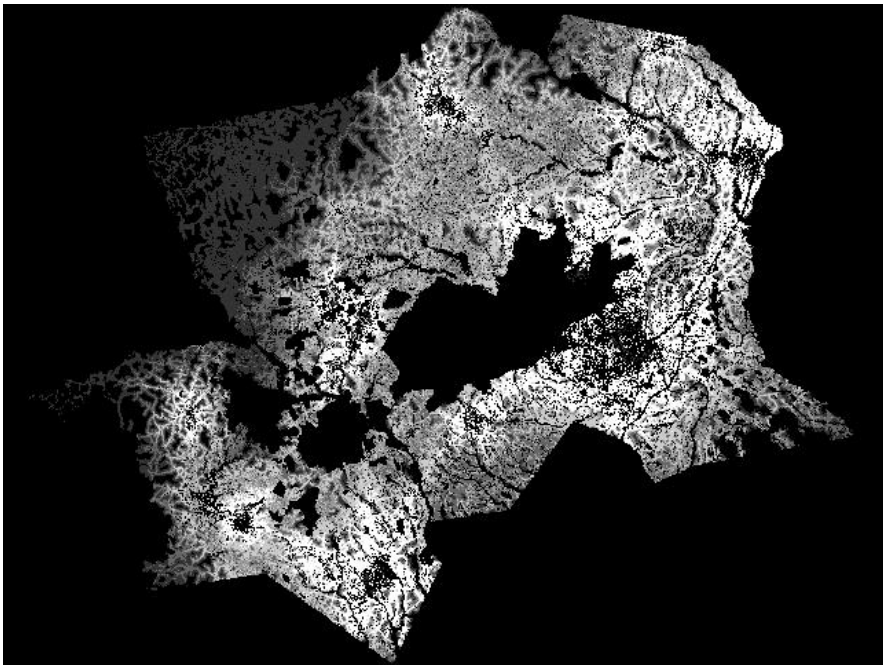


Figure 32. Difference in the growth patterns between two scenarios.

Summary of Data Sources

NC County boundary data:

http://www.census.gov/geo/www/cob/co2000.html

USGS NLCD data for NC:

http://edcsgs9.cr.usgs.gov/pub/data/landcover/states

NC roads by county:

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

NC municipal boundaries:

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

Census population NC data:

http://www.census.gov

NC Federal & Indian lands:

http://www.nationalatlas.gov

NC landmark polygons:

http://arcdata.esri.com/data/tiger2000/tiger_download.cfm

NC environmental lands: JLUS CD (provided by Fort Bragg area commission)
USGS 1:24K DEM:

http://seamless.usgs.gov

New highway: provided by Fort Bragg, digitized by Kyle Brock

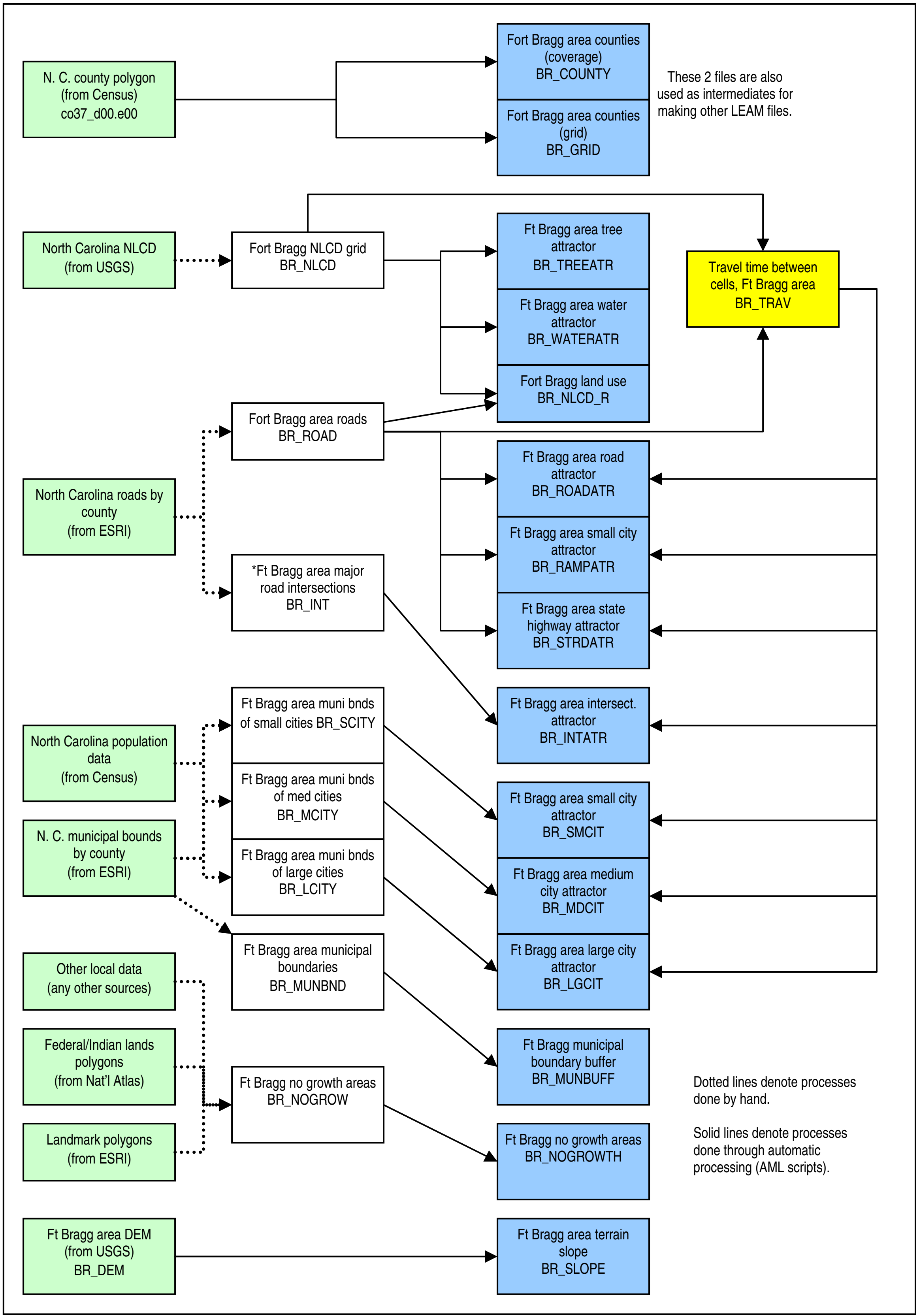
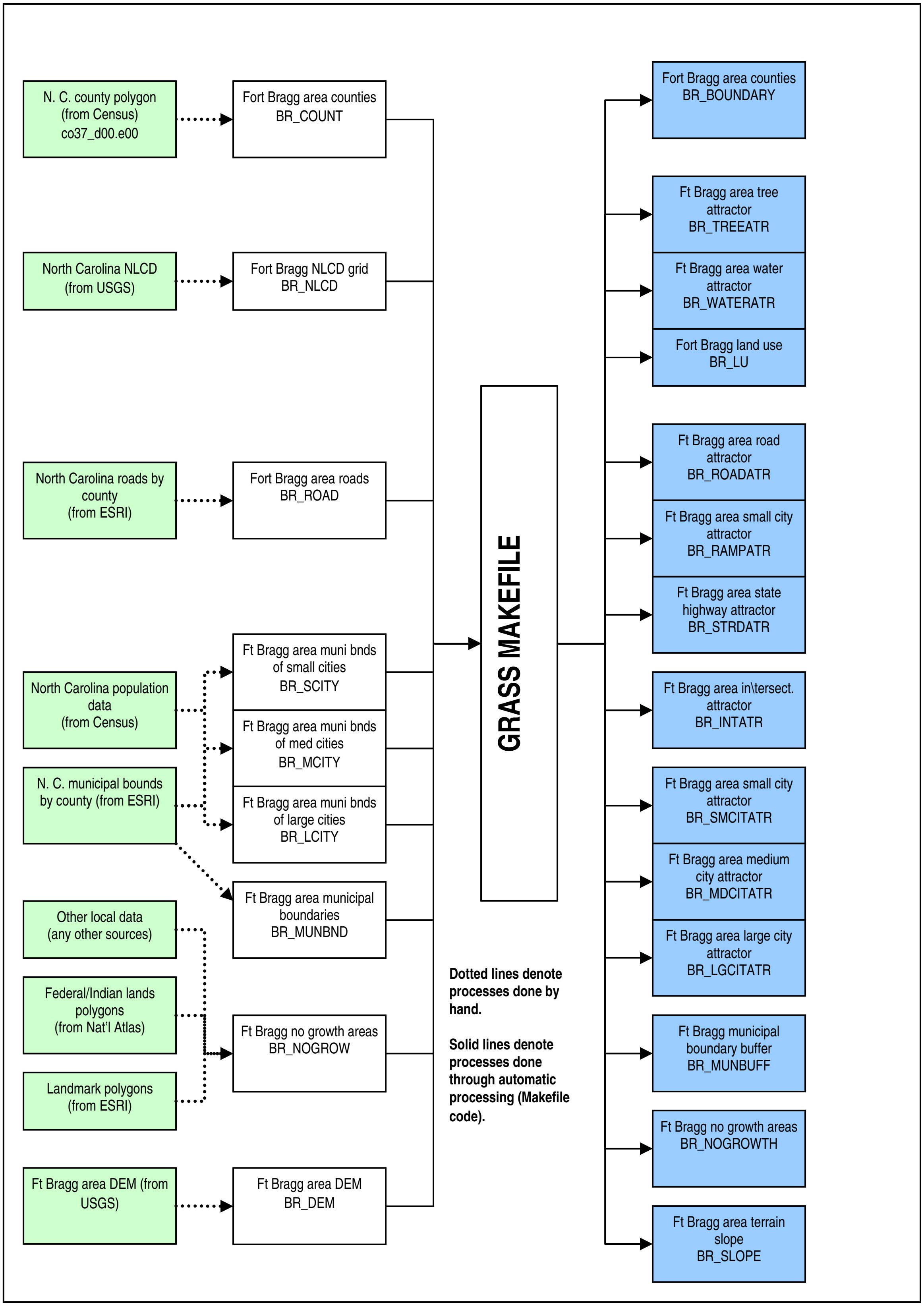

N. C. county polygon
(from Census)
co37_d00.e00
Fort Bragg area counties
(coverage)
BR_COUNTY
Fort Bragg area counties
(grid)
BR_GRID
These 2 files are also used as intermediates for making other LEAM files.
North Carolina NLCD
(from USGS)
Fort Bragg NLCD grid
BR_NLCD
Ft Bragg area tree attractor
BR_TREEATR
Ft Bragg area water attractor
BR_WATERATR
Fort Bragg land use
BR_NLCD_R
Travel time between cells, Ft Bragg area
BR_TRAV
Fort Bragg area roads
BR_ROAD
North Carolina roads by county
(from ESRI)
*Ft Bragg area major road intersections
BR_INT
Ft Bragg area road attractor
BR_ROADATR
Ft Bragg area small city attractor
BR_RAMPATR
Ft Bragg area state highway attractor
BR_STRDATR
Ft Bragg area intersect. attractor
BR_INTATR
North Carolina population data
(from Census)
N. C. municipal bounds by county
(from ESRI)
Ft Bragg area muni bnds of small cities BR_SCITY
Ft Bragg area muni bnds of med cities
BR_MCITY
Ft Bragg area muni bnds of large cities
BR_LCITY
Ft Bragg area small city attractor
BR_SMCIT
Ft Bragg area medium city attractor
BR_MDCIT
Ft Bragg area large city attractor
BR_LGCIT
Ft Bragg area municipal boundaries
BR_MUNBND
Ft Bragg municipal boundary buffer
BR_MUNBUFF
Other local data
(any other sources)
Federal/Indian lands polygons
(from Nat'l Atlas)
Landmark polygons
(from ESRI)
Ft Bragg no growth areas
BR_NOGROW
Ft Bragg no growth areas
BR_NOGROWTH
Dotted lines denote processes done by hand.
Solid lines denote processes done through automatic processing (AML scripts).
Ft Bragg area DEM
(from USGS)
BR_DEM
Ft Bragg area terrain slope
BR_SLOPE

Figure 33. Schematic of data conversion using ArcGIS.


N. C. county polygon (from Census) co37_d00.e00
Fort Bragg area counties BR_COUNT
North Carolina NLCD (from USGS)
Fort Bragg NLCD grid BR_NLCD
North Carolina roads by county (from ESRI)
Fort Bragg area roads BR_ROAD
North Carolina population data (from Census)
N. C. municipal bounds by county (from ESRI)
Ft Bragg area muni bnds of small cities BR_SCITY
Ft Bragg area muni bnds of med cities BR_MCITY
Ft Bragg area muni bnds of large cities BR_LCITY
Ft Bragg area municipal boundaries BR_MUNBND
Other local data (any other sources)
Federal/Indian lands polygons (from Nat'l Atlas)
Landmark polygons (from ESRI)
Ft Bragg no growth areas BR_NOGROW
Ft Bragg area DEM (from USGS)
Ft Bragg area DEM BR_DEM
GRASS MAKEFILE
Fort Bragg area counties BR_BOUNDARY
Ft Bragg area tree attractor BR_TREEATR
Ft Bragg area water attractor BR_WATERATR
Fort Bragg land use BR_LU
Ft Bragg area road attractor BR_ROADATR
Ft Bragg area small city attractor BR_RAMPATR
Ft Bragg area state highway attractor BR_STRDATR
Ft Bragg area in\tersect. attractor BR_INTATR
Ft Bragg area small city attractor BR_SMCITATR
Ft Bragg area medium city attractor BR_MDCITATR
Ft Bragg area large city attractor BR_LGCITATR
Ft Bragg municipal boundary buffer BR_MUNBUFF
Ft Bragg no growth areas BR_NOGROWTH
Ft Bragg area terrain slope BR_SLOPE

Dotted lines denote processes done by hand.

Solid lines denote processes done through automatic processing (Makefile code).

Figure 34. Schematic of data conversion using Grass.

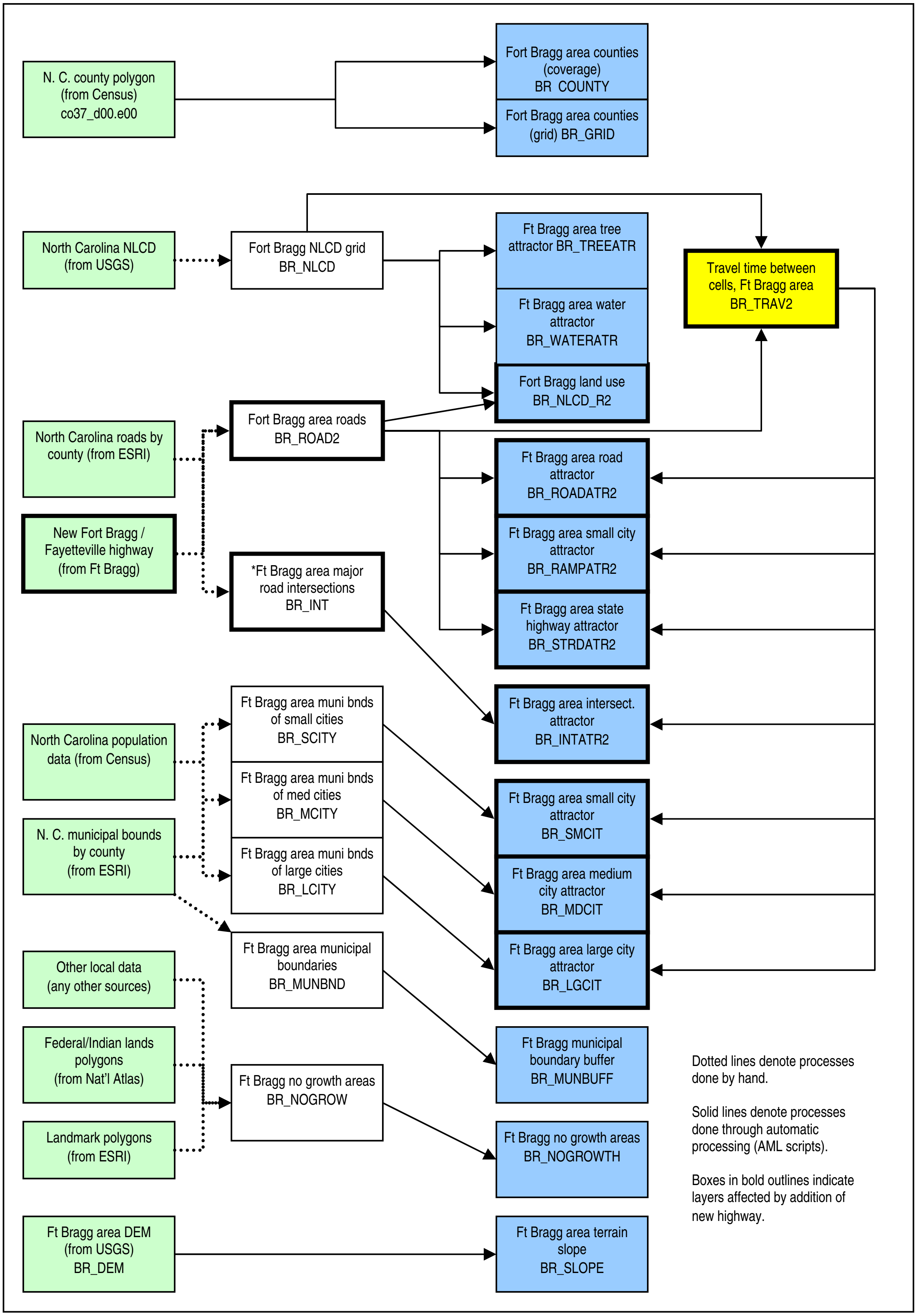
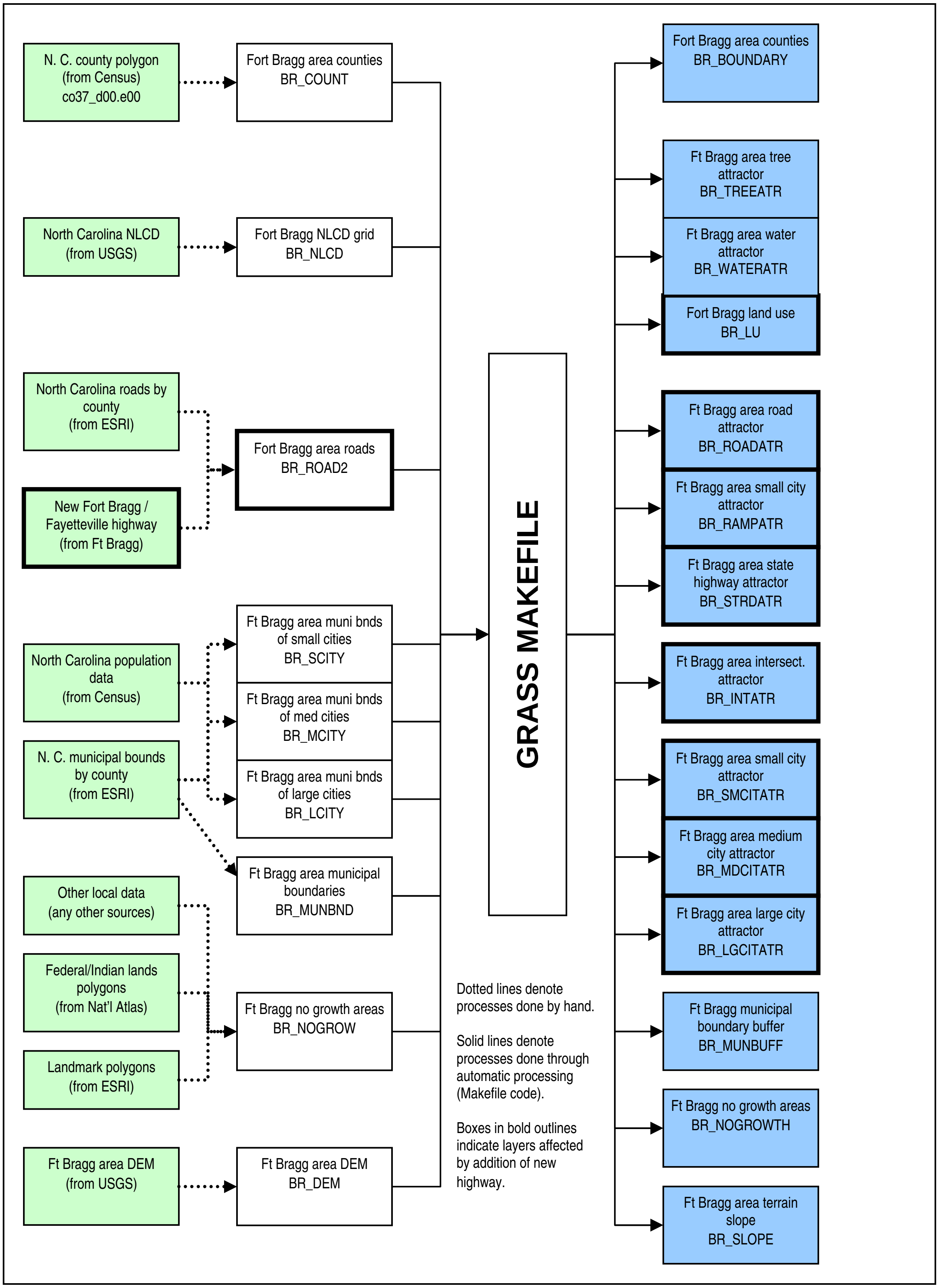

N. C. county polygon (from Census) co37_d00.e00
Fort Bragg area counties (coverage) BR_COUNTY
Fort Bragg area counties (grid) BR_GRID
North Carolina NLCD (from USGS)
Fort Bragg NLCD grid BR_NLCD
Ft Bragg area tree attractor BR_TREEATR
Ft Bragg area water attractor BR_WATERATR
Fort Bragg land use BR_NLCD_R2
Travel time between cells, Ft Bragg area BR_TRAV2
North Carolina roads by county (from ESRI)
Fort Bragg area roads BR_ROAD2
New Fort Bragg / Fayetteville highway (from Ft Bragg)
*Ft Bragg area major road intersections BR_INT
Ft Bragg area road attractor BR_ROADATR2
Ft Bragg area small city attractor BR_RAMPATR2
Ft Bragg area state highway attractor BR_STRDATR2
Ft Bragg area intersect. attractor BR_INTATR2
North Carolina population data (from Census)
N. C. municipal bounds by county (from ESRI)
Ft Bragg area muni bnds of small cities BR_SCITY
Ft Bragg area muni bnds of med cities BR_MCITY
Ft Bragg area muni bnds of large cities BR_LCITY
Ft Bragg area small city attractor BR_SMCIT
Ft Bragg area medium city attractor BR_MDCIT
Ft Bragg area large city attractor BR_LGCIT
Ft Bragg area municipal boundaries BR_MUNBND
Ft Bragg municipal boundary buffer BR_MUNBUFF
Other local data (any other sources)
Federal/Indian lands polygons (from Nat'l Atlas)
Landmark polygons (from ESRI)
Ft Bragg no growth areas BR_NOGROW
Ft Bragg no growth areas BR_NOGROWTH
Ft Bragg area DEM (from USGS) BR_DEM
Ft Bragg area terrain slope BR_SLOPE
Dotted lines denote processes done by hand.
Solid lines denote processes done through automatic processing (AML scripts).
Boxes in bold outlines indicate layers affected by addition of new highway.

Figure 35. Schematic of data conversion using Arc/Info only (new highway scenario).


N. C. county polygon (from Census) co37_d00.e00
Fort Bragg area counties BR_COUNT
North Carolina NLCD (from USGS)
Fort Bragg NLCD grid BR_NLCD
North Carolina roads by county (from ESRI)
New Fort Bragg / Fayetteville highway (from Ft Bragg)
Fort Bragg area roads BR_ROAD2
North Carolina population data (from Census)
N. C. municipal bounds by county (from ESRI)
Ft Bragg area muni bnds of small cities BR_SCITY
Ft Bragg area muni bnds of med cities BR_MCITY
Ft Bragg area muni bnds of large cities BR_LCITY
Ft Bragg area municipal boundaries BR_MUNBND
Other local data (any other sources)
Federal/Indian lands polygons (from Nat'l Atlas)
Landmark polygons (from ESRI)
Ft Bragg no growth areas BR_NOGROW
Ft Bragg area DEM (from USGS)
Ft Bragg area DEM BR_DEM
GRASS MAKEFILE
Fort Bragg area counties BR_BOUNDARY
Ft Bragg area tree attractor BR_TREEATR
Ft Bragg area water attractor BR_WATERATR
Fort Bragg land use BR_LU
Ft Bragg area road attractor BR_ROADATR
Ft Bragg area small city attractor BR_RAMPATR
Ft Bragg area state highway attractor BR_STRDATR
Ft Bragg area intersect. attractor BR_INTATR
Ft Bragg area small city attractor BR_SMCITATR
Ft Bragg area medium city attractor BR_MDCITATR
Ft Bragg area large city attractor BR_LGCITATR
Ft Bragg municipal boundary buffer BR_MUNBUFF
Ft Bragg no growth areas BR_NOGROWTH
Ft Bragg area terrain slope BR_SLOPE

Dotted lines denote processes done by hand.

Solid lines denote processes done through automatic processing (Makefile code).

Boxes in bold outlines indicate layers affected by addition of new highway.

Figure 36. Schematic of data conversion using ArcGIS and Grass (new highway scenario).

5 Conclusion

Prediction of urban patterns is accomplished using the Land Use Change (LUC) model, which is part of a suite of tools called the military Land use Evolution and Assessment Model (mLEAM). This work has provided information to assist GIS technicians in preparing input maps for the mLEAM LUC model, and also descriptions of the specific steps used to create maps for the multi-county area surrounding Fort Bragg, NC. This provides installation planners with better tools to:

1. Predict urban growth patterns that might occur in response to proposed regional plans involving certain investments and policies, and
2. Predict the change in training and testing opportunity that might occur as a result of those patterns.

Appendix A: CD Contents and Directory Listing

CD—Fort Bragg

The CD for Fort Bragg mLEAM includes all files associated with the production of the mLEAM scenario. Table A1 lists the files. (You will need to unzip (uncompress) the file on the CD to read the files.

File Folders (The files in these folders are largely support or starter files for the maps shown in the table below. There is little reason to look within these folders):

- calc
- Clip
- Tiger_files
- Info.

Table A1. Files associated with the production of the mLEAM scenario.

File name	Description	How obtained
br_county	Polygon coverage of 7-county area around Fort Bragg. Includes Cumberland, Harnett, Hoke, Lee, Moore, Richmond, and Scotland Counties.	AML script “1bnd_county.aml”
co37_d00	North Carolina county boundaries, projected Albers	internet
55687211	Bragg area DEM—Western half	internet
87615094	Bragg area DEM—Eastern half	internet
92088612	Bragg area DEM—entire	internet
nc_nlcd	NLCD for North Carolina	manipulated from Tiff
north_carolina_ NLCD_erd04220.tif	Tiff image of NLCD of North Carolina	internet
br_dem	Bragg area DEM	manipulated from 8-digit DEM
br_demjunk2	Bragg area DEM, clipped with counties	manipulated from br_dem and br_county
br_demmerge	Bragg area DEM, entire	combination of W and E halves

File name	Description	How obtained
br_nlcd	NLCD data for Fort Bragg area	AML script “1bnd_grid.aml”
brgrid	Raster grid of 7-county area	“1bnd_grid.aml”
br_lu	NLCD data for Fort Bragg area, with major roads merged	“2landuse_new.aml”
br_buf	7-county area—1.5 mi buffer around municipal boundaries	“3munbuf.aml”
br_trav	Travel time across grid cell; “travel time” is based on the number of minutes it takes to travel across one grid cell, and is based solely on that cell’s NLCD land cover category and/or existence of roads	“4travel.aml”
br_roadatr	Travel time to nearest road	“6roadatr.aml”
br_lgcit	Travel time to nearest large city	“8cityatr.aml”
br_mdit	Travel time to nearest medium sized city	“8cityatr.aml”
br_smcit	Travel time to nearest small city	“8cityatr.aml”
br_nogrowth	No growth areas in Fort Bragg area	“9nogrow.aml”
br_rampatr	Travel time to nearest highway ramp	“10rampatr.aml”
br_intatr	Travel time to nearest major intersection	“1lrdintatr.aml”
br_slope	Physical slope of each grid cell	“12slope.aml”
br_stateatr	Travel time to nearest state highway	“13stateatr.aml”
br_treeatr	Distance to nearest forest	“14treeatr.aml”
br_wateratr	Distance to nearest water/wetland	“15wateratr.aml”
a2a3road.shp	Major roads, used in determining intersections	manipulated from br_road.shp using ArcView 3.2
br_lcity.shp	Boundaries of large cities within Fort Bragg area—shapefile	manipulated from br_munbnd.shp
br_mcity.shp	Boundaries of medium sized cities within Fort Bragg area—shapefile	manipulated from br_munbnd.shp
br_scity.shp	Boundaries of small cities within Fort Bragg area—shapefile	manipulated from br_munbnd.shp
br_munbnd.shp	Boundaries of cities within Fort Bragg area	combined from Census Tiger files
nogrowth.shp	No growth areas in Fort Bragg area	combined from various sources
br_road.shp	All roads in Fort Bragg area	combined from Census Tiger files
Bragg_bnd.shp	Boundary of Fort Bragg	selected from U.S. military base shapefile
Bragg_county.shp	7-county area, projected Albers	manipulated from co37_d00
brng.shp	No growth areas in Fort Bragg area	combined from various sources
cn_a2a1.shp	Major roads, used in determining intersections	calculated in ArcView 3.2
nd_a2a1.shp	Major road intersections	calculated in ArcView 3.2

File name	Description	How obtained
nd_cn_2.shp	Major road nodes (intersections, bends)	calculated in ArcView 3.2
fedlanp020.shp	Federal & Indian lands	internet
nc_mil.shp	Fort Bragg and Camp Mackall boundaries	manipulated from managed.shp
nc_state.shp	N. Carolina state outline polygon	internet
lgcity1.shp	Boundaries of major cities in Fort Bragg area (similar but not the same as br_lcity.shp)	manipulated from br_munbnd.shp
managed.shp	Managed environmental lands within 7-county area	combined from files on a CD donated by Fort Bragg planning commission
br_count.e00	7-county area—interchange file	“1bnd_cover.aml”
br_road1.e00	Bragg area roads, not projected—interchange file	
br_roadnew.e00	Bragg area roads, projected Albers—interchange file	
co37_d00.e00	NC counties—interchange file	internet
AML scripts		
1bnd_cover.aml		
1bnd_grid.aml		
2landuse_new.aml		
3munbuf.aml		
4travel.aml		
6roadatr.aml		
8cityatr.aml		
9nogrow.aml		
10rampatr.aml		
11rdintatr.aml		
12slope.aml		
13stateatr.aml		
14treeatr.aml		
15wateratr.aml		
Support files for AML scripts		
boundremap		
boundremap2		
cityremap		
facilityremap		
fldremap		
growremap		
munbndremap		
nogrowremap		
roadremap		

File name	Description	How obtained
travelremap1		
travelremap2		
travelremap3		
travelremap4		
travelremap5		
astrgrdrp.txt		
astrgrdrp2.txt		

Appendix B: AML Scripts

```
1bnd_cover.aml
/*************************************************************
/* 1bnd_cover.aml                                            *
/*                                                           *
/* This AML will import the *.e00 file as cover and         *
/* select counties of interest with appropriate projection  *
/*                                                           *
/* Made by Yong Wook Kim on Dec 16, 2002,                    *
/* modified by Woonsup Choi on June 24, 2003                 *
/*************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&setvar input_exchangein = [response 'Enter a exchange file name without
extension']
&setvar cty_bnd = [response 'Enter a name of the resulting cover']
&setvar input_exchange = [locase %input_exchangein%]

&TYPE
&Type All THE EXISTING COVERAGE OR GRID NAMED "%input_exchange%,”
"%input_exchange%1,” "%input_exchange%2,” "%cty_bnd%" WILL BE DELETED !!!
&TYPE
&setvar delete_yesnoin = [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You should type either 'y' or 'n'

&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

/* &if ^ [exists %tifname% -file] &then
/* &if [exists [value tempprj%prj%] -file] &then

/*&setvar input_unitin:=[response 'Is the unit of input file feet or meter?
(f/m)']
/*&setvar input_unit = [locase %input_unitin%]

/*&if %input_unit% ne:=f and %input_unit% ne:=m &then
/* &return ERROR! You have to type either 'f' of 'm' for the unit.

&if [exists %input_exchange% -COVER] or [exists %input_exchange% -GRID]
&then
   kill %input_exchange% all
&if [exists %input_exchange%1 -COVER] or [exists %input_exchange%1 -GRID]
&then
   kill %input_exchange%1 all
&if [exists %input_exchange%2 -COVER] or [exists %input_exchange%2 -GRID]
&then
   kill %input_exchange%2 all
&if [exists %cty_bnd% -COVER] or [exists %cty_bnd% -GRID] &then
   kill %cty_bnd% all
&if [exists %cty_bnd%1 -COVER] or [exists %cty_bnd%1 -GRID] &then
   kill %cty_bnd%1 all
```

```
/********************************************
/* import ArcInfo exchange file to Coverage *
/********************************************
import cover %input_exchange%.e00 %input_exchange%

/****************************************************
/* projecting the Coverage to the target projection *
/****************************************************
project cover %input_exchange% %input_exchange%1
output
projection albers
datum nad83
spheroid grs80
units meters
parameters
29 30 00
45 30 00
-96 00 00
23 00 00
0
0
end
clean %input_exchange%1

/***********************************************
/* renaming the cover                  *
/***********************************************
kill %input_exchange%
rename %input_exchange%1 %input_exchange%

/*********************************************
/* selecting counties of interest        *
/*********************************************

reselect %input_exchange% %cty_bnd% poly

1bnd_grid.aml
/*************************************************************
/* 1bnd_grid.aml                              *
/*                                        *
/* This AML will change the coverage to GRID            *
/*                                        *
/* Made by Yong Wook Kim on Dec 16, 2002,             *
/* modified by Woonsup Choi on July 03, 2003            *
/*************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&setvar input_exchangein = [response 'Enter a county boundary cover name']
&setvar input_snlcd = [response 'Enter a state NLCD grid name you have']
/* You must have a state NLCD grid converted from TIFF file

&setvar input_bnlcd = [response 'Enter a boundary NLCD grid name you want']
/* This is NLCD of the study area. This will be used to define boundary grid
cell alignment.

&setvar cty_bnd = [response 'Enter a name of the resulting grid']
&setvar input_exchange = [locase %input_exchangein%]

&TYPE
&Type All THE EXISTING COVERAGE OR GRID NAMED "%input_exchange%1,”
"%input_exchange%2,” "%input_bnlcd%,” "%cty_bnd%,” "%cty_bnd%"_1 WILL BE
DELETED !!!
&TYPE
```

```
&setvar delete_yesnoin = [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You should type either 'y' or 'n'

&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

&if [exists %input_exchange%1 -COVER] or [exists %input_exchange%1 -GRID]
&then
   kill %input_exchange%1 all
&if [exists %input_exchange%2 -COVER] or [exists %input_exchange%2 -GRID]
&then
   kill %input_exchange%2 all
&if [exists %input_bnlcd% -COVER] or [exists %input_bnlcd% -GRID] &then
   kill %input_bnlcd% all
&if [exists %cty_bnd% -COVER] or [exists %cty_bnd% -GRID] &then
   kill %cty_bnd% all
&if [exists %cty_bnd%1 -COVER] or [exists %cty_bnd%1 -GRID] &then
   kill %cty_bnd%1 all
&if [exists %cty_bnd%_1 -COVER] or [exists %cty_bnd%_1 -GRID] &then
   kill %cty_bnd%_1 all

/*polygrid %input_exchange% %cty_bnd%1
/*30
/*y
grid
gridclip %input_snlcd% %input_bnlcd% cover %input_exchange%
%cty_bnd% = %input_bnlcd% * 0 + 1
q
gridascii %cty_bnd% %cty_bnd%.asc

/**********************************************
/*  exporting the cover as e00            *
/**********************************************
export cover %input_exchange% %input_exchange%.e00

2landuse_new.aml
/*--------------------------------------------------------------------------
-------------------------------
/*  YWKIM's AML for Fort Bragg
/*--------------------------------------------------------------------------
-------------------------------
/*  Program: 2landuse_new.aml
/*  Purpose: Convert the ESRI street cover to GRID including A1-A3, and A6.
/*--------------------------------------------------------------------------
-------------------------------
/*  Usage: 2landuse_new <INSTR_SHP> <INLU_GRID> <INMUN_SHP> <OUT_GRID>
/*
/*  Arguments: <INSTR_SHP> - INPUT SHAPEFILE THAT CLIPPPED FROM ESRI STREET
MAP
/*         <INLU_GRID> - INPUT NLCD LANDUSE GRID of the study area -- should
be clipped from the state data
/*         <OUT_GRID> - OUTPUT GRID
/*       <INMUN_SHP> - INPUT MUNICIPAL BOUNDARY FROM TIGER
/*--------------------------------------------------------------------------
-------------------------------
/*  Prerequisite: Street Coverage converted from the ESRI street map
/*  Process: Clip the shapefile, shapearc, then routearc
/*--------------------------------------------------------------------------
-------------------------------

/************************
/* basic parameter setup *
```

```
/*************************
&args incover lugrid inmun outgrid

&if [NULL %incover%] = .TRUE. &then
&return &inform Usage: ASTRGRD <INSTR_SHP> <INLU_GRID> <INMUN_SHP>
<OUT_GRID>

&if [NULL %outgrid%] = .TRUE. &then
&return &inform Usage: ASTRGRD <INSTR_SHP> <INLU_GRID> <INMUN_SHP>
<OUT_GRID>

&if ^ [exists %incover%.shp -file] &then
   &return ERROR! The SHAPEFILE %incover% does not exist.
&if ^ [exists %lugrid% -grid] &then
   &return ERROR! The GRID %lugrid% does not exist.
&if ^ [exists %inmun%.shp -file] &then
   &return ERROR! The SHAPEFILE %inmun% does not exist.
&if [exists %incover% -COVER] or [exists %incover% -GRID] &then
   &return ERROR! The COVERAGE or GRID %incover% already exist.
&if [exists astrgrd -COVER] or [exists astrgrd -GRID] &then
   &return ERROR! The COVERAGE or GRID astrgrd already exist.
&if [exists astrgrd1 -COVER] or [exists astrgrd1 -GRID] &then
   &return ERROR! The COVERAGE or GRID astrgrd1 already exist.
&if [exists astrgrd2 -COVER] or [exists astrgrd2 -GRID] &then
   &return ERROR! The COVERAGE or GRID astrgrd2 already exist.
&if [exists astrgrd3 -COVER] or [exists astrgrd3 -GRID] &then
   &return ERROR! The COVERAGE or GRID astrgrd3 already exist.
&if [exists astrgrd4 -COVER] or [exists astrgrd4 -GRID] &then
   &return ERROR! The COVERAGE or GRID astrgrd4 already exist.
&if [exists %outgrid% -COVER] or [exists %outgrid% -GRID] &then
   &return ERROR! The COVERAGE or GRID %outgrid% already exist.

/**************************************
/* converting road shapefile to cover *
/**************************************
SHAPEARC %incover%.shp astrgrd1 order
ROUTEARC astrgrd1 order astrgrd2
KILL astrgrd1

/*****************************
/* converting the projection *
/*****************************
PROJECT COVER astrgrd2 %incover%
input
projection geographic
units dd
datum nad83
spheroid grs80
parameters
output
projection albers
units meters
datum nad83
spheroid grs80
parameters
29 30 00
45 30 00
-96 00 00
23 00 00
0
0
end

KILL astrgrd2 ALL
```

```
/******************************************
/* convert municipal boundary to coverage *
/******************************************
SHAPEARC %inmun%.shp astrgrd4
CLEAN astrgrd4
PROJECT COVER astrgrd4 astrgrd5
input
projection geographic
units dd
datum nad83
spheroid grs80
parameters

output
projection albers
units meters
datum nad83
spheroid grs80
parameters
29 30 00
45 30 00
-96 00 00
23 00 00
0
0
end

KILL astrgrd4
CLEAN astrgrd5

/**********************************************
/* reselect the A6, A1 - A3 from the coverage *
/**********************************************
RESELECT %incover% astrgrd1 LINE
res cfcc2 = 'A1' OR cfcc2 = 'A2' OR cfcc2 = 'A3' OR cfcc = 'A63'
~
n
n

/*************************************
/* convert the road coverage to grid *
/*************************************
LINEGRID astrgrd1 astrgrd2
30
y
NODATA

GRID
astrgrd = (astrgrd2 * 0) + 24
Q
KILL astrgrd2

/*************************************************
/* create 70m GIRD along the new street coverage *
/*************************************************
BUFFER astrgrd1 astrgrd2 # # 70 # LINE
KILL astrgrd1 ALL

/*******************************************
/* clip the buffer with municipal boundary *
/*******************************************

ERASE astrgrd2 astrgrd5 astrgrd1
KILL astrgrd2
CLEAN astrgrd1 astrgrd2
```

```
KILL astrgrd1
KILL astrgrd5

/*******************
/* convert to grid *
/*******************
POLYGRID astrgrd2 astrgrd1 inside
30
Y

KILL astrgrd2 ALL

GRID
astrgrd2 = select(astrgrd1, 'value = 100')

KILL astrgrd1 ALL

/**********************************
/* separate the NLCD landuse grid *
/**********************************
astrgrd3 = select(%lugrid%, 'value = 21 OR value = 22')
astrgrd4 = select(%lugrid%, 'value = 23')
astrgrd1 = merge(astrgrd3, astrgrd2, astrgrd4)

KILL astrgrd2 ALL
KILL astrgrd3 ALL
KILL astrgrd4 ALL

astrgrd2 = reclass(astrgrd1, astrgrdrp.txt, NODATA)
astrgrd3 = reclass(%lugrid%, astrgrdrp2.txt, DATA)
astrgrd4 = merge(astrgrd2, astrgrd3)

KILL astrgrd1 ALL
KILL astrgrd2 ALL
KILL astrgrd3 ALL

%outgrid% = merge(astrgrd, astrgrd4)
KILL astrgrd4 ALL
KILL astrgrd ALL
Q
KILL %incover%

3munbuf.aml
/*********************************************************************
/* 3munbuf.aml                                     *
/*                                        *
/* This AML will import the shape file and                  *
/* change it to Cover/GRID with appropriate projection         *
/* then create 1.5mil buffer with it                 *
/*                                        *
/* Made by Yong Wook Kim on Jan 7, 2003                    *
/* Modified by Woonsup Choi on June 25, 2003                 *
/*********************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&sv input_exchangein [response 'Enter a municipal boundary shape file name
without extension']
&sv input_exchange = [locase %input_exchangein%]
&sv input_bound [response 'Enter the boundary GRID name']
&sv input_bnd_exchange [response 'Enter the BOUNDARY cover name']
&sv x [response 'Enter the left end coordinate of the boundary grid']
&sv y [response 'Enter the bottom end coordinate of the boundary grid']
&sv nrows [response 'Enter the # of rows of the boundary grid']
```

```
&sv ncols [response 'Enter the # of columns of the boundary grid']

/* &if %input_type% ne c and %input_type% ne g &then
/*  &return ERROR! You have to type either 'c' for Coverage or 'g' for GRID
for the type.
/* &if %input_type% = c &then
   &sv input_type = COVER
/* &if %input_type% = g &then
/*  &sv input_type = GRID

/* &sv change_grid = n

/* &if %input_type% = COVER &then
/*   &do
/*  &sv change_grid [response 'Do you want to change the output buffer to
GRID? (y/n)']
/*      &sv change_grid = [locase %change_grid%]
/*        &if %change_grid% ne y and %change_grid% ne n &then
/*     &return ERROR! You have to type either 'y' or 'n'
/*   &end

&TYPE
&Type All THE EXISTING COVERAGE OR GRID NAMED "%input_exchange%,”
"%input_exchange%1,” "%input_exchange%buf,” "%input_bound%1,”
"%input_bound%2" WILL BE DELETED !!!
&TYPE
&sv delete_yesnoin [response 'Is it oaky for you? (y/n)' ]
&sv delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne n and %delete_yesno% ne y &then
   &return ERROR! You have to type either 'y' or 'n'

&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

/* &sv project_yesnoin [response 'Do you want to perform the projection
change (y/n)?']
/* &sv project_yesno = [locase %project_yesnoin%]

/* &sv input_unitin [response 'Is the unit of input file feet or meter?
(f/m)']
/* &sv input_unit = [locase %input_unitin%]

/* &if %input_unit% ne f and %input_unit% ne m &then
/*  &return ERROR! You have to type either 'f' of 'm' for the unit.

&if [exists %input_exchange% -COVER] or [exists %input_exchange% -GRID]
&then
   kill %input_exchange% all
&if [exists %input_exchange%1 -COVER] or [exists %input_exchange%1 -GRID]
&then
   kill %input_exchange%1 all
&if [exists %input_exchange%buf -COVER] or [exists %input_exchange%buf -
GRID] &then
   kill %input_exchange%buf all
/*&if [exists %input_bound% -COVER] or [exists %input_bound% -GRID] &then
/*  kill %input_bound% all
&if [exists %input_bound%1 -COVER] or [exists %input_bound%1 -GRID] &then
   kill %input_bound%1 all
&if [exists %input_bound%2 -COVER] or [exists %input_bound%2 -GRID] &then
   kill %input_bound%2 all

/*****************************************************************
/*  Import boundary exchange file as a cover and covert to a grid *
/*****************************************************************
```

```
/*import cover %input_bnd_exchange%.e00 %input_bound%
/*polygrid %input_bound% %input_bound%1
/*30
/*y
/*rename %input_bound%1 %input_bound%

/********************************************
/* changing a shape file to poly Coverage  *
/********************************************
shapearc %input_exchange%.shp %input_exchange% subclass
clean %input_exchange% %input_exchange%1

kill %input_exchange%
rename %input_exchange%1 %input_exchange%

/******************************
/* projecting the output file *
/******************************

   projectdefine %input_type% %input_exchange%
   projection geographic
   datum nad83

   spheroid grs80
   units dd
   parameters

project %input_type% %input_exchange% %input_exchange%1
   output
   projection albers
   datum nad83
   spheroid grs80
   units meters
   parameters
   29 30 00
   45 30 00
   -96 00 00
   23 00 00
   0
   0
   end

   kill %input_exchange% all
   rename %input_exchange%1 %input_exchange%
   build %input_exchange%
/*    &end

/****************************
/* create a 1.5 mile buffer *
/****************************
/* &if %input_type% = COVER &then
/*   &do
   buffer %input_exchange% %input_exchange%buf # # 2414.016
/*   &end

/* &if %input_type% = GRID &then
/*   &do

/*   &end

/*******************************
/* change the coverage to GRID *
/*******************************
/* &if %change_grid% = y &then
/*   &do
/*    kill %input_exchange% all
   rename %input_exchange%buf %input_exchange%1
   polygrid %input_exchange%1 %input_exchange%buf
   30
   n
   %x%, %y%
   %nrows%, %ncols%

   kill %input_exchange%1 all
/*   &end



/****************************************
/* create temporary input boundary GRID *
/****************************************
COPY %input_bound% %input_bound%1
GRID
%input_bound%2 = reclass(%input_bound%1,boundremap2)
kill %input_bound%1 all

/********************
/* reclassify the GRID *
/********************
%input_exchange%1 = reclass(%input_exchange%buf,munbndremap)
kill %input_exchange%buf

/***************
/* merge GRIDS *
/***************
%input_bound%1 = merge(%input_exchange%1,%input_bound%2)
kill %input_bound%2
kill %input_exchange%1

/*****************
/* clipping GRID *
/*****************
gridclip %input_bound%1 %input_bound%2 cover %input_bnd_exchange%
kill %input_bound%1 all
q
rename %input_bound%2 %input_exchange%buf
gridascii %input_exchange%buf %input_exchange%buf.asc

4travel.aml
/***************************************************************
/* 4travel.aml                                 *
/*                                    *
/* This AML will produce Travelminute_Map GRID by using      *
/* Landuse GRID and roadmap exchange file               *
/* Projection will be changed too, upon request           *
/*                                    *
/* Made by Yong Wook Kim on Jan 10, 2003                *
/* Modified by Woonsup Choi on Feb 19, 2003             *
/***************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&sv output_name [response 'Enter the FINAL OUTPUT GRID name']

&sv input_exchange [response 'Enter the ROAD cover exchange file w/o
extension']

&sv value_item [response 'Enter the VALUE ITEM FIELD of the road exchange
file']
```

```
&sv input_lu [response 'Enter the name of the LANDUSE GRID']

&sv input_bound [response 'Enter the name of the BOUNDARY GRID']

/* &sv buffer_yesnoin [response 'Do you want to buffer the limited-access?
(y/n)']
/* &sv buffer_yesno = [locase %buffer_yesnoin%]

/* &if %buffer_yesno% = y &then
/*   &do
   &sv buffer_item [response 'Enter the class of the limited-
access(buffer)']
   &sv ramp_item [response 'Enter the class of the ramp']
/*   &end

&TYPE
&Type All THE EXISTING Coverage or GRID NAMED "%output_name%,”
"%input_exchange%,” "%input_exchange%1,” "%input_exchange%2,”
"%input_exchange%3,” "%input_exchange%4,” "%input_exchange%5,”
"%input_exchange%9,” "tspeed" WILL BE DELETED !!!
&TYPE

&setvar delete_yesnoin [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You have to type either 'y' of 'n' for the answer.
&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

/* &sv project_yesnoin [response 'Do you want to perform the projection
change (y/n)?']
/* &sv project_yesno = [locase %project_yesnoin%]

&if [exists %output_name% -COVER] or [exists %output_name% -GRID] &then
   kill %output_name% all
/* Pay attention HERE! This command can be hidden.
/*&if [exists %input_exchange% -COVER] or [exists %input_exchange% -GRID]
&then
/* kill %input_exchange% all
/*
&if [exists %input_exchange%1 -COVER] or [exists %input_exchange%1 -GRID]
&then
   kill %input_exchange%1 all
&if [exists %input_exchange%2 -COVER] or [exists %input_exchange%2 -GRID]
&then
   kill %input_exchange%2 all
&if [exists %input_exchange%3 -COVER] or [exists %input_exchange%3 -GRID]
&then
   kill %input_exchange%3 all
&if [exists %input_exchange%4 -COVER] or [exists %input_exchange%4 -GRID]
&then
   kill %input_exchange%4 all
&if [exists %input_exchange%5 -COVER] or [exists %input_exchange%5 -GRID]
&then
   kill %input_exchange%5 all
&if [exists %input_exchange%9 -COVER] or [exists %input_exchange%9 -GRID]
&then
   kill %input_exchange%9 all
&if [exists tspeed -COVER] or [exists tspeed -GRID] &then
   kill tspeed all
```

```
/**************************************************************************
/* convert shapefile to coverage                                      *
/* Pay attention here! This command can be hidden when you use a cover, not
an exchange file    *
/***************************************************************************
**********************

/*IMPORT cover %input_exchange%.e00 %input_exchange%
/*shapearc %input_exchange%.shp %input_exchange% subclass
/*routearc %input_exchange% subclass %input_exchange%1
/*kill %input_exchange%
/*rename %input_exchange%1 %input_exchange%

/********************************
/* projecting the road coverage *
/********************************
&sv input_type = COVER
/* &if %project_yesno% = y &then
/*   &do

   /*************************************************
   /* give the projection to the converted Coverage *
   /*************************************************
   projectdefine %input_type% %input_exchange%
   projection geographic
   datum nad83
   spheroid grs80
   units dd
   parameters

   /*****************
   /* projecting it *
   /*****************
   project %input_type% %input_exchange% %input_exchange%1
   output
   projection albers
   datum nad83
   spheroid grs80
   units meters
   parameters
   29 30 00
   45 30 00
   -96 00 00
   23 00 00
   0
   0
   end
   kill %input_exchange%

RESELECT %input_exchange%1 %input_exchange% line
res %value_item% <= 6 and %value_item% >= 1
~
n
n
kill %input_exchange%1 all

/**********************************************************
/* buffering limited-access hwys and converting to a grid *
/**********************************************************
/* &if %buffer_yesno% = y &then
/*   &do
   RESELECT %input_exchange% %input_exchange%1 line
   res %value_item% = %buffer_item%
   ~
   n
   n

   BUFFER %input_exchange%1 %input_exchange%2 # # 70 0.5 line flat
   POLYGRID %input_exchange%2 %input_exchange%4
   30
   y
   kill %input_exchange%2 all
   LATTICEPOLY %input_bound% %input_exchange%2 ASPECT
   LATTICECLIP %input_exchange%4 %input_exchange%2 %input_exchange%3
   kill %input_exchange%4 all
   kill %input_exchange%2 all

   /*************************************
   /* converting the interstate to GRID *
   /*************************************
   LINEGRID %input_exchange%1 %input_exchange%4 %value_item%
   30
   y
   NODATA
   kill %input_exchange%1 all

   /*******************************
   /* converting the ramp to GRID *
   /*******************************
   RESELECT %input_exchange% %input_exchange%1 line
   res %value_item% = %ramp_item%
   ~
   n
   n

LINEGRID %input_exchange%1 %input_exchange%5 %value_item%
   30
   y
   NODATA
   kill %input_exchange%1 all

/************************************************
/* converting the roads to grid            *
/************************************************
LINEGRID %input_exchange% %input_exchange%1 %value_item%
30
y
NODATA
/* kill %input_exchange% all

/*****************************************
/* merge road GRID and interstate buffer *
/*****************************************
GRID
/* buffer
%input_exchange%2 = reclass(%input_exchange%3,travelremap1)
kill %input_exchange%3 all

/* roads
%input_exchange%9 = int(%input_exchange%1)
kill %input_exchange%1 all
%input_exchange%1 = reclass(%input_exchange%9,travelremap2)
kill %input_exchange%9 all

/* limited-access hwy
%input_exchange%3 = reclass(%input_exchange%4,travelremap3)
kill %input_exchange%4 all

/* ramp
%input_exchange%4 = reclass(%input_exchange%5,travelremap4)
```

```
kill %input_exchange%5 all

%input_exchange%5 =
merge(%input_exchange%3,%input_exchange%4,%input_exchange%2,%input_exchange%
1)
kill %input_exchange%1 all
kill %input_exchange%2 all
kill %input_exchange%3 all
kill %input_exchange%4 all

/* landuse
%input_exchange%1 = reclass(%input_lu%,travelremap5)

%input_exchange%2 = merge(%input_exchange%5,%input_exchange%1)
kill %input_exchange%5 all
kill %input_exchange%1 all

%input_exchange%1 = 1000 / (26.8224 * %input_exchange%2)
rename %input_exchange%2 tspeed
rename %input_exchange%1 %output_name%

q

/* %output_name% = int(%input_exchange%)
/* kill %input_exchange%


6roadatr.aml
/****************************************************************
/* 6roadatr.aml                                 *
/*                                     *
/* This AML will produce CT ROAD ATTRACTOR MAP GRID        *
/* by using roadmap exchange file and Travel Minute Map       *
/* Projection will be changed too                     *
/*                                     *
/* Made by Yong Wook Kim on Jan 15, 2003                *
/* Modified by Woon-sup Choi on Jul 11, 2003             *
/****************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&sv x [response 'Enter the left end coordinate of the boundary grid']
&sv y [response 'Enter the bottom end coordinate of the boundary grid']
&sv nrows [response 'Enter the # of rows of the boundary grid']
&sv ncols [response 'Enter the # of columns of the boundary grid']
&sv output_name [response 'Enter the FINAL OUTPUT GRID name']
&sv input_exchange [response 'Enter the ROAD EXCHANGE FILE name without
extension']
&sv value_item [response 'Enter the VALUE ITEM FIELD of the road exchange
file']
&sv road_class [response 'Enter the CLASS of ROADS and STREETS']
&sv travel_map [response 'Enter the name of the TRAVEL MINUTE MAP']
&sv input_bound [response 'Enter the name of the BOUNDARY grid']

&TYPE
&Type All THE EXISTING Coverage or GRID NAMED "%output_name%,”
"%input_exchange%,” "%input_exchange%1,” "%input_exchange%2" WILL BE DELETED
!!!
&TYPE

&setvar delete_yesnoin [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
```

```
   &return ERROR! You have to type either 'y' of 'n' for the answer.
&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

&if [exists %output_name% -COVER] or [exists %output_name% -GRID] &then
   kill %output_name% all
&if [exists %input_exchange% -COVER] or [exists %input_exchange% -GRID]
&then
   kill %input_exchange% all
&if [exists %input_exchange%1 -COVER] or [exists %input_exchange%1 -GRID]
&then
   kill %input_exchange%1 all
&if [exists %input_exchange%2 -COVER] or [exists %input_exchange%2 -GRID]
&then
   kill %input_exchange%2 all

/******************************
/* convert shapefile coverage *
/******************************
IMPORT cover %input_exchange%.e00 %input_exchange%x
RESELECT %input_exchange%x %input_exchange%1 line
res %value_item% = %road_class%
~
n
n

kill %input_exchange%x all
RESELECT %input_exchange%1 %input_exchange% line
res fetype lk 'rd'
~
n
n
kill %input_exchange%1 all

/********************************
/* projecting the road coverage *
/********************************
&sv input_type = COVER

/*************************************************
/* give the projection to the converted Coverage *
/*************************************************
projectdefine %input_type% %input_exchange%
projection geographic
datum nad27
spheroid clarke1866
units DD
parameters

/*****************
/* projecting it *
/*****************
project %input_type% %input_exchange% %input_exchange%1

output
projection albers
datum nad83
spheroid grs80
units meters
parameters
29 30 00
45 30 00
-96 00 00
23 00 00
0
```

```
0
end

kill %input_exchange% all

/************************************************
/* converting the roads to grid and clipping it *
/************************************************
LINEGRID %input_exchange%1 %input_exchange% %value_item%
30
n
%x%, %y%
%nrows%, %ncols%
NODATA
kill %input_exchange%1 all

grid

/****************************
/* calculating costdistance *
/****************************
%input_exchange%1 = costdistance(%input_exchange%,%travel_map%,#,#,120,#)
%output_name%1 = merge (int(%input_exchange%1),%input_bound% * 0 + 120)
kill %input_exchange%1 all
kill %input_exchange% all
q
rename %output_name%1 %output_name%
gridascii %output_name% %output_name%.asc

9nogrow.aml
/***************************************************************
/* 9nogrow.aml                                 *
/*                                    *
/* This AML will produce NO GROWTH ZONE MAP GRID          *
/* by using various exchange files such as natrual areas,    *
/* nature preserves, state conservation areas, and so on     *
/* Projection will be changed too                    *
/*                                    *
/* Made by Yong Wook Kim on Jan 15, 2003                *
/* Modified by Woonsup Choi Jul 11, 2003                *
/***************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&sv x [response 'Enter the left end coordinate of the boundary grid']
&sv y [response 'Enter the bottom end coordinate of the boundary grid']
&sv nrows [response 'Enter the # of rows of the boundary grid']
&sv ncols [response 'Enter the # of columns of the boundary grid']
&sv pub_exchange [response 'Enter the PUBLIC LAND SHAPE FILE name without
extension']
/* Before running, combined all the protected area shapefiles into one file
in geographic coord.
/*&sv np_exchange [response 'Enter the NATURAL PRESERVES EXCHANGE FILE name
without extension']
/*&sv sc_exchange [response 'Enter the STATE CONSERVATION AREAS EXCHANGE
FILE name without extension']
/*&sv sf_exchange [response 'Enter the STATE FORESTS EXCHANGE FILE name
without extension']
/*&sv sfw_exchange [response 'Enter the STATE FISH EXCHANGE FILE name
without extension']
/*&sv sp_exchange [response 'Enter the STATE PARK EXCHANGE FILE name without
extension']
&sv rd_exchange [response 'Enter the ROAD EXCHANGE FILE name without
extension']
```

```
&sv value_item [response 'Enter the VALUE ITEM FIELD of the road exchange
file']
&sv road_class [response 'Enter the CLASS of the LIMITED-ACCESS HWY']
&sv input_grid [response 'Enter the name of the BOUNDARY grid']
&sv output_name [response 'Enter the FINAL OUTPUT GRID name']

&TYPE
&Type All THE EXISTING Coverage or GRID NAMED "%output_name%,”
"%pub_exchange%" "%rd_exchange%" "%rd_exchange%1" WILL BE DELETED !!!
&TYPE

&setvar delete_yesnoin [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You have to type either 'y' of 'n' for the answer.
&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

&if [exists %output_name% -COVER] or [exists %output_name% -GRID] &then
   kill %output_name% all
&if [exists %pub_exchange% -COVER] or [exists %pub_exchange% -GRID] &then
   kill %pub_exchange% all
&if [exists %pub_exchange%1 -COVER] or [exists %pub_exchange%1 -GRID] &then
   kill %pub_exchange%1 all
/*&if [exists %np_exchange% -COVER] or [exists %np_exchange% -GRID] &then
/* kill %np_exchange% all
/*&if [exists %sc_exchange% -COVER] or [exists %sc_exchange% -GRID] &then
/* kill %sc_exchange% all
/*&if [exists %sf_exchange% -COVER] or [exists %sf_exchange% -GRID] &then
/* kill %sf_exchange% all
/*&if [exists %sfw_exchange% -COVER] or [exists %sfw_exchange% -GRID] &then
/* kill %sfw_exchange% all
/*&if [exists %sp_exchange% -COVER] or [exists %sp_exchange% -GRID] &then
/* kill %sp_exchange% all
&if [exists %rd_exchange% -COVER] or [exists %rd_exchange% -GRID] &then
   kill %rd_exchange% all
&if [exists %rd_exchange%1 -COVER] or [exists %rd_exchange%1 -GRID] &then
   kill %rd_exchange%1 all
&if [exists %rd_exchange%2 -COVER] or [exists %rd_exchange%2 -GRID] &then
   kill %rd_exchange%2 all
&if [exists %input_grid%1 -COVER] or [exists %input_grid%1 -GRID] &then
   kill %input_grid%1 all

/********
/* road *
/********
/* import coverage
IMPORT cover %rd_exchange%.e00 %rd_exchange%x

/* extracting the interstate *
RESELECT %rd_exchange%x %rd_exchange% line
res %value_item% = %road_class%
~
n
n
kill %rd_exchange%x all

projectdefine cover %rd_exchange%
projection geographic
datum nad27
spheroid clarke1866
units DD
parameters
```

```
/* projecting it
project cover %rd_exchange% %rd_exchange%1
output
projection albers
datum nad83
spheroid grs80
units meters
parameters

29 30 00
45 30 00
-96 00 00
23 00 00
0
0
end

kill %rd_exchange% all

/* create 70 meter buffer to interstate *
BUFFER %rd_exchange%1 %rd_exchange% # # 70 0.002
POLYGRID %rd_exchange% %rd_exchange%2
30
n
%x%, %y%
%nrows%, %ncols%

kill %rd_exchange%
kill %rd_exchange%1
rename %rd_exchange%2 %rd_exchange%

/***************************************************************************
*
/* convert shapefile to coverage with proper projection and convert to GRID
*
/***************************************************************************
*
&sv input_type = COVER

/************************
/* Any public land area *
/************************

SHAPEARC %pub_exchange%.shp %pub_exchange% subclass
clean %pub_exchange% %pub_exchange%1
kill %pub_exchange%
rename %pub_exchange%1 %pub_exchange%

/* project define
projectdefine %input_type% %pub_exchange%
   projection geographic
   datum nad83
   spheroid grs80
   units dd
   parameters

/* projecting it
project %input_type% %pub_exchange% %pub_exchange%1
output
projection albers
datum nad83
spheroid grs80
units meters
parameters
29 30 00
```

```
45 30 00
-96 00 00
23 00 00
0
0
end

kill %pub_exchange% all
build %pub_exchange%1

/* convert to grid
POLYGRID %pub_exchange%1 %pub_exchange%
30
n
%x%, %y%
%nrows%, %ncols%
kill %pub_exchange%1 all
grid
%pub_exchange%1 = %pub_exchange% * 0 + 1
kill %pub_exchange%
q
rename %pub_exchange%1 %pub_exchange%

/**********************
/* merging final GRID *

/**********************
grid
%rd_exchange%1 = %rd_exchange% * 0 + 1
%rd_exchange%2 = %input_grid% * 0
%output_name%1 = merge(%rd_exchange%1,%pub_exchange%,%rd_exchange%2)
%output_name% = %output_name%1 * %input_grid%
q
gridascii %output_name% %output_name%.asc
/*kill %na_exchange% all
/*kill %np_exchange% all
/*kill %sc_exchange% all
/*kill %sf_exchange% all
/*kill %sfw_exchange% all
kill %rd_exchange% all
kill %rd_exchange%1 all
kill %rd_exchange%2 all
kill %pub_exchange% all
kill %output_name%1 all
```

10rampatr.aml

```
/***************************************************************
/* 10rampatr.aml                                *
/*                                    *
/* This AML will produce RAMP ATTRACTOR MAP GRID          *
/* by using roadmap exchange file and Travel Minute Map      *
/* Projection will be changed too                    *
/*                                    *
/* Made by Yong Wook Kim on Jan 15, 2003                *
/***************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&sv x [response 'Enter the left end coordinate of the boundary grid']
&sv y [response 'Enter the bottom end coordinate of the boundary grid']
&sv nrows [response 'Enter the # of rows of the boundary grid']
&sv ncols [response 'Enter the # of columns of the boundary grid']
&sv output_name [response 'Enter the FINAL OUTPUT GRID name']
```

```
&sv input_exchange [response 'Enter the ROAD EXCHANGE FILE name without
extension']

&sv value_item [response 'Enter the VALUE ITEM FIELD of the road exchange
file']

&sv road_class [response 'Enter the CLASS of the RAMP']

&sv travel_map [response 'Enter the name of the TRAVEL MINUTE MAP']

&sv input_grid [response 'Enter the name of the BOUNDARY grid']

&TYPE
&Type All THE EXISTING Coverage or GRID NAMED "%output_name%,”
"%input_exchange%,” "%input_exchange%1,” "%input_exchange%2" WILL BE DELETED
!!!
&TYPE

&setvar delete_yesnoin [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You have to type either 'y' of 'n' for the answer.
&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

&if [exists %output_name% -COVER] or [exists %output_name% -GRID] &then
   kill %output_name% all
&if [exists %input_exchange% -COVER] or [exists %input_exchange% -GRID]
&then
   kill %input_exchange% all
&if [exists %input_exchange%1 -COVER] or [exists %input_exchange%1 -GRID]
&then
   kill %input_exchange%1 all
&if [exists %input_exchange%2 -COVER] or [exists %input_exchange%2 -GRID]
&then
   kill %input_exchange%2 all
&if [exists xxx -COVER] or [exists xxx -GRID] &then
   kill xxx all

/*******************
/* import coverage *
/*******************
IMPORT cover %input_exchange%.e00 %input_exchange%x
RESELECT %input_exchange%x %input_exchange%1 line
res %value_item% = %road_class%
~
n
n
kill %input_exchange%x all

/********************************
/* projecting the ramp coverage *
/********************************
&sv input_type = COVER

/*************************************************
/* give the projection to the converted Coverage *
/*************************************************
projectdefine %input_type% %input_exchange%1
projection geographic
datum nad27
spheroid clarke1866
units DD
```

```
parameters


/*****************
/* projecting it *
/*****************
project %input_type% %input_exchange%1 %input_exchange%
output
projection albers
datum nad83
spheroid grs80
units meters
parameters
29 30 00
45 30 00
-96 00 00
23 00 00
0
0
end

kill %input_exchange%1 all

/************************************************
/* converting the roads to grid and clipping it *
/************************************************
LINEGRID %input_exchange% %input_exchange%1 %value_item%
30
n
%x%, %y%
%nrows%, %ncols%
NODATA
kill %input_exchange% all

grid
rename %input_exchange%1 %input_exchange%

/****************************
/* calculating costdistance *
/****************************
%input_exchange%1 = costdistance(%input_exchange%,%travel_map%,#,#,120,#)
%output_name% = merge (int(%input_exchange%1),%input_grid% * 0 + 120)
kill %input_exchange% all
kill %input_exchange%1 all

q
gridascii %output_name% %output_name%.asc
```

11rdintatr.aml

```
/***************************************************************
/* 11rdintatr.aml                               *
/*                                    *
/* This AML will produce RD INTERSETION MAP GRID          *
/* by using roadmap exchange file and Travel Minute Map      *
/* Projection will be changed too                    *
/*                                    *
/* Made by Yong Wook Kim on Jan 15, 2003                *
/* Modified by Woonsup Choi on July 15, 2003             *
/***************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&sv output_name [response 'Enter the FINAL OUTPUT GRID name']
```

```
/*&sv input_exchange [response 'Enter the INTERSECTION POINT EXCHANGE FILE
name without extension']
&sv input_exchange [response 'Enter the INTERSECTION POINT SHAPE FILE name
without extension']
/* A node shapefile must be created before running this AML. Do it with the
"intersection_point.apr" project file. Select only Node_type = "real"

&sv travel_map [response 'Enter the name of the TRAVEL MINUTE MAP']

&sv input_bound [response 'Enter the name of the BOUNDARY COVER']

&sv input_grid [response 'Enter the name of the BOUNDARY grid']
&sv x [response 'Enter the left end coordinate of the boundary grid']
&sv y [response 'Enter the bottom end coordinate of the boundary grid']
&sv nrows [response 'Enter the # of rows of the boundary grid']
&sv ncols [response 'Enter the # of columns of the boundary grid']

&TYPE
&Type All THE EXISTING Coverage or GRID NAMED "%output_name%,"
"%input_exchange%," "%input_exchange%," "%input_exchange%1,"
"%input_exchange%2" WILL BE DELETED !!!
&TYPE

&setvar delete_yesnoin [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You have to type either 'y' of 'n' for the answer.
&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

&if [exists %output_name% -COVER] or [exists %output_name% -GRID] &then
   kill %output_name% all
&if [exists %input_exchange% -COVER] or [exists %input_exchange% -GRID]
&then
   kill %input_exchange% all
&if [exists %input_exchange%1 -COVER] or [exists %input_exchange%1 -GRID]
&then
   kill %input_exchange%1 all
&if [exists %input_exchange%2 -COVER] or [exists %input_exchange%2 -GRID]
&then
   kill %input_exchange%2 all

/*******************
/* import coverage *
/*******************
/*IMPORT cover %input_exchange%.e00 %input_exchange%

/*******************************************
/* Converting the shapefile to a cover    *
/*******************************************
shapearc %input_exchange%.shp %input_exchange%

/******************************************
/* Projecting the cover             *
/******************************************
projectdefine cover %input_exchange%
projection geographic
datum nad27
spheroid clarke1866
units DD
parameters

project cover %input_exchange% %input_exchange%1
output
```

```
projection albers
datum nad83
spheroid grs80
units meters
parameters
29 30 00
45 30 00
-96 00 00
23 00 00
0
0
end

kill %input_exchange%
rename %input_exchange%1 %input_exchange%

/*************************************
/* Changing the point cover to grid  *
/*************************************

pointgrid %input_exchange% %input_exchange%1
30
n
%x%, %y%
%nrows%, %ncols%
NODATA

kill %input_exchange% all

GRID
/*GRIDCLIP %input_exchange%1 %input_exchange% cover %input_bound%
/*kill %input_exchange%1 all
rename %input_exchange%1 %input_exchange%

/****************************
/* calculating costdistance *
/****************************
%input_exchange%1 = costdistance(%input_exchange%,%travel_map%,#,#,120,#)
%output_name%1 = merge(int(%input_exchange%1),%input_grid% * 0 + 120)
gridclip %output_name%1 %output_name% cover %input_bound%

kill %input_exchange% all
kill %input_exchange%1 all
kill %output_name%1 all
q

gridascii %output_name% %output_name%.asc

/* You must have merged all the necessary DEM quadrangles that cover the
study area. It's in geographic coord.
/* You must have a projection file "geog_alb.txt" in the workspace.
/*****************************
/* setup the input parameter *
/*****************************
&sv clip_out [response 'Type the name of the output SLOPE GRID name']

&sv demgrid [response 'Enter a merged dem name']

&sv clip_bound [response 'Type the name of the boundary Cover']

/*&sv index = 0

/*&sv biglat = %demprefix%0

&TYPE
```

```
&Type All THE EXISTING Coverage or GRID NAMED "%clip_out%,” "%clip_out%1,”
WILL BE DELETED !!!
&TYPE

&setvar delete_yesnoin [response 'Is it okay for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You have to type either 'y' of 'n' for the answer.
&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

&if [exists %clip_out% -COVER] or [exists %clip_out% -GRID] &then
   kill %clip_out% all
&if [exists %clip_out%1 -COVER] or [exists %clip_out%1 -GRID] &then
   kill %clip_out%1 all

/******************************
/* projecting the merged grid *
/******************************
project grid %demgrid% %demgrid%1 geog_alb.txt bilinear 30
kill %demgrid%

grid
/***************************************************
/* clip the image GRID by using the boundary cover *
/***************************************************
&if [exists %clip_bound% -cover] &then
&do
   gridclip %demgrid%1 %demgrid% cover %clip_bound%
&end
kill %demgrid%1
/*******************
/* create slop_map *
/*******************
%clip_out% = slope(%demgrid%)
%clip_out%1 = int (%clip_out%)
kill %clip_out% all
rename %clip_out%1 %clip_out%
kill %clip_out%1 all
q
gridascii %clip_out% %clip_out%.asc

13stateatr.aml
/***************************************************************
/* 13stateatr.aml                               *
/*                                    *
/* This AML will produce ST RD ATTRACTOR MAP GRID          *
/* by using roadmap exchange file and Travel Minute Map      *
/* Projection will be changed too                    *
/*                                    *
/* Made by Yong Wook Kim on Jan 15, 2003                *
/* Modified by woonsup Choi on Jul 10, 2003              *
/***************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&sv x [response 'Enter the left end coordinate of the boundary grid']
&sv y [response 'Enter the bottom end coordinate of the boundary grid']
&sv nrows [response 'Enter the # of rows of the boundary grid']
&sv ncols [response 'Enter the # of columns of the boundary grid']

&sv input_exchange [response 'Enter the ROAD EXCHANGE FILE name']
```

```
/* This must be a cover converted from TIGER shapefile in geographic
coordinate

&sv value_item [response 'Enter the VALUE ITEM FIELD of the road exchange
file']

&sv road_class [response 'Enter the CLASS of the STATE HIGHWAY']

&sv travel_map [response 'Enter the name of the TRAVEL MINUTE MAP']

&sv input_bound [response 'Enter the name of the BOUNDARY grid']
&sv output_name [response 'Enter the FINAL OUTPUT GRID name']

&TYPE
&Type All THE EXISTING Coverage or GRID NAMED "%output_name%,”
"%input_exchange%,” "%input_exchange%1,” "%input_exchange%2" WILL BE DELETED
!!!
&TYPE

&setvar delete_yesnoin [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You have to type either 'y' of 'n' for the answer.
&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

&if [exists %output_name% -COVER] or [exists %output_name% -GRID] &then
   kill %output_name% all
&if [exists %input_exchange% -COVER] or [exists %input_exchange% -GRID]
&then
   kill %input_exchange% all
&if [exists %input_exchange%1 -COVER] or [exists %input_exchange%1 -GRID]
&then
   kill %input_exchange%1 all
&if [exists %input_exchange%2 -COVER] or [exists %input_exchange%2 -GRID]
&then
   kill %input_exchange%2 all

/*******************
/* import coverage *
/*******************
IMPORT cover %input_exchange%.e00 %input_exchange%x
RESELECT %input_exchange%x %input_exchange% line
res %value_item% = %road_class%
~
n
n
kill %input_exchange%x all
/*rename %input_exchange%1 %input_exchange%

/********************************
/* projecting the ramp coverage *
/********************************
&sv input_type = COVER

/*************************************************
/* give the projection to the converted Coverage *
/*************************************************
projectdefine %input_type% %input_exchange%
projection geographic
datum nad27
spheroid clarke1866
units DD
parameters
```

```
/*****************
/* projecting it *
/*****************
project %input_type% %input_exchange% %input_exchange%1
output
projection albers
datum nad83
spheroid grs80
units meters
parameters
29 30 00
45 30 00
-96 00 00
23 00 00
0
0
end

kill %input_exchange% all

/************************************************
/* converting the roads to grid and clipping it *
/************************************************
LINEGRID %input_exchange%1 %input_exchange%
30
n
%x%, %y%
%nrows%, %ncols%
NODATA
kill %input_exchange%1 all

GRID

/****************************
/* calculating costdistance *
/****************************
%input_exchange%1 = costdistance(%input_exchange%,%travel_map%,#,#,120,#)
%output_name% = merge(int(%input_exchange%1), %input_bound% * 0 + 120)

kill %input_exchange% all
kill %input_exchange%1 all
q

gridascii %output_name% %output_name%.asc

14treeatr.aml
/***************************************************************
/* 14treeatr.aml                                *
/*                                    *
/* This AML will produce TREE ATTRACTOR MAP GRID          *
/* by using LANDUSE GRID                          *
/* Projection will not be changed                    *
/*                                    *
/* Made by Yong Wook Kim on Jan 15, 2003                *
/***************************************************************

/*****************************
/* setup the input parameter *
/*****************************
&sv output_name [response 'Enter the FINAL OUTPUT GRID name']

&sv input_exchange [response 'Enter the LANDUSE GRID name']

/* &sv value_item [response 'Enter the VALUE ITEM FIELD name']
```

```
&sv input_bound [response 'Enter the name of the BOUNDARY cover']

&TYPE
&Type All THE EXISTING Coverage or GRID NAMED "%output_name%,”
"%output_name%1,” "%output_name%2" WILL BE DELETED !!!
&TYPE

&setvar delete_yesnoin [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You have to type either 'y' of 'n' for the answer.
&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

&if [exists %output_name% -COVER] or [exists %output_name% -GRID] &then
   kill %output_name% all
&if [exists %output_name%1 -COVER] or [exists %output_name%1 -GRID] &then
   kill %output_name%1 all
&if [exists %output_name%2 -COVER] or [exists %output_name%2 -GRID] &then
   kill %output_name%2 all

/*****************************************
/* select forest cells from landuse GRID *
/*****************************************
/* &sv command = '%value_item% >= 41 and %value_item% <= 43'
/* &type %value_item%
/* &type %command%

GRID
%output_name%1 = select(%input_exchange%,'value >= 41 and value <= 43')

/*********************************
/* create the euclidian distance *
/*********************************
%output_name% = eucdistance(%output_name%1)
kill %output_name%1

%output_name%1 = int(%output_name%)
kill %output_name%
rename %output_name%1 %output_name%
q

/****************************
/* clipping the output GRID *
/****************************
LATTICECLIP %output_name% %input_bound% %output_name%1
kill %output_name% all
/*kill %output_name%1 all
rename %output_name%1 %output_name%
gridascii %output_name% %output_name%.asc

15treeatr.aml
/***************************************************************
/* 15treeatr.aml                                *
/*                                    *
/* This AML will produce WATER ATTRACTOR MAP GRID          *
/* by using LANDUSE GRID                          *
/* Projection will not be changed                    *
/*                                    *
/* Made by Yong Wook Kim on Jan 15, 2003                *
/* Modified by Woonsup Choi on Jul 10, 2003              *
/***************************************************************
```

```
/*****************************
/* setup the input parameter *
/*****************************
&sv output_name [response 'Enter the FINAL OUTPUT GRID name']

&sv input_exchange [response 'Enter the LANDUSE GRID name']

/* &sv value_item [response 'Enter the VALUE ITEM FIELD name']

&sv input_bound [response 'Enter the name of the BOUNDARY cover']

&TYPE
&Type All THE EXISTING Coverage or GRID NAMED "%output_name%,”
"%output_name%1,” "%output_name%2" WILL BE DELETED !!!
&TYPE

&setvar delete_yesnoin [response 'Is it oaky for you? (y/n)' ]
&setvar delete_yesno = [locase %delete_yesnoin%]

&if %delete_yesno% ne y and %delete_yesno% ne n &then
   &return ERROR! You have to type either 'y' of 'n' for the answer.
&if %delete_yesno% = n &then
   &return Please take care of the file name and relaunch the script.

&if [exists %output_name% -COVER] or [exists %output_name% -GRID] &then
   kill %output_name% all
&if [exists %output_name%1 -COVER] or [exists %output_name%1 -GRID] &then
   kill %output_name%1 all
&if [exists %output_name%2 -COVER] or [exists %output_name%2 -GRID] &then
   kill %output_name%2 all

/****************************************
/* select water cells from landuse GRID *
/****************************************
GRID
%output_name%1 = select(%input_exchange%,'value = 11 or value = 91 or value
= 92')

/*********************************
/* create the euclidian distance *
/*********************************
%output_name% = int(eucdistance(%output_name%1))
kill %output_name%1

/****************************
/* clipping the output GRID *
/****************************
GRIDCLIP %output_name% %output_name%1 cover %input_bound%

q
kill %output_name% all
/*kill %output_name%1 all
rename %output_name%1 %output_name%
gridascii %output_name% %output_name%.asc
```

Appendix C: GRASS Makefile

```
#############################################################################
#
# Last modified: Tue Dec 16 22:05:17 CST 2003
#
# This file is used by the unix make program to create the
# LEAM landuse evolution input files using grass commands.
#
# To use,
#  Start GRASS and use a mapset that will contain the results
#  Create a directory named cell in this mapset (if not already there)
#  Copy this Makefile to that directory
#  Set the name of the needed input maps (see below)
#  Set the size of the cities (see below)
#  Set the attractor weights (see below)
#  From the cell directory, run make on this file
#############################################################################
#
#  Input Maps and Map Categories
#  Make sure the input maps use the following categories ...
#   SMALL_CITY - Location of "small cities"
#      0 No small city
#      1 Small city
#   MED_CITY - Location of "medium cities"
#      0 No small city
#      1 Small city
#   LARGE_CITY - Location of "large cities"
#      0 No small city
#      1 Small city
#   NO_GROWTH - Explicit identification of areas of no growth
#      0 No restrictions
#      1 No growth areas
#   DEM
#      Categories are meters above sea level
#   MUNICIPAL_BOUNDARY
#      City boundaries
#   STUDY_AREA
#      0 Outside study area
#      1 Inside study area
#   ROADS
#      0 No road
#      1 Limited access highway
#      2 Federal highway
#      3 State highway
#      4 County road
#      5
#      6 Ramps
#      7 Private road
#   LANDCOVER  - NLCD Landcover Map
#     11 Open water
#     12 Perennial Ice/Snow
#     21 Low Intensity Residential
#     22 High Intensity Residential
#     23 Commercial/Industrial/Transportation
#     31 Bare Rock/Sand/Clay
#     32 Quarries/Strip Mines/Gravel Pits
#     33 Transitional
#     41 Deciduous Forest
#     42 Evergreen Forest
#     43 Mixed Forest
#     51 Shrubland
#     61 Orchards/Vineyards/Other
#     71 Grasslands/Herbaceous
#     81 Pastures/Hay
#     82 Row Crops
#     83 Small Grains
#     84 Fallow
#     85 Urban/Recreational Grasses
#     91 Woody Wetlands
#     92 Emergent Herbaceous Wetlands
#
############################################################################
#
# Provide names of the input maps (name@mapset)
SMALL_CITY=small_city@LEAMBase
MED_CITY=med_city@LEAMBase
LARGE_CITY=large_city@LEAMBase
NO_GROWTH=br_nogrowth@LEAMBase
DEM=dem@kbrock
MUNICIPAL_BOUNDARY=br_mnbnd@LEAMBase
STUDY_AREA=Bragg_county@LEAMBase
LANDCOVER=landcover@LEAMBase
ROADS=road@LEAMBase
#
############################################################################
#
# Set the population average for large, medium, and small cities. These
# will be used to create the cities_attractor map, which combines these
# populations driving time maps.
LARGEPOP=95000
MEDIUMPOP=22000
SMALLPOP=3500
#
############################################################################
#
# Set the multipliers for residential growth drivers.  These are     #
multiplied times their respective 0-1 index maps.  The results are  # summed
and divided by the sum of the multipliers to give an overall  # 0-1 growth
attractor index map.
CT_RD_ATTRACTOR_RES_MULT=.7
DEM_RES_MULT=1
RAMP_ATTRACTOR_RES_MULT=0
ST_RD_ATTRACTOR_RES_MULT=2
TREE_ATTRACTOR_RES_MULT=.5
WATER_ATTRACTOR_RES_MULT=.2
CITIES_ATTRACTOR_RES_MULT=5
############################################################################
#
############################################################################
#
############################################################################
#
# IDEALLY NO EDITING IS REQUIRED BELOW HERE
############################################################################
#
############################################################################
#
############################################################################
#
```

```
LEmaps=LElandcover LEwater_att LEforest_att LEroad_att LEramp_att
LEcity_buff LEintersect_att LEstaterd_att LEslope LEno_growth LEboundary
LEcities_att

IndexMaps=ST_RD_ATTRACTOR_RES ST_RD_ATTRACTOR_OPENSPACE \
ST_RD_ATTRACTOR_COM_IND DEM_RES DEM_OPENSPACE DEM_COM_IND \
WATER_ATTRACTOR_RES WATER_ATTRACTOR_OPENSPACE WATER_ATTRACTOR_COM_IND \
TREE_ATTRACTOR_OPENSPACE TREE_ATTRACTOR_COM_IND TREE_ATTRACTOR_RES  \
RD_INTERSECTION_RES RD_INTERSECTION_OPENSPACE RD_INTERSECTION_COM_IND \
RAMP_ATTRACTOR_RES RAMP_ATTRACTOR_OPENSPACE RAMP_ATTRACTOR_COM_IND \
CT_RD_ATTRACTOR_RES CT_RD_ATTRACTOR_OPENSPACE CT_RD_ATTRACTOR_COM_IND \
CITIES_ATTRACTOR_RES CITIES_ATTRACTOR_OPENSPACE CITIES_ATTRACTOR_COM_IND \

OtherMaps=distCost hwyBuff road travelTime30 LElandcover municipalBoundary \
intersection smallCity mediumCity largeCity dem travelSpeed30 travelTime90 \
largeCityTime largeCityTime30 largeCityTime90 mediumCityTime \
mediumCityTime30 mediumCityTime90 smallCityTime smallCityTime30 \
smallCityTime90 landTravelSpeed30 roadTravelSpeed30 travelableLandSpeed \
travelSpeed90 travelTime

LEfiles=$(shell echo ${LEmaps} | sed -e 's/ /,/g')
Otherfiles=$(shell echo ${OtherMaps} | sed -e 's/ /,/g')
Indexfiles=$(shell echo ${IndexMaps} | sed -e 's/ /,/g')

all: ${LEmaps}

indexmaps: ${IndexMaps}

######################################################################
# This is the "final" step, combining residential attractors
# into an overall residential attractor map
ATTRACTOR_RES:CT_RD_ATTRACTOR_RES DEM_RES RAMP_ATTRACTOR_RES \
   ST_RD_ATTRACTOR_RES TREE_ATTRACTOR_RES WATER_ATTRACTOR_RES \
   CITIES_ATTRACTOR_RES LEno_growth LElandcover LEboundary
   r.mapcalc ATTRACTOR_RES=eval'( \
     developable=if(LEno_growth || \
     LElandcover==11 || LElandcover==21 || \
     LElandcover==22 || LElandcover==23 || LElandcover==24 || \
     LElandcover==91 || LElandcover==92 || LElandcover==85, 0, 1), \
     developable * (CT_RD_ATTRACTOR_RES * ${CT_RD_ATTRACTOR_RES_MULT} + \
     DEM_RES * ${DEM_RES_MULT} + \
     RAMP_ATTRACTOR_RES * ${RAMP_ATTRACTOR_RES_MULT} + \
     ST_RD_ATTRACTOR_RES * ${ST_RD_ATTRACTOR_RES_MULT} + \
     TREE_ATTRACTOR_RES * ${TREE_ATTRACTOR_RES_MULT} + \
     WATER_ATTRACTOR_RES * ${WATER_ATTRACTOR_RES_MULT} + \
     CITIES_ATTRACTOR_RES * ${CITIES_ATTRACTOR_RES_MULT}) / \
     (${CT_RD_ATTRACTOR_RES_MULT} + ${DEM_RES_MULT} + \
     ${RAMP_ATTRACTOR_RES_MULT} + ${ST_RD_ATTRACTOR_RES_MULT} + \
     ${TREE_ATTRACTOR_RES_MULT} + ${WATER_ATTRACTOR_RES_MULT} + \
     ${CITIES_ATTRACTOR_RES_MULT}))'

######################################################################
# The city attractor map is prepared in stages. The basic approach is
# as follows:
# Fetch the small, medium, and large city footprint maps
# For each (small, medium, and large)
#  Create road travel time map at 30 meter resolution
#  Create overland travel time map starting from times set above
#  Merge the two maps
# Combine the small, medium, and large cumulative maps
######################################################################

# Combine the travel-time maps to get to nearest small, medium, and large
# city into one map using an inverse-distance weighting algorithm using the
# population of each type as the base weight.
```

```
LEcities_att: smallCityTime mediumCityTime largeCityTime
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'LEcities_att=eval( \
     lt=if(largeCityTime==0,1,largeCityTime) / 6.0, \
     mt=if(mediumCityTime==0,1,mediumCityTime) / 6.0, \
     st=if(smallCityTime==0,1,smallCityTime) / 6.0, \
     float(${LARGEPOP})/(lt * lt) + \
     float(${MEDIUMPOP})/(mt * mt) + \
     float(${SMALLPOP})/(st * st))'
#  r.mapcalc 'LEcities_att=round(LEcities_att)'

# The cumulative cost developed at 30 meter resolution over roads
# is combined with the grosser 90 meter resolution overland times
smallCityTime: smallCityTime30 smallCityTime90
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'smallCityTime=\
     if(isnull(smallCityTime30),smallCityTime90,smallCityTime30)'

# The cumulative cost developed at 30 meter resolution over roads
# is combined with the grosser 90 meter resolution overland times
mediumCityTime: mediumCityTime30 mediumCityTime90
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'mediumCityTime=\
     if(isnull(mediumCityTime30),mediumCityTime90,mediumCityTime30)'

# The cumulative cost developed at 30 meter resolution over roads
# is combined with the grosser 90 meter resolution overland times
largeCityTime: largeCityTime30 largeCityTime90
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'largeCityTime=\
     if(isnull(largeCityTime30),largeCityTime90,largeCityTime30)'

# A cumulative cost map from small Cities at the 90 meter resolution.
# The results averages cell values across 90 meters for every 30 meter cell
smallCityTime90: smallCityTime30 travelTime90
   @echo '#########################'; echo $@ ; date
   r.mapcalc 'tmp=if(isnull(smallCityTime30),99999,smallCityTime30)'
   r.mapcalc 'tmp=min(\
     tmp[-1,-1], tmp[-1,0], tmp[-1,1], \
     tmp[ 0,-1], tmp    , tmp[ 0,1], \
     tmp[ 1,-1], tmp[ 1,0], tmp[ 1,1]) '
   r.null tmp setnull=99999
   g.region res=90
   r.cost -v -r travelTime90 out=tmp2 start_rast=tmp percent=100
   g.region res=30
   r.mapcalc 'smallCityTime90= \
     round((tmp2[-1,-1] + tmp2[-1,0] + tmp2[-1,1] + \
     tmp2[ 0,-1] + tmp2     + tmp2[ 0,1] + \
     tmp2[ 1,-1] + tmp2[ 1,0] + tmp2[ 1,1]) / 9) '
   g.remove tmp,tmp2

# A cumulative cost map from medium Cities at the 90 meter resolution.
# The results averages cell values across 90 meters for every 30 meter cell
mediumCityTime90: mediumCityTime30 travelTime90
   @echo '#########################'; echo $@ ; date
   r.mapcalc 'tmp=if(isnull(mediumCityTime30),99999,mediumCityTime30)'
   r.mapcalc 'tmp=min(\
     tmp[-1,-1], tmp[-1,0], tmp[-1,1], \
     tmp[ 0,-1], tmp    , tmp[ 0,1], \
     tmp[ 1,-1], tmp[ 1,0], tmp[ 1,1]) '
   r.null tmp setnull=99999
   g.region res=90
   r.cost -v -r travelTime90 out=tmp2 start_rast=tmp percent=100
   g.region res=30
   r.mapcalc 'mediumCityTime90= \
     round((tmp2[-1,-1] + tmp2[-1,0] + tmp2[-1,1] + \
     tmp2[ 0,-1] + tmp2     + tmp2[ 0,1] + \
     tmp2[ 1,-1] + tmp2[ 1,0] + tmp2[ 1,1]) / 9) '
   g.remove tmp,tmp2

# A cumulative cost map from large Cities at the 90 meter resolution.
# The results averages cell values across 90 meters for every 30 meter cell
largeCityTime90: largeCityTime30 travelTime90
   @echo '#########################'; echo $@ ; date
   r.mapcalc 'tmp=if(isnull(largeCityTime30),99999,largeCityTime30)'
   r.mapcalc 'tmp=min(\
     tmp[-1,-1], tmp[-1,0], tmp[-1,1], \
     tmp[ 0,-1], tmp    , tmp[ 0,1], \
     tmp[ 1,-1], tmp[ 1,0], tmp[ 1,1]) '
   r.null tmp setnull=99999
   g.region res=90
   r.cost -v -r travelTime90 out=tmp2 start_rast=tmp percent=100
   g.region res=30
   r.mapcalc 'largeCityTime90= \
     round((tmp2[-1,-1] + tmp2[-1,0] + tmp2[-1,1] + \
     tmp2[ 0,-1] + tmp2     + tmp2[ 0,1] + \
     tmp2[ 1,-1] + tmp2[ 1,0] + tmp2[ 1,1]) / 9) '
   g.remove tmp,tmp2

# A cumulative cost map at the 30 meter resolution is produced out to the
# 30-minute travel time threshold.  With a sufficiently fast r.cost, this
# step could suffice for the entire area.
smallCityTime30: smallCity travelTime30
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.cost -v travelTime30 out=smallCityTime30 start_rast=smallCity
percent=100
   r.mapcalc 'smallCityTime30=round(smallCityTime30)'

# A cumulative cost map at the 30 meter resolution is produced out to the
# 30-minute travel time threshold.  With a sufficiently fast r.cost, this
# step could suffice for the entire area.
mediumCityTime30: mediumCity travelTime30
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.cost -v travelTime30 out=mediumCityTime30 start_rast=mediumCity
percent=100
   r.mapcalc 'mediumCityTime30=round(mediumCityTime30)'

# A cumulative cost map at the 30 meter resolution is produced out to the
# 30-minute travel time threshold.  With a sufficiently fast r.cost, this
# step could suffice for the entire area.
largeCityTime30: largeCity travelTime30
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.cost -v travelTime30 out=largeCityTime30 start_rast=largeCity
percent=100
   r.mapcalc 'largeCityTime30=round(largeCityTime30)'

# Fetches a map of the locations of the small cities
smallCity:
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'smallCity=${SMALL_CITY}'
   r.null smallCity setnull=0
```

```
# Fetches a map of the locations of the medium cities
mediumCity:
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'mediumCity=${MED_CITY}'
   r.null mediumCity setnull=0

# Fetches a map of the locations of the large cities
largeCity:
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'largeCity=${LARGE_CITY}'
   r.null largeCity setnull=0

# Fetches a map of no growth areas.
# Should be replaced with fetching maps of various no growth areas that can
be
# edited and then merged.
LEno_growth:
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'LEno_growth=round(${NO_GROWTH})'

# Creates a slope map (in degrees) from the dem.  Assumes the dem z values
are meters
LEslope: dem
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.slope.aspect dem slope=LEslope
   r.mapcalc 'LEslope=round(LEslope)'

# Fetches the digital elevation model
dem:
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'dem=${DEM}'

# Create a travel cost surface for travel to ramps (road cat 6)
LEramp_att: travelTime30 travelTime90 road
   @echo '#########################'; echo $@ ; date
   # Start the travel time at 30 m resolution
   g.region res=30
   # Look for ramp (cat 6) cells that border limited access roads (cat 1)
   r.mapcalc tmp='if(road==6 && (\
      road[-1,-1]==1 || road[-1,0]==1 || road[-1,1]==1 || \
      road[ 0,-1]==1 ||            road[ 0,1]==1 || \
      road[ 1,-1]==1 || road[ 1,0]==1 || road[ 1,1]==1),1,null())'
   # Run r.cost from these locations
   r.cost -v start_rast=tmp input=travelTime30 output=tmp30 percent=100
   # Run travel time at 90 meter resolution
   g.region res=90
   r.cost -v -r start_rast=tmp30 input=travelTime90 output=tmp90
percent=100
   # Combine
   g.region res=30
   r.mapcalc 'LEramp_att=round(if(isnull(tmp30),tmp90,tmp30))'
#  g.remove tmp30,tmp90

# Create a travel cost surface for travel to state roads (road cat 6)
LEstaterd_att: travelTime30 travelTime90 road
   @echo '#########################'; echo $@ ; date
   # Start the travel time at 30 m resolution
   g.region res=30
   # Look for ramp (cat 6) cells that border limited access roads (cat 1)
   r.mapcalc tmp='if(road==6,1,null())'
   # Run r.cost from these locations
   r.cost -v start_rast=tmp input=travelTime30 output=tmp30 percent=100
   # Run travel time at 90 meter resolution
   g.region res=90
   r.cost -r -v start_rast=tmp30 input=travelTime90 output=tmp90
percent=100
   # Combine
   g.region res=30
   r.mapcalc 'LEstaterd_att=round(if(isnull(tmp30),tmp90,tmp30))'
   g.remove tmp30,tmp90

# Create a travel cost surface for travel to intersections
LEintersect_att: travelTime30 travelTime90 intersection
   @echo '#########################'; echo $@ ; date
   # Start the travel time at 30 m resolution
   g.region res=30
   # Look for ramp (cat 6) cells that border limited access roads (cat 1)
   r.mapcalc tmp='if(intersection,1,null())'
   # Run r.cost from these locations
   r.cost -v start_rast=tmp input=travelTime30 output=tmp30 percent=100
   # Run travel time at 90 meter resolution
   g.region res=90
   r.cost -r -v start_rast=tmp30 input=travelTime90 output=tmp90
percent=100
   # Combine
   g.region res=30
   r.mapcalc 'LEintersect_att=round(if(isnull(tmp30),tmp90,tmp30))'
   g.remove tmp30,tmp90

# Finds state (2) and county  (3) road  intersections.  Create a map of
# just state and county roads. Expand these locations by one cell and
# then thin them down to one cell (to take care of any small map errors).
# Then find cells that are surrounded by more than two cells containing
# roads.
intersection: road
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'tmp=if((road==2) || (road==3), 1, 0)'
   r.thin tmp output=tmp
   r.null tmp null=0
   r.mapcalc 'intersection=if(tmp==1,if(2< \
     (tmp[-1,-1] + tmp[-1,0] + tmp[-1,1] + \
     tmp[ 0,-1]         + tmp[ 0,1] + \
     tmp[ 1,-1] + tmp[ 1,0] + tmp[ 1,1]),1,0))'
   g.remove tmp

# Create the municipal buffer file for LEAM
LEcity_buff: municipalBoundary LEboundary
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc
'LEcity_buff=if(isnull(LEboundary),1,if(isnull(municipalBoundary),0,1))'

# Create a municipal boundary map based on the municipal boundary in PERM
municipalBoundary:
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.buffer in=${MUNICIPAL_BOUNDARY} out=municipalBoundary dist=1.5
unit=miles

# Fetch the study area boundary from PERM
LEboundary:
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'LEboundary=if(isnull(${STUDY_AREA}),null(),1)'
```

```
# Create a distance surface to the nearest cell containing water.
LEwater_att: LElandcover distCost
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc tmp='if(LElandcover==11,1,0)'
   r.null tmp setnull=0
   r.buffer tmp output=LEwater_att
distances=30,60,90,120,150,180,210,240,270,300,330,360,390,420,450,480,510,5
40,570,600
   r.mapcalc LEwater_att='if(isnull(LEwater_att),600,(LEwater_att-1)*30)'
   g.remove tmp

# Create a distance surface to the nearest cell containing forest -
LEforest_att: LElandcover distCost
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc tmp='if(LElandcover==41 || LElandcover==42 || \
   LElandcover==43 || LElandcover==91,1,0)'
   r.null tmp setnull=0
   r.buffer tmp output=LEforest_att
distances=100,200,300,400,500,600,700,800,900,100,1100,1200,1300,1400,1500,1
600,1700,1800,1900,2000,2500,3000
   r.mapcalc LEforest_att='if(isnull(LEforest_att),3000,(LEforest_att-
1)*100)'
   g.remove tmp

# Computes travel time to the nearest road - assuming an off-road travel
speed
# of .5 mph.  Uses the buffer program rather than the cost accumulation
program.
LEroad_att: road
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc tmp='if(road==4,1,null())'
   r.buffer tmp output=LEroad_att
distances=30,60,90,120,150,180,210,240,270,300,330,360,390,420,450,480,510,5
40,570,600,630,660,690,720,750,780,810,840,870,900,930,960,990
   # Convert distance in meters to time in minutes - assuming 5 mph
   #        60 min     mile      hr
   #  X min  =  ------- *  ----------  * ------  *  Y meter
   #          hr    1609 meter   .5 mile
   #
   # X min = 60 / (1609 * .5) min/meter * Y meter
   # X min = .0745 min/meter * Y meter
   # X min = .0745 min/meter * Y meter
   # Max is 1000 meter = 75 min
   r.mapcalc LEroad_att='if(isnull(LEroad_att), 75,round((LEroad_att-
1)*30*.0745))'
   g.remove tmp

# Creates a travel time across 90 meter cells by choosing the shortest time
# from among the 9 associated 30 meter cells.  It preserves travel time
along
# road cat 1 (limited access) and preserves the highway buffer
travelTime90: travelSpeed90
   @echo '#########################'; echo $@ ; date
   g.region res=90
   # 90m/cell * 1mile/1609meter * 60min/hr = 3.35538min.mile/cell.hr
   # 3.35538min.mile/cell.hr / x miles/hr = Y min/cell
   r.mapcalc 'travelTime90=if(travelSpeed90>0,3.35538/travelSpeed90,
null())'

# Computes the time required to traverse a cell in the N-S or E-W axes
travelTime30: travelSpeed30
```

```
   @echo '#########################'; echo $@ ; date
   g.region res=30
   # 30m/cell * 1mile/1609meter * 60min/hr = 1.11846min.mile/cell.hr
   # 1.11846min.mile/cell.hr / x miles/hr = Y min/cell
   r.mapcalc 'travelTime30=if(travelSpeed30>0,1.11846/travelSpeed30,
null())'

# Assigns travel speed to cells based on road and landcover types.
# Speeds are in miles per hour (mph)
travelSpeed30: LEno_growth LEboundary hwyBuff roadTravelSpeed30
landTravelSpeed30
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc travelSpeed30='\
   if(isnull(LEboundary), null(), \
   if(isnull(roadTravelSpeed30),null(), \
   if(roadTravelSpeed30!=0, roadTravelSpeed30, \
   if(LEno_growth, null(), \
   if(hwyBuff, null(), landTravelSpeed30)))))'

travelSpeed90: hwyBuff roadTravelSpeed30 landTravelSpeed30
travelableLandSpeed
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc
tmp='if(isnull(landTravelSpeed30),travelableLandSpeed,landTravelSpeed30)'
   r.mapcalc tmp='\
   eval( \
   highRoad = max( \
     roadTravelSpeed30[-1,-1],\
     roadTravelSpeed30[-1,0],\
     roadTravelSpeed30[-1,1],\
     roadTravelSpeed30[ 0,-1],\
     roadTravelSpeed30    ,\
     roadTravelSpeed30[ 0,1],\
     roadTravelSpeed30[ 1,-1],\
     roadTravelSpeed30[ 1,0],\
     roadTravelSpeed30[ 1,1]),\
   highLand = max( \
     tmp[-1,-1],\
     tmp[-1,0],\
     tmp[-1,1],\
     tmp[ 0,-1],\
     tmp    ,\
     tmp[ 0,1],\
     tmp[ 1,-1],\
     tmp[ 1,0],\
     tmp[ 1,1]), \
   haveBuffer = if( \
     hwyBuff[-1,-1]==1 || hwyBuff[-1,0]==1 || hwyBuff[-1,1]==1 || \
     hwyBuff[ 0,-1]==1 || hwyBuff==1     || hwyBuff[ 0,1]==1 || \
     hwyBuff[ 1,-1]==1 || hwyBuff[ 1,0]==1 || hwyBuff[ 1,1]==1, 1, 0), \
   if(highRoad > 0, highRoad, \
   if(haveBuffer > 0, null(), \
   if(highLand > 0, highLand, null()))))) '

   r.mapcalc travelSpeed90='\
   eval(\
   haveBuffer = if( \
     isnull(tmp[-1,-1]) || isnull(tmp[-1,0]) || isnull(tmp[-1,1]) || \
     isnull(tmp[ 0,-1]) || isnull(tmp)     || isnull(tmp[ 0,1]) || \
     isnull(tmp[ 1,-1]) || isnull(tmp[ 1,0]) || isnull(tmp[ 1,1]), 1, 0), \
   if(tmp==70,70,if(haveBuffer,null(),tmp)))'

roadTravelSpeed30: road
```

```
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc roadTravelSpeed30='\
     if(road==1,70, \
     if(road==2,50, \
     if(road==3,40, \
     if(road==4,30, \
     if(road==5,20, \
     if(road==6,30, \
     if(road==7,10,0)))))))'

landTravelSpeed30: LElandcover
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc landTravelSpeed30='\
     if(LElandcover==21,10, \
     if(LElandcover==22,10, \
     if(LElandcover==23,10,null())))' \

travelableLandSpeed: LElandcover
   r.mapcalc travelableLandSpeed=' \
     if(LEno_growth!=1 && \
       LElandcover!=11 && \
       LElandcover!=12 && \
       LElandcover!=91 && \
       LElandcover!=92, 1, 0)'

# Creates a cost in distance map.  Every 30 meter cell is 30.  Used by
# the water and forest attractor map processes
distCost: LEboundary
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc 'distCost=if(isnull(LEboundary), null(), 30)'

# Creates a map of 1-cell wide buffers along limited access highways
#  except where other roads intersect.  Problems:
#  - No distinction between bridges and true intersections
#  - Parallel roads in the next cell are considered access points
hwyBuff: road
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc hwyBuff='\
   if(road > 0, 0, if(road[-1,-1]==1 || road[-1,0]==1 || road[-1,1]==1 || \
     road[ 0,-1]==1 ||           road[ 0,1]==1 || \
     road[ 1,-1]==1 || road[ 1,0]==1 || road[ 1,1]==1, 1, 0))'

# Fetches the landcover file from PERM and masks it with the study area
boundary
LElandcover: LEboundary
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc LElandcover='if(isnull(LEboundary),null(), \
     if(isnull(${LANDCOVER}),0,${LANDCOVER}))'

# Fetches the road file from PERM and masks it with the study area boundary
road: LEboundary
   @echo '#########################'; echo $@ ; date
   g.region res=30
   r.mapcalc
road='if(isnull(LEboundary),null(),if(${ROADS}==0,null(),${ROADS}))'
   r.null road null=0

CITIES_ATTRACTOR_COM_IND:LEcities_att
   r.mapcalc 'CITIES_ATTRACTOR_COM_IND=graph(LEcities_att, \
   0.0, 0,580,0,1191,0.0608, 2184,0.1514, 3656,0.2830, 5826,0.4010,
9615,.6704, 15755,1.0, 68706,.9929, 118000,1.0)'

CITIES_ATTRACTOR_OPENSPACE:LEcities_att
   r.mapcalc 'CITIES_ATTRACTOR_OPENSPACE=graph(LEcities_att, \
   0,0.0, 2000,.0924, 4000,0.9322, 6000,1.0, 8000,0.8383, 10000,0.5656,
20000,0.4888, 30000,0,40000,0.1448, 118000,0.0)'

#CITIES_ATTRACTOR_RES:LEcities_att
#  r.mapcalc 'CITIES_ATTRACTOR_RES=graph(LEcities_att, \
#  0,0.0, 580,0.0, 1191,0.0464, 2184,0.0880, 3656,0.0648, 5826,0.1637,
9615,0.2693, 15755,1.0000, 68706,0.5567, 118000,0)'

CITIES_ATTRACTOR_RES:LEcities_att
   r.mapcalc 'CITIES_ATTRACTOR_RES=graph(LEcities_att, \
   0,0.0, 20000,1.0000)'

CT_RD_ATTRACTOR_RES:LEroad_att
   r.mapcalc 'CT_RD_ATTRACTOR_RES=graph(LEroad_att, \
   0,1,200,0.4044, 400,0.2687, 600,0.2123, 800,0.3106, 1400,0.0,
1600,0,1800,0,2000,0,2500,0,2677,0)'

RAMP_ATTRACTOR_RES:LEramp_att
   r.mapcalc 'RAMP_ATTRACTOR_RES=graph(LEramp_att, \
   0,0.0339, 1,0.5525, 2,0.8295, 3,1,4,0.8033, 5,0.6142, 6,0.5015, 7,0.2427,
8,0.3111, 9,0.2729, 10,0.2293, 11,0.2577, 12,0.1855, 13,0.1943, 14,0.1534,
15,0.1310, 16,0.1261, 17,0.1455, 18,0.1411, 19,0.1641, 20,0.0872, 21,0.0361,
22,0.0467, 23,0.0, 695,0)'

RD_INTERSECTION_RES:LEintersect_att
   r.mapcalc 'RD_INTERSECTION_RES=graph(LEintersect_att, \
   0,1,1,0.4499, 2,0.3401, 3,0.2182, 4,0.1560, 5,0.1632, 6,0.1400, 7,0.0969,
8,0.0797, 9,0.0, 679,0)'

TREE_ATTRACTOR_RES:LEforest_att
   r.mapcalc 'TREE_ATTRACTOR_RES=graph(LEforest_att, \
   0,1, 200,0.1145, 400,0.0158, 600,0.0030, 800,0.0, 2769,0)'

WATER_ATTRACTOR_RES:LEwater_att
   r.mapcalc 'WATER_ATTRACTOR_RES=graph(LEwater_att, \
   0,0.8807, 200,1,400,0.9947, 600,0.6515, 800,0.7220, 1000,0.5147,
1200,0.2106, 1400,0.0791, 1600,0.1166, 1800,0.0648, 2000,0.0251, 2500,0.0,
8377,0)'

DEM_RES:LEslope
   r.mapcalc 'DEM_RES=graph(LEslope, \
   0.00, 1.00, 5.00, 1.00, 10.0, 1.00, 15.0, 1.00, 20.0, 0.935, 25.0, 0.31,
30.0, 0.195, 35.0, 0.16, 40.0, 0.105, 45.0, 0.03, 50.0, 0.00, 55.0, 0.00,
60.0, 0.00, 65.0, 0.00, 70.0, 0.00, 75.0, 0.00, 80.0, 0.00, 85.0, 0.00,
90.0, 0.00)'

ST_RD_ATTRACTOR_RES:LEstaterd_att
   r.mapcalc 'ST_RD_ATTRACTOR_RES=graph(LEstaterd_att, \
   0,1,875,0.7913, 1785,0.5048, 2744,0.4177, 3765,0.3408, 4871,0.3229,
6099,0.5550, 7512,0.2844, 8000,0.0, 13060,0)'

CT_RD_ATTRACTOR_COM_IND:LEroad_att
   r.mapcalc 'CT_RD_ATTRACTOR_COM_IND=graph(LEroad_att, \
   0,1,200,0.5967, 400,0.4657, 600,0.3524, 800,0.3173, 1000,0.2330,
1200,0.0, 1400,0.0, 1600,0,1800,0,2000,0,2500,0,2677,0)'

CT_RD_ATTRACTOR_OPENSPACE:LEroad_att
   r.mapcalc 'CT_RD_ATTRACTOR_OPENSPACE=graph(LEroad_att, \
   0,1,200,0.4044, 400,0.2687, 600,0.2123, 800,0.3106, 1000,0.5916,
1200,0.6238, 1400,0.1263, 1600,0,1800,0,2000,0,2500,0,2677,0)'
```

```
RAMP_ATTRACTOR_COM_IND:LEramp_att
   r.mapcalc 'RAMP_ATTRACTOR_COM_IND=graph(LEramp_att, \
   0,0.9254, 1,1,2,0.7202, 3,0.4261, 4,0.3061, 5,0.1874, 6,0.1017, 7,0.0839,
8,0.0714, 9,0.0666, 10,0.0781, 11,0.0627, 12,0.0277, 13,0.0200, 14,0.0207,
15,0.0293, 16,0.0205, 17,0.0148, 18,0.0116, 19,0.0140, 20,0.0158, 21,0.0134,
22,0.0047, 23,0.00, 695,0)'

RAMP_ATTRACTOR_OPENSPACE:LEramp_att
   r.mapcalc 'RAMP_ATTRACTOR_OPENSPACE=graph(LEramp_att, \
   0.00, 0.33, 10.0, 0.49, 20.0, 0.45, 30.0, 0.43, 40.0, 0.4, 50.0, 0.39,
60.0, 0.38, 70.0, 0.35, 80.0, 0.325, 90.0, 0.295, 100,0.14)'

RD_INTERSECTION_COM_IND:LEintersect_att
   r.mapcalc 'RD_INTERSECTION_COM_IND=graph(LEintersect_att, \
   0,1,1,0.4236, 2,0.2289, 3,0.1423, 4,0.0838, 5,0.0537, 6,0.0409, 7,0.0403,
8,0.0259, 9,0.0, 679,0)'

RD_INTERSECTION_OPENSPACE:LEintersect_att
   r.mapcalc 'RD_INTERSECTION_OPENSPACE=graph(LEintersect_att, \
   0,1,1,0.7619, 2,0.5447, 3,0.3471, 4,0.2191, 5,0.1479, 6,0.0723, 7,0.0368,
8,0.0274, 9,0.0281, 10,0.0354, 11,0.0233, 12,0.0177, 679,0)'

TREE_ATTRACTOR_COM_IND:LEforest_att
   r.mapcalc 'TREE_ATTRACTOR_COM_IND=graph(LEforest_att, \
   0,1,200,0.1357, 400,0.0058, 600,0.0011, 800,0.0, 2769,0)'

TREE_ATTRACTOR_OPENSPACE:LEforest_att
   r.mapcalc 'TREE_ATTRACTOR_OPENSPACE=graph(LEforest_att, \
   0.00, 0.405, 10.0, 0.385, 20.0, 0.375, 30.0, 0.365, 40.0, 0.35, 50.0,
0.335, 60.0, 0.305, 70.0, 0.285, 80.0, 0.27, 90.0, 0.255, 100,0.06)'

WATER_ATTRACTOR_COM_IND:LEwater_att
   r.mapcalc 'WATER_ATTRACTOR_COM_IND=graph(LEwater_att, \
   0,1,200,0.5305, 400,0.3258, 600,0.2480, 800,0.1734, 1000,0.0954,
1200,0.0706, 1400,0.0328, 1600,0.0, 8377,0)'

WATER_ATTRACTOR_OPENSPACE:LEwater_att
   r.mapcalc 'WATER_ATTRACTOR_OPENSPACE=graph(LEwater_att, \
   0.00, 0.455, 10.0, 0.44, 20.0, 0.435, 30.0, 0.43, 40.0, 0.415, 50.0,
0.405, 60.0, 0.37, 70.0, 0.345, 80.0, 0.33, 90.0, 0.315, 100,0.235)'

DEM_COM_IND:LEslope
   r.mapcalc 'DEM_COM_IND=graph(LEslope, \
   0.00, 1.00, 5.00, 1.00, 10.0, 1.00, 15.0, 1.00, 20.0, 0.935, 25.0, 0.31,
30.0, 0.195, 35.0, 0.16, 40.0, 0.105, 45.0, 0.03, 50.0, 0.00, 55.0, 0.00,
60.0, 0.00, 65.0, 0.00, 70.0, 0.00, 75.0, 0.00, 80.0, 0.00, 85.0, 0.00,
90.0, 0.00)'

DEM_OPENSPACE:LEslope
   r.mapcalc 'DEM_OPENSPACE=graph(LEslope, \
   0.00, 1.00, 5.00, 1.00, 10.0, 1.00, 15.0, 1.00, 20.0, 0.935, 25.0, 0.31,
30.0, 0.195, 35.0, 0.16, 40.0, 0.105, 45.0, 0.03, 50.0, 0.00, 55.0, 0.00,
60.0, 0.00, 65.0, 0.00, 70.0, 0.00, 75.0, 0.00, 80.0, 0.00, 85.0, 0.00,
90.0, 0.00)'

ST_RD_ATTRACTOR_COM_IND:LEstaterd_att
   r.mapcalc 'ST_RD_ATTRACTOR_COM_IND=graph(LEstaterd_att, \
   0,1,875,0.2825, 1785,0.1750, 2744,0.1578, 3765,0.0, 13060,0)'

ST_RD_ATTRACTOR_OPENSPACE:LEstaterd_att
   r.mapcalc 'ST_RD_ATTRACTOR_OPENSPACE=graph(LEstaterd_att, \
   0,1,875,0.4310, 1785,0.4017, 2744,0.1961, 3765,0.1695, 4871,0.1052,
6099,0.1548, 7512,0.1381, 9356,0.0100, 13060,0)'
```

```
# Evolve Landscape
futureLanduse: ATTRACTOR_RES
   r.mapcalc
'futureLanduse=if((ATTRACTOR_RES*ATTRACTOR_RES*ATTRACTOR_RES*ATTRACTOR_RES)>
rand(0.0,1.0),if(rand(0.0,1.0)>.9,21,null()),null())'
#  r.mapcalc futureLanduse=if'(ATTRACTOR_RES * ATTRACTOR_RES * ATTRACTOR_RES
/ 4>\
#  rand(0.0,1.0),21,null())'

check:
   d.what.rast
ATTRACTOR_RES,CT_RD_ATTRACTOR_RES,DEM_RES,RAMP_ATTRACTOR_RES,ST_RD_ATTRACTOR
_RES,TREE_ATTRACTOR_RES,WATER_ATTRACTOR_RES,CITIES_ATTRACTOR_RES,LEno_growth
,LElandcover,LEboundary

altfutureLanduse: ATTRACTOR_RES
#  x=$(shell r.stats -c ATTRACTOR_RES | sed -e 's/-[^ ]*/*/' -e 's/$$/+\\/'
-e 's/^\*.*/0/' | bc ); \
#  echo $$x ; \
   r.mapcalc futureLanduse=if'(ATTRACTOR_RES * ATTRACTOR_RES >\
   rand(0.0,1.0),21,null())'

# Removes all index files
cleanIndex:
   g.remove rast=${Indexfiles}

# Removes all the temporary files used in the creation of the LEAM inputs
clean:
   g.remove rast=${Otherfiles}

# Removes all the temporary and LEAM input files
veryclean: clean
   g.remove rast=${LEfiles}
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)
11-2004

2. REPORT TYPE
Final

3. DATES COVERED (From - To)

4. TITLE AND SUBTITLE
Development of Input Maps for the military Land use Evolution and Assessment Model (mLEAM) Land Use Change (LUC) Simulation Model

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
James D. Westervelt, Kyle Brock, Woonsup Choi, and Yong Wook Kim

5d. PROJECT NUMBER
SERDP

5e. TASK NUMBER

5f. WORK UNIT NUMBER
CS-1257

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Engineer Research and Development Center (ERDC)
Construction Engineering Research Laboratory (CERL)
PO Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER
ERDC/CERL TR-04-24

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Strategic Environmental Research and Development Program (SERDP)
901 North Stuart St.
Suite 303
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
SERDP

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

14. ABSTRACT

This document is intended to be used by GIS technicians in the development of input maps for the military Land use Evolution and Assessment Model (mLEAM) Land Use Change (LUC) simulation model. Military installations are facing growing challenges with adjacent incompatible land use patterns that limit the installation's ability to train soldiers and test equipment. Training and testing area associated with noise, dust, smokes, and radio transmissions that carry beyond an installations fenceline. Urban growth in these areas results in increasing potential for complaints from homeowners. The patterns of that urban growth are in direct response to past regional plans that involve specific investments and policies. Current plans and policies must be evaluated with respect to their direct, indirect, and cumulative impacts. The military version of the Land use Evolution and Assessment Model procedures described in this document provide an improved opportunity to evaluate proposed regional plans.

15. SUBJECT TERMS
simulation modeling
mLEAM
urban growth
encroachment
military training
land management
land use planning
geographic information

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** James D. Westervelt
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	112	**19b. TELEPHONE NUMBER (include area code)** (217) 373-4530