The Design of the Land-Use Change Analysis System (LUCAS): Part I – Graphical User Interface

R.L. MacIntyre, B.C. Hazen & M.W. Berry

Computer Science Department

CS-94-266 December 1994

Abstract

This is first of two technical reports documenting the Land-Use Change Analysis System (or LUCAS) developed in the Department of Computer Science at the University of Tennessee. This report (Part I) presents the Motif-based user interface for utilizing LUCAS. The internal design and implementation on a distributed network of workstations will constitute the second report (Part II) scheduled for release in Spring 1995.

The LUCAS Graphical User Interface (GUI) is a Windows-based user interface that allows access to the LUCAS software for users of different levels of experience. Early user of LUCAS are expected to be experts in landscape management. Explicit knowledge of the integration of socioeconomic and ecological variables is not required by users such as ecologists, economists, sociologists, or forestry personnel. To accomodate the needs of such experts, a graphical user interface (GUI) was developed for LUCAS. A GUI is the best way to provide an intuitive interface for an application which integrates interdisciplinary data of different scales and type. One of the most challenging aspects of designing an intuitive interface for LUCAS was to first understand the needs of landscape management, and then to develop the interface with these requirements in mind. The LUCAS GUI can be compiled and executed on any workstation that supports the X-Window System Release 5 (X11R5) and the Open Software Foundation's (OSF) Motif 1.2.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Overview . . . 2

2 Design Considerations **7**
- 2.1 Enabling User Specifications . . . 7
- 2.2 Widget Selection . . . 9
- 2.3 Displaying Graphs . . . 10

3 Implementation **12**
- 3.1 Setup Menu . . . 12
  - 3.1.1 Socioeconomic Model Selection . . . 12
  - 3.1.2 Watershed Selection . . . 13
  - 3.1.3 Saving Generated Maps . . . 13
  - 3.1.4 Scenario Selection . . . 13
  - 3.1.5 Scenario Repetition . . . 14
  - 3.1.6 Landscape Change Methods . . . 14
  - 3.1.7 Initial Map Year . . . 15
  - 3.1.8 Species Impacts Modules . . . 15
  - 3.1.9 Options . . . 16
  - 3.1.10 Help . . . 16
- 3.2 LUCAS Analysis Status Window . . . 17
- 3.3 LUCAS Graphs Window . . . 17

4 Sample LUCAS Session **18**
- 4.1 Customizing LUCAS . . . 18
- 4.2 Initializing a Simulation . . . 18
- 4.3 Displaying Graphs . . . 22

5 Current Platforms and Future Work **28**
- 5.1 Current Platforms . . . 28

5.2 Future Work 28

A User Defaults **31**

B X Windows Resources **33**

C Little Tennessee River Basin Scenario Description **36**

List of Figures

1.1 LUCAS modules of the Olympic Peninsula/Southeastern Appalachian Biosphere Integration Model. 6

2.1 LUCAS Setup Menu with default values. 8
2.2 LUCAS Graphs Menu. 11

4.1 LUCAS Setup Menu with values for simulation. 21
4.2 LUCAS Analysis Progress Status window. 22
4.3 GRASS monitor of completed simulation. 23
4.4 LUCAS Graphs window. 24
4.5 Proportion of Unvegetated Landscape. 25
4.6 Proportion of Forest Landscape. 26
4.7 Area of Unvegetated Landscape. 27

Chapter 1

Introduction

In this report, a graphical user interface for the Land-Use Change and Analysis System (LUCAS) is presented. LUCAS is a UNIX-based software tool that enables the study of the effects of land use on landscape structure [B+92]. The motivation for the LUCAS Graphical User Interface or GUI and pertinent background information (see also [Mac94]) is presented in this chapter. Chapter 2 describes the design considerations of the LUCAS GUI. In Chapter 3, the implementation of the LUCAS GUI is discussed followed by a sample LUCAS session in Chapter 4. The current computing environment for LUCAS simulations and a discussion of future LUCAS GUI development is presented in Chapter 5. Default interface selections are provided in Appendix A along with information on customizable interface resources in Appendix B. A list of land-use scenarios for the simulation of land-cover change in the Little Tennessee River basin is provided in Appendix C.

1.1 Motivation

The LUCAS graphical user interface provides an intuitive, user-friendly interface to the LUCAS application. One of the initial design goals of the LUCAS GUI was to develop an interface which could be easily used and understood by users with a wide range of technical abilities and windows experience. In order to provide a *comfortable* interface which facilitates utilization of the underlying LUCAS software, the LUCAS GUI allows the user to customize the interface. To provide this extensive degree of customization, the LUCAS GUI incorporates user-defined resources. The end-user can indicate a preference for interface component colors and window sizes and locations. A set of default resource values is provided for novice LUCAS users.

The underlying LUCAS software consists of a socioeconomic model module, a landscape-change model module, and an impact models module. The Geographical Resources Analysis Support System (GRASS) developed by the U.S. Army Construc-

tion Engineering Research Laboratories [S+94] is utilized as a common database which allows these three modules to communicate and to share map data files. Since these LUCAS components were developed on UNIX-based workstations, the graphical user interface was primarily designed for UNIX environments.

The LUCAS graphical user interface was developed using the X Windows System Release 5 (X11R5) as the underlying windowing software. X Windows is the *de facto* industry standard for windowing systems and is available on a wide variety of computing platforms. A number of windowing toolkit libraries written to facilitate the development of X Windows-compliant software are available. These toolkits provide a suite of pre-defined graphical user interface components. The Motif Version 1.2 libraries are an X Windows-based toolkit developed by the Open Software Foundation. Because Motif can be obtained for most computing platforms, including both UNIX and non-UNIX computers, the Motif libraries were selected for development of the LUCAS GUI. Another advantage of using the X Windows windowing system is that the LUCAS GUI can be displayed on any end-user computing platform which supports the X Windows system. This includes a range of display devices from high-end graphics workstations to personal computers running X Windows emulators.

One of the primary motivations to develop a user interface for LUCAS was to provide an integrated environment for the components of the LUCAS project. Before the interface was created, it was necessary for the user to run each of the components *manually*, feeding the output of one module to the input to the next module. This required sophisticated knowledge of the interworkings of the modules from the researchers. The graphical user interface allows the user to run the models and simulations without an extensive knowledge of how all of the pieces are integrated. Thus, the researcher has an environment in which he can concentrate on his field of study.

Comments from potential users of LUCAS at the Oak Ridge National Laboratory on the design of the user interface were critical in order to develop an interface which would be useful to landscape ecologists and land-use managers. In order to gather important feedback from such users while they were available, the Hewlett-Packard UIM/X user interface design system was used to create a prototype interface in a timely manner. This prototype was used to facilitate the design of the user interface but was not used in the final software application.

1.2 Overview

Ecological dynamics in human-influenced landscapes are strongly affected by socioeconomic factors that influence land-use decision making. Incorporating these factors into a spatially-explicit landscape-change model requires integrating multidisciplinary data. LUCAS facilitates the study of the effects of land use on landscape structure in regions such as the Little Tennessee River basin in western North Carolina and

the Olympic Peninsula in the state of Washington. The map layers used by LUCAS are derived from remotely-sensed images, census and ownership maps, topographical maps, and outputs from economic models. These map layers are stored, displayed, and analyzed using GRASS [S+94] as a geographical information system. Simulations using LUCAS generate new maps of land cover representing the amount of land-cover change so that issues such as biodiversity conservation, assessing the importance of landscape elements to meet conservation goals, and long-term landscape integrity can be addressed.

Landscape pattern is a product of the interaction between ecological and socioeconomic processes. Understanding the function and structure of landscapes, primarily in terms of human impacts, requires integration of biological and socioeconomic knowledge. Natural resource managers in particular need this integration to effectively evaluate the social and environmental consequences of alternative management scenarios. The Man and the Biosphere (MAB) project focuses on land use and its impacts as compared between the Olympic Peninsula and the Southeastern Appalachian Biosphere reserves [LFT+92].

This project integrates knowledge spanning many disciplines in order to evaluate land use and its impacts. Integration requires not only interpretation across disciplines, but also compatibility in the forms of data acquired. Such forms include spatial and tabular databases, results of mathematical models, spatial analyses, and expert opinions. Unfortunately, conventional approaches of integrating and applying knowledge are not adequate to examine the complex and highly-variable ecological and socioeconomic issues that influence human land-use decision making and the impacts these decisions have on landscapes [CFL87].

Technologies are now available to facilitate the development of a multidisciplinary model for studying sustainability. Geographic information systems such as GRASS can easily be used to represent and manipulate spatial data on workstations. In addition, adaptive management approaches provide a conceptual framework for evaluating alternative scenarios [Hol78]. The Land-Use Change Analysis System (LUCAS) is a prototype computer application specifically designed to integrate ecological and socioeconomic information using GRASS for adaptive approaches to landscape management.

The model discussed in [LFT+92] examines the impact of human activities on environmental and natural resource sustainability. The premise of the model is that landscape properties such as fragmentation, connectivity, spatial dynamics, and the degree of dominance of habitat types are influenced by market processes, human institutions, landowner knowledge, and ecological processes. Therefore, modeling environmental sustainability of human-dominated landscapes will benefit from the integration of human and ecological processes.

The structure adopted for LUCAS consists of three subject modules linked by a common database (see Figure 1.1). The first LUCAS module contains the socioeco-

nomic models that are used to derive transition probabilities associated with changes in land cover. These probabilities are computed as a function of socioeconomic driving variables including:

1. transportation networks (access and transportation costs),
2. slope and elevation (indicators of land-use potential),
3. ownership (land holder characteristics),
4. land cover (vegetation), and
5. population density.

The result of studying the effects of these driving variables on the landscape is stored in a Transition Probability Matrix (TPM). The TPM is a data structure which contains the information used to calculate the probability that any data point pixel with a given landscape condition will change to another landscape condition based on the pre-determined scenario. The TPM contains the independent variable coefficients which are used in an equation along with the landscape condition value for the pixel to calculate the probability of change. The independent variable coefficients were empirically derived by studying landscape change as it relates to social and economic factors, the driving variables mentioned in the above list. The current TPM used by LUCAS was derived by Wear et al. in [WTF94]. The TPM is not calculated during a LUCAS simulation, but is used as input to the landscape-change module of LUCAS. Specifically, the TPM is used by the stochastic process used to predict the resulting landscape based on the initial state of the landscape and the land-use scenario selected by the user. The TPM is stored in the scenario disk file on the workstation that LUCAS has been installed on. Each supported scenario includes a TPM derived specifically for that scenario. A sample scenario will be illustrated in Chapter 4. When the LUCAS GUI invokes the underlying analysis software to run a simulation, the TPM is read from the scenario file on disk into memory and is kept in memory.

The landscape-change model resides in the second LUCAS module (see Figure 1.1). This module receives as its input the TPM produced in the socioeconomic model (Module 1) and accesses the same spatial database of driving variables. A single iteration of the landscape-change model produces a map of land cover that reflects socioeconomic motivations behind human land-use decision making (represented in the transition probability matrix). The user chooses a number of timesteps and a number of replicates from the GUI. The GUI passes this information to the underlying analysis code through the use of UNIX environment variables. The analysis code generates one map for each timestep in each simulation. Thus, if the user chooses 4 timesteps and 10 replicates, 40 maps will be generated during the simulation. Each timestep map is based on the previous timestep map. These maps are stored on disk

in a temporary area while they are needed to generate the next map, and then are deleted from disk. The final timestep maps (one for each replicate chosen by the user) are stored in a temporary disk area. The last n maps for each replicate can be saved by the user in his map area by choosing n on the *Maps Saved* scrollbar.

The impact models defined in the third module of LUCAS (see Figure 1.1) use the land-cover maps produced by the landscape-change module to estimate impacts to selected environmental and resource-supply variables. These environmental variables include the amount and spatial arrangement of habitat for selected species and changes in water quality caused by human land use. Potential resource-supply variables include timber yields and real estate values. The resultant simulation maps stored in the temporary disk area are used as input to the impact models. The user chooses the impact models that he/she is interested in from the GUI. This information is passed to the underlying analysis code through the use of UNIX environment variables. Graphs of the species module results and of the landcover analysis are available after the simulation has completed. The impact module and landcover analysis results are stored in a file which is updated throughout the simulation. This file is used as input to Xgraph (a public-domain X Windows graphing package) [Har89] if the user chooses to view the results graphically. The results are also stored in a SAS-format [SAS92] file which can be used by the researcher to make SAS plots.

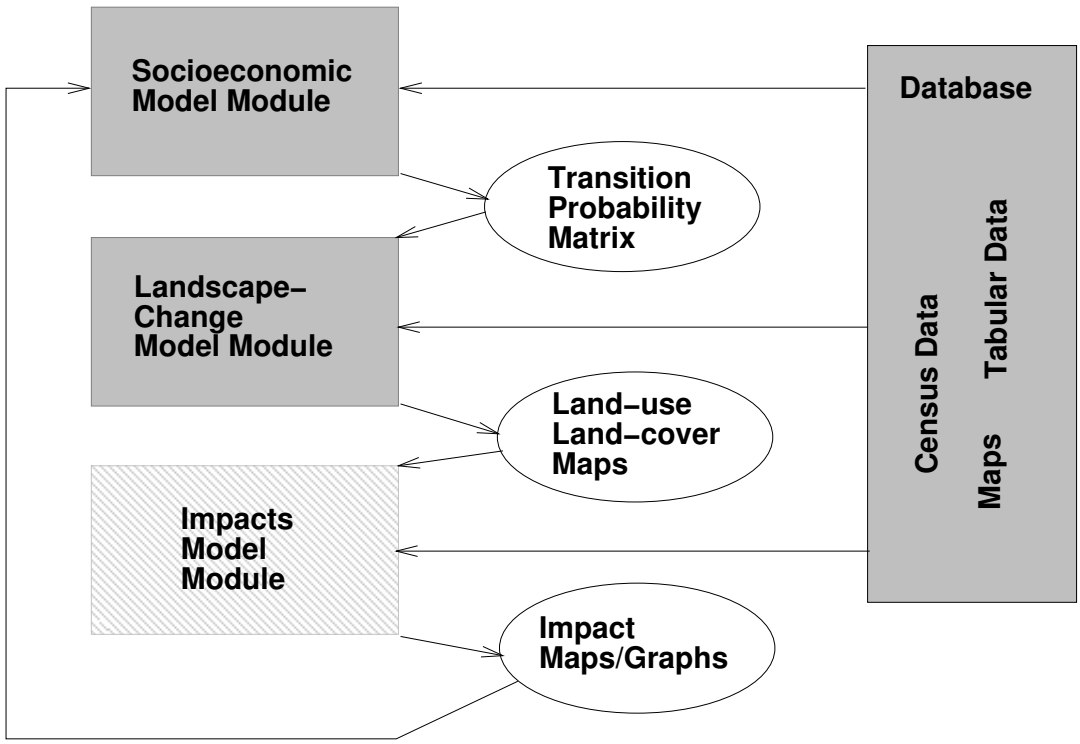

Figure 1.1: LUCAS modules of the Olympic Peninsula/Southeastern Appalachian Biosphere Integration Model.

Chapter 2

Design Considerations

Several fundamental design principles were incorporated into the design of the LUCAS GUI in order to produce an interface which facilitates the utilization of the LUCAS software by members of diverse user communities. Potential LUCAS users will be scientific researchers who will use LUCAS as an environment in which to test the validity of their socioeconomic, land-use change, and impacts models. Each of these researchers may have a great deal of knowledge in their particular area of expertise but not in all of the modules in the integrated LUCAS environment. Other potential LUCAS users will be land-use management decision policy makers. Such users can use LUCAS to simulate the effects of policy decisions on the landscape. Typically, this class of LUCAS users will not have an in-depth knowledge of any of the underlying LUCAS modules.

2.1 Enabling User Specifications

One of the primary considerations in designing the LUCAS GUI was to develop an interface which will enable the research scientist or natural resource manager to run simulations without requiring the user to know a great deal about module initialization or the flow of data between the modules. There are several pieces of required information which must be collected from the user before a simulation can be executed. This information is passed from the user interface to the underlying LUCAS modules. A default value for each of these selections is provided on the user interface. A list of the defaults is located in Appendix A. Although default values are predefined for each LUCAS module, the user has the option of running a simulation with the default values or overriding the default selections.

To prevent from overwhelming the user with options, the user interface was designed to collect all of the initial information to run a simulation from a single window — the *LUCAS Setup Menu* (see Figure 2.1). The options which will most likely be

modified by the user are included in the main window area of the *LUCAS Setup Menu*. Options which will be accessed less often are located on a pulldown menu. By locating these particular choices elsewhere from the main window area, the user is less likely to be confused by too many choices. Yet at the same time, the user can easily locate these selections under the *Options* button at the top of the *LUCAS Setup Menu*. The location of options which are the least likely to be used are consistently located on a pulldown menu at the top of each LUCAS user interface window.

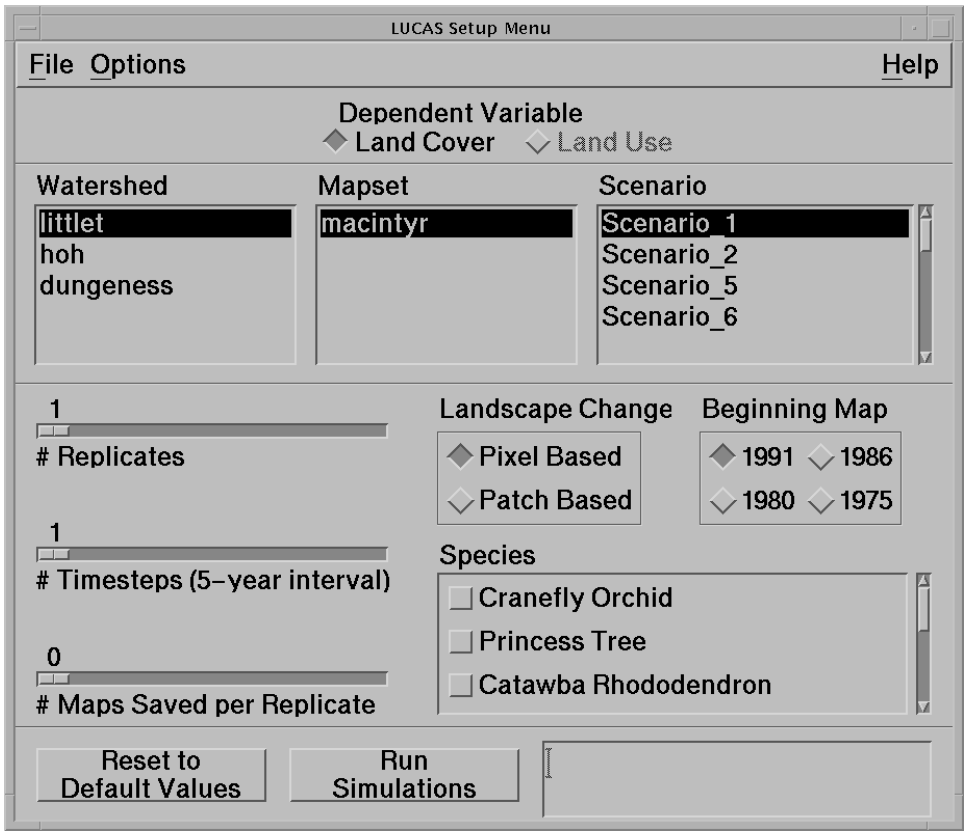

Figure 2.1: LUCAS Setup Menu with default values.

2.2 Widget Selection

A *widget* is a picture of a virtual input device (e.g., menu, slider, gauge, text box) and an event-handler for managing events directed at this device (e.g., mouse downs, mouse ups, keyboard presses). The LUCAS user interface contains several different types of widgets selected from the OSF/Motif toolkit library to collect user input. Specific inputs will be discussed in Chapter 3. In order to prevent the user from making an invalid selection, the widgets used in the LUCAS interface contain all of the possible choices and the user simply chooses one. No widgets used in the LUCAS GUI require user keyboard input.

One widget type used on the *Setup Menu* is the slider (or scale) widget (see Figure 2.1). This type of widget is used to receive the user's choice of number of replicates, number of timesteps, and number of maps saved per replicate. The advantage of using a slider as opposed to a text input field for this type of input data is that the user cannot enter an incorrect value for these fields. The minimum and maximum values on the slider widget indicate and enforce the acceptable limits of values for the input variables. Input widgets which contain a selection list have a scrollbar located to the right of the widget if the list of possible choices is too long to display in the widget. This scrollbar is only visible when there are more choices than fit on the list widget.

There are several input fields on the LUCAS user interface for which only one choice from a list of possible choices is acceptable. This is called a *one-of-many* relationship and is indicated on the user interface by a selection box containing diamond-shaped buttons. For these input variables, the user interface enforces the condition that the user can choose one and only one option for the variable. Only one option from the list of choices can be selected for the following input variables:

1. Dependent Variable,
2. Landscape Change,
3. Beginning Map,
4. Cover Type, and
5. Graph.

A selection box widget containing buttons which allow a number (including zero) of selections from the possible choices is used to gather user input for the species impact model on the *LUCAS Setup Menu*. This *many-of-many* relationship is indicated on the user interface by a selection box containing square-shaped buttons. The selection box widget will allow the user to choose any number of species from the list of species impacts models.

There are two buttons on the *Setup Menu* which cause an action to be taken by the LUCAS interface code. These buttons are located together at the bottom of the *LUCAS Setup Menu* and are separated by a horizontal separator to help indicate to the user that these are *action buttons* and are not components of the interface which are waiting for input. The LUCAS user interface follows the design principle of locating the buttons which initiate an action at the bottom of the window so that users familiar with other windowing packages will recognize this familiar *look-and-feel*.

A status window is also located at the bottom of the *Setup Menu*. The status window is a visual indicator to the user that the underlying LUCAS code is performing some action while the user interface remains unchanged. The status window also indicates the percentage of the processing which has been completed.

2.3 Displaying Graphs

Once a simulation has completed, the user can view the generated data in a graph displayed on the screen. Each graph contains a statistic for a given cover type based on the output map generated. The supported statistics include:

1. Proportion of Landscape,
2. Area of Covertype,
3. Edge/Area Ratio,
4. Edge of Covertype Pairs,
5. Number of Patches,
6. Mean Patch Size,
7. Standard Deviation of Patch Size,
8. Largest Patch,
9. Mean Patch Shape, and
10. Distribution of PatchSize.

Any of the above listed statistics can be displayed for each of the cover types found on the map. There is one graph available for each statistic for each cover type, making the number of possible graphs quite large. Because of the large number of possible graphs, the graph selection window was separated into two possible selection boxes — one for the cover type and one for the statistic (see Figure 2.2). The user selects the cover type that he/she is interested in and then selects the statistic to be graphed. The graph of the statistic for the chosen cover type is displayed using version 11.3.2 of `xgraph`, a public-domain graphing tool for X Windows systems [Har89].

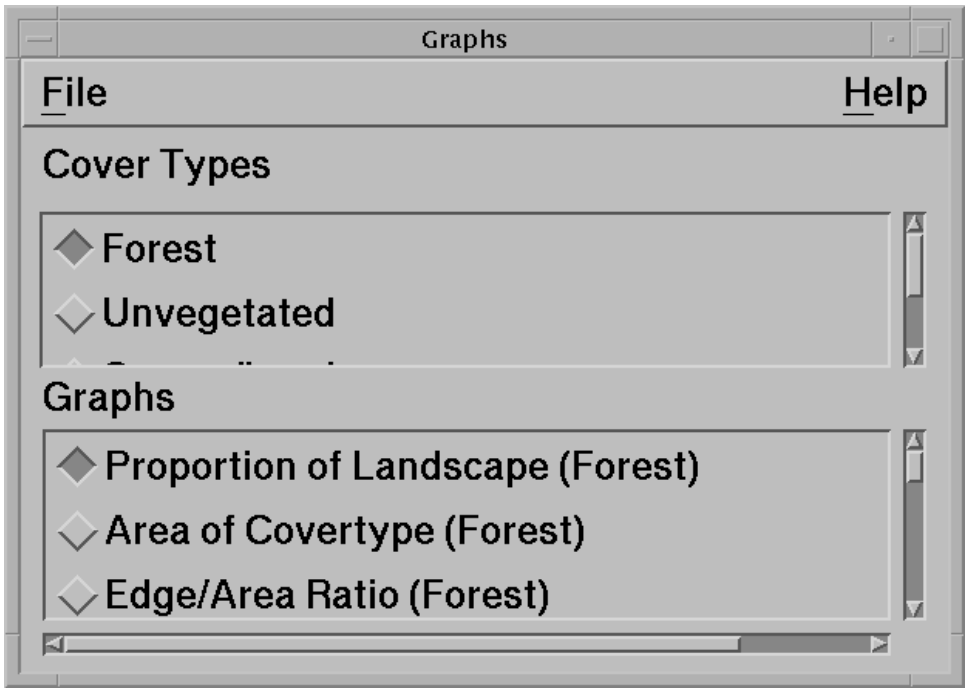

Figure 2.2: LUCAS Graphs Menu.

By receiving input from the user with the use of two selection boxes, the user is not required to enter a selection from the keyboard. In this way, the user cannot make a wrong selection when indicating which graph should be plotted on the screen. Another advantage is that the user does not have to consult a user manual to determine the set of available choices.

Chapter 3

Implementation

The LUCAS GUI is implemented using X11R5 and the Motif 1.2 toolkit library. The following sections describe each individual interface implementation and the specific implementation of the X11R5 and Motif 1.2 toolkits.

3.1 Setup Menu

LUCAS enables the user to run simulations based on a pre-defined set of environmental conditions called *scenarios*. The simulations demonstrate how the landscape conditions will change based on the information specified in the appropriate scenario file, including the Transition Probability Matrix (see Section 1.2). There are a number of variables which must be set by the user before the simulation can begin. These variables are input by the user on the LUCAS *Setup Menu*. A description of the implementation of each variable selection on the *Setup Menu* is presented in the following subsections.

3.1.1 Socioeconomic Model Selection

The socioeconomic model currently supported by LUCAS is the land-cover model developed by Wear et al. ([WTF94]). In this model, the dependent variable is land cover. The LUCAS Setup Menu displays two choices for the dependent variable: land cover and land use. The land-use dependent variable is greyed-out, indicating that it is not currently supported. The land-use dependent variable choice is inactive since the underlying land-use change socioeconomic model has yet to be integrated into the LUCAS environment. To indicate that the land-use option is not currently supported by LUCAS, the `XmNsensitive` resource of the Motif `XmToggleButton` widget is set to `FALSE` so that this option appears grey on the user interface.

3.1.2 Watershed Selection

For each landscape area on which land cover change is simulated, a set of gridded maps must exist. The integration of socioeconomic and ecological variables is accomplished spatially through the use of these gridded maps. Individual maps may represent a single data theme which describes physical landscape attributes (e.g., land cover, slope, soil type), spatial features (e.g., distance relationships, adjacency rules), the results of socioeconomic and ecological processes (e.g., changes in real estate values, species abundance, and erosion), and land-ownership characteristics (e.g., tract size, shape, and history). Pixels in each map are assigned to one of the discrete categories used to describe that data theme [B+92].

Currently, a set of gridded maps exist for three study areas: the Little Tennessee River Basin, the Hoh Watershed, and the Dungeness Watershed. The LUCAS user must choose one of the available watersheds prior to running the simulation. To display the watershed selection, the LUCAS GUI utilizes the Motif `XmScrolledList` widget. The `XmScrolledList` widget allows the user to select the watershed on which they would like to simulate land cover change. Selection occurs when the user *points-and-clicks* on an item in the list. The `XmNsingleSelectionCallback` resource specifies the callback routine which determines which watershed was selected. The list of watersheds is not *hard-coded*, but is instead generated at run-time by parsing the names of the available scenario files. The format of the scenario filenames is `location-watershed-depvar-scenario--title-year`, where `location` indicates the watershed that the scenario has been prepared for and `depvar` represents the dependent variable of the simulation. The scenario files are located in a directory indicated by the `LUCAS_DIR` environment variable. The LUCAS GUI parses each filename in the `LUCAS_DIR` directory and creates the scrolled list of watersheds.

3.1.3 Saving Generated Maps

Running a simulation of land-cover change produces an output map for each timestep and replicate. If the user chooses to save any of these intermediate maps, they will be saved in the directory which is specified from the *Mapset* `XmScrolledList` widget on the LUCAS Setup Menu. The *Mapset* list contains only directories which the user has ownership (write permission). The number of maps saved per replicate is input by the user. The user has the option of saving between 0 to nt maps, where nt is the maximum number of timesteps. If the user specifies to save n maps, where n is less than nt, the last n maps generated will be saved in his mapset directory.

3.1.4 Scenario Selection

There are a number of pre-defined environmental conditions called *scenarios* for each watershed. Associated with each scenario is the TPM used to predict land cover

change. These scenario files are located in the directory indicated by the `LUCAS_DIR` environment variable. As is done for the list of available watersheds, the LUCAS GUI parses the name of each file in the `LUCAS_DIR` directory to build a list of scenario names. The scenario names are then displayed to the user in a Motif `XmScrolledList` widget. The user chooses a scenario other than the default by *pointing-and-clicking* on the scenario name. When the user chooses a scenario, a description of the scenario is displayed in a *Scenario Description* interface. It is important to show the description of the scenario to the user since the description is normally too long to capture in the scenario title.

Each of the `XmScrolledList` widgets used in the LUCAS interface has its `XmNscrollBarDisplayPolicy` resource set to `XmAS_NEEDED`, which indicates that scrollbars will be displayed only if the workspace exceeds the clip area in one or both dimensions. This allows the interface to place multiple text strings into a small interface window.

3.1.5 Scenario Repetition

The LUCAS user may repeat the scenario simulation a number of times. The LUCAS user interface gathers this value from the user as the number of replicates. Within each replicate, the landcover map will be generated at 5-year intervals. The user can specify how many 5-year intervals (or timesteps) he wants to simulate landscape change per replicate. The user also has the option of saving some of the maps generated at each replicate.

The input mechanism for the number of replicates, the number of timesteps, and the number of maps saved per replicate is an `XmScrollBar` Motif widget. The user selects the value for the input variable by sliding the slider on the `XmScrollBar` widget with the mouse button pressed. The LUCAS `XmNminimum` and `XmNmaximum` resources of the `XmScrollBar` widget indicate the range of acceptable values for the input variable. One advantage of using the `XmScrollBar` widget is that this widget guarantees that the value selected for the variable is within the range of acceptable values. The `XmNmaximum` resource of the number of maps saved per replicate is dependent on the number of timesteps that the user has chosen. This `XmNmaximum` resource is updated by the user interface software each time the user changes the value for number of timesteps.

3.1.6 Landscape Change Methods

Landscape change can be calculated with either a *pixel-based* or *patch-based* method. A *pixel-based* landscape change method calculates the probability of change on a pixel-by-pixel basis. A *patch* is a collection of contiguous pixels with identical landscape condition labels. In a *patch-based* landscape change method, the entire patch

is identified and a transition is predicted for the entire patch.

The socioeconomic model developed by Wear et al. ([WTF94]) assumes a *pixel-based* landscape change model. Research scientists may investigate the difference that calculating landscape change with a *patch-based* model will have on the resulting landscape. Thus, the user has the option of choosing either a *pixel-based* (default) or *patch-based* landscape change model. A Motif `XmRadioBox` widget is used for the landscape change method selection. One button is created for each of the possible landscape change method options. The `XmNbuttons` resource of the `XmRadioBox` is set to the array of landscape change buttons. The `XmRadioBox` widget enforces a *one-and-only-one* selection for landscape change. The *one-and-only-one* selection policy for the landscape change `XmRadioBox` widget is indicated to the user by the use of diamond-shaped `XmToggleButtons` for the landscape change selections. Using different button shapes is the standard Motif mechanism to indicate selection policy to the user.

3.1.7 Initial Map Year

Four sets of gridded map data have been created for each of the watersheds supported by LUCAS. These map sets were created from remote satellite data collected in 1975, 1980, 1986, and 1991. The user can begin simulations at any one of these initial map years. A `XmRadioBox` widget is used for the selection of the beginning map year using the *one-and-only-one* selection policy.

Each scenario is specific for a particular combination of dependent variable, watershed, and beginning map year, therefore, when the value of any one of these three variables changes the scenario `XmScrolledList` widget must be updated. Therefore, the callback routine `begmapChangedProc` associated with the `XmNsimpleCallback` resource of the beginning map `XmRadioBox` widget updates the list of scenario names based on the newly selected value for beginning map year. A similar callback routine is associated with the `XmNsimpleCallback` resource of the dependent variable `XmRadioBox` widget and the `XmNsingleSelectionCallback` resource of the watershed `XmScrolledList` widget.

3.1.8 Species Impacts Modules

LUCAS currently incorporates an impacts module which calculates the impact on organism biodiversity based on the land-cover maps generated by the simulation. The user must indicate the organisms on which the land cover change impact should be calculated before the simulations begin. The underlying LUCAS software generates a binary map of suitable habitat for each of the species selected, and collects statistics at each timestep which are saved to a file. Although not currently implemented, the LUCAS GUI in the future will be able to plot graphs of the species–impact

maps for the user. The species that the user has chosen are passed to the underlying LUCAS software by using an environment variable. The selection policy of the species `XmScrolledList` is *many-of-many*. This is indicated to the user by square-shaped buttons for the species impact `XmToggleButtons`.

3.1.9 Options

In addition to the `XmText` status widget at the bottom of the *Setup Menu*, a *Status Window* indicates the current land-cover map which is being generated. The *Status Window* conveys more information to the user than the `XmText` status widget, which only provides the percentage of computation which has been completed. Displaying the *Status Window* is optional and is controlled by a `XmRadioBox` widget located on the *Status Window Configuration* window which is *realized* when the user selects the *Status Window Configuration* from the *Options* button located at the top of the *Setup Menu*. The location of the *Options* button is in accordance with the OSF/Motif Style Guide.

A finite number of GRASS [S+94] display monitors are available on each workstation running GRASS. Each workstation user must choose a different GRASS monitor to display map data. The available GRASS monitors are displayed on the *GRASS Monitor Configuration* window which is *realized* when the *GRASS Monitor Configuration* button is selected from the *Options* menu. By default, the first GRASS monitor available will be used to display the landcover maps. The user can select a different monitor from the GRASS Monitor `XmRadioBox` located on the *GRASS Monitor Configuration* window. Users who do not want to see the landcover maps displayed in a GRASS monitor may turn off the GRASS monitor by selecting the *OFF* button in the *GRASS Monitor Configuration* `XmRadioBox`.

3.1.10 Help

Although the LUCAS interface was designed to be used with a minimal amount of documentation, there are some LUCAS interactions which may require additional explanation. The LUCAS interface supports an on-line help facility. Each *Help* submenu provides an explanation of the particular interface's displayed information and the interaction of the interface's components. The interfaces included in the help subsystem include:

1. *Setup Menu*,
2. *Status Window configuration*,
3. *GRASS Monitor Configuration*, and
4. *Graphs*.

In accordance with the OSF/Motif Style Guide, the *Help* menu pane is located in the rightmost location on the menu bar by setting the `XmNmenuHelpWidget` resource. Online help is implemented by displaying an ASCII text help file when the user chooses the Help button. There is a separate help file for each *Help* choice in the LUCAS menus.

3.2 LUCAS Analysis Status Window

While the underlying LUCAS software is generating landcover maps, an *Analysis Status* window is displayed which indicates the status of the simulation. The *Analysis Status* window consists of an `XmText` field for each of the following variables: `Watershed`, `Scenario`, `Current Replicate`, `Base Map Year`, and `Current Map Year`. The `XmNeditable` resource of each `XmText` field is set to `FALSE` so that the user cannot modify the values displayed in the field. The status window was added to the LUCAS interface to display the current value of the variables during a simulation which was taped for a video.

The watershed `XmText` field is set to the full name of the watershed which is stored in a user-configurable environment variable named by the watershed abbreviation. For instance, the environment variable named `littlet` contains the full name *Little Tennessee River Basin*. Similarly, the scenario `XmText` field is set to the full name of the scenario which is stored in a user-configurable environment variable named by the scenario name abbreviation. For example, the environment variable named `scenario_1` contains the full name for the first scenario. The current replicate `XmText` field is initially set to 1 and is incremented each time a replicate is completed. The user chooses the number of replicates to complete in the simulation from the *LUCAS Setup Menu*. The base map year `XmText` field is set to the value that the user selected for *Beginning Map* on the LUCAS Setup Menu. The current map year `XmText` field is initially set to the value that the user selected for *Beginning Map* on the LUCAS *Setup Menu* and is reset to the year of the map currently being generated by the LUCAS analysis code. The current map year increment is 5 years.

3.3 LUCAS Graphs Window

After the simulations are complete, the user can view graphs of the resulting data using `xgraph`, a public-domain graphing package for X Windows systems [Har89]. The user selects which graph to view from the Graphs Window which contains two `XmScrolledList` regions labeled *Cover Types* and *Graphs*. The *Cover Types* scrolled list contains a list of the landscape cover types found in the landcover map. The *Graphs* scrolled list contains a list of each statistic that was calculated for the cover types.

Chapter 4

Sample LUCAS Session

In this chapter, the features and functionality of LUCAS are demonstrated. The session begins with the specification of the variables prior to a simulation. After the land-cover change simulation has completed, graphs of the output data are viewed. Throughout the session, the advanced enhancements of LUCAS are demonstrated.

4.1 Customizing LUCAS

The customization of the LUCAS interface is consistent with other X Windows applications. In particular, a resource file (`LUCAS.rf`) is used to notify the X Windows window manager of default values for particular resources (e.g., font). For simplicity, the notification of these values to the window manager is provided through the local `XENVIRONMENT` environment variable. Before invoking the LUCAS interface, the user executes `XENVIRONMENT=LUCAS.rf` at the system prompt to notify the window manager of the user-defined resources (see Appendix B for this example resource file).

4.2 Initializing a Simulation

Before beginning the land-cover change simulation, the user can specify input parameters to the simulation. The user selects the input parameters from the LUCAS *Setup Menu* (recall Figure 2.1).

The user first chooses the watershed on which the land-cover change simulation will be run. In this example, the Little Tennessee River Basin is selected. To select this particular watershed, the user places the pointer (by moving the mouse) over the watershed name in the *Watershed* scrolled list and double clicks. The text field is highlighted upon selection. The LUCAS interface retrieves the watershed selection and updates the *Scenario* scrolled list based on the watershed choice, as the scenarios are developed for each watershed. The `LOCATION_NAME` environment variable is then

set to the appropriate value based on the watershed selected by the user.

Next, the user can select the mapset directory in which output maps are saved. The *Mapset* selection list contains only directories into which the user can write files because of ownership rules. In our example, the only possible selection for mapset is *macintyr* because that is the only directory in which user *macintyr* has ownership. The `MAPSET` environment variable used by the underlying analysis software is set to the highlighted value on the *Mapset* scrolled list.

The *Scenario* scrolled list contains the list of pre-defined scenarios which are available for the selected watershed. There are sixteen pre-defined scenarios for the Little Tennessee River Basin watershed. A description of these scenarios is provided in Appendix C. Because all sixteen scenarios can not be displayed in the *Scenario* scrolled list at once, a scrollbar has been placed to the right of the *Scenario* scrolled list by the `XmScrolledList` widget. The user can change the scenario titles displayed in the list by moving the mouse to place the pointer over the scrollbar, pressing the left mouse button to *grab* the location indicator on the scrollbar, and moving the mouse to modify the scenario names displayed. In this example, *scenario 1* is selected from the list of scenarios. Scenario 1 is the land-cover change based on the historical transition probabilities for the period 1986–1991 applied to both public and private land [B+92]. For this example simulation, we will be constructing a future landscape on which the landcover will change based on patterns recorded between the years 1986–1991. The resulting landscape will appear as it could if the same land-use decisions are made as were for the years 1986-1991. The environment variables `LUCAS_SCENARIO` and `LUCAS_STATS` are set according to the scenario selected from the *Scenario* scrolled list to communicate this information with the underlying analysis software.

Several simulations are run in order to gather statistically valid results [WTF94] from the simulation. In this example, the simulation will be run ten times, so ten replicates are chosen from the *Number of Replicates* scale widget. The user can select the number of replicates by moving the pointer with the mouse to the indicator on the *Number of Replicates* scale widget, pressing the left mouse button to *grab* the indicator on the scrollbar, and moving the mouse to modify the number of replicates. The value selected by the user for replicates is communicated to the analysis software by the `LUCAS_REPLICATES` environment variable.

To predict how the landscape will look in 100 years, 20 timesteps are selected from the *Number of Timesteps* scale widget. The user chooses the number of timesteps in the same manner in which the number of replicates are selected. The value chosen for the number of timesteps is communicated to the analysis software by the `LUCAS_TIMESTEPS` environment variable.

No maps will be saved in this example, so the default of 0 maps saved is not altered on the *Number of Maps Saved per Replicate* scrollbar. The number of maps to be saved is communicated to the analysis software by the `LUCAS_MAPS_SAVED` environment variable.

In this example, landscape change will be based on a pixel-based model. Since pixel-based is the default method of simulating landscape change, the value for landscape change is not altered on the *Landscape Change* selection box. The method of landscape change is communicated to the analysis software by the `LUCAS_PIXEL_PATCH` environment variable.

For this example, the simulation will begin in the year 1991 and will simulate landscape change 100 years into the future. Since the default value in the *Beginning Map* selection box is 1991, the value for beginning map is not altered. The selection for the beginning map year is indicated to the underlying analysis code in the title of the selected scenario.

No species are selected for this simulation. If the user wants to see the effect of the resulting landscape with respect to species biodiversity, the user would choose values from the *Species* scrolled list. The selected species are communicated to the analysis code by the `LUCAS_SPECIES` environment variable.

After selecting values for all of the setup information, the user presses the *Run Simulations* push button located at the bottom of the setup window, invoking the underlying analysis code (see Figure 4.1). The *LUCAS Analysis Status* window (see Figure 4.2) is displayed indicating the current status of the analysis in progress.

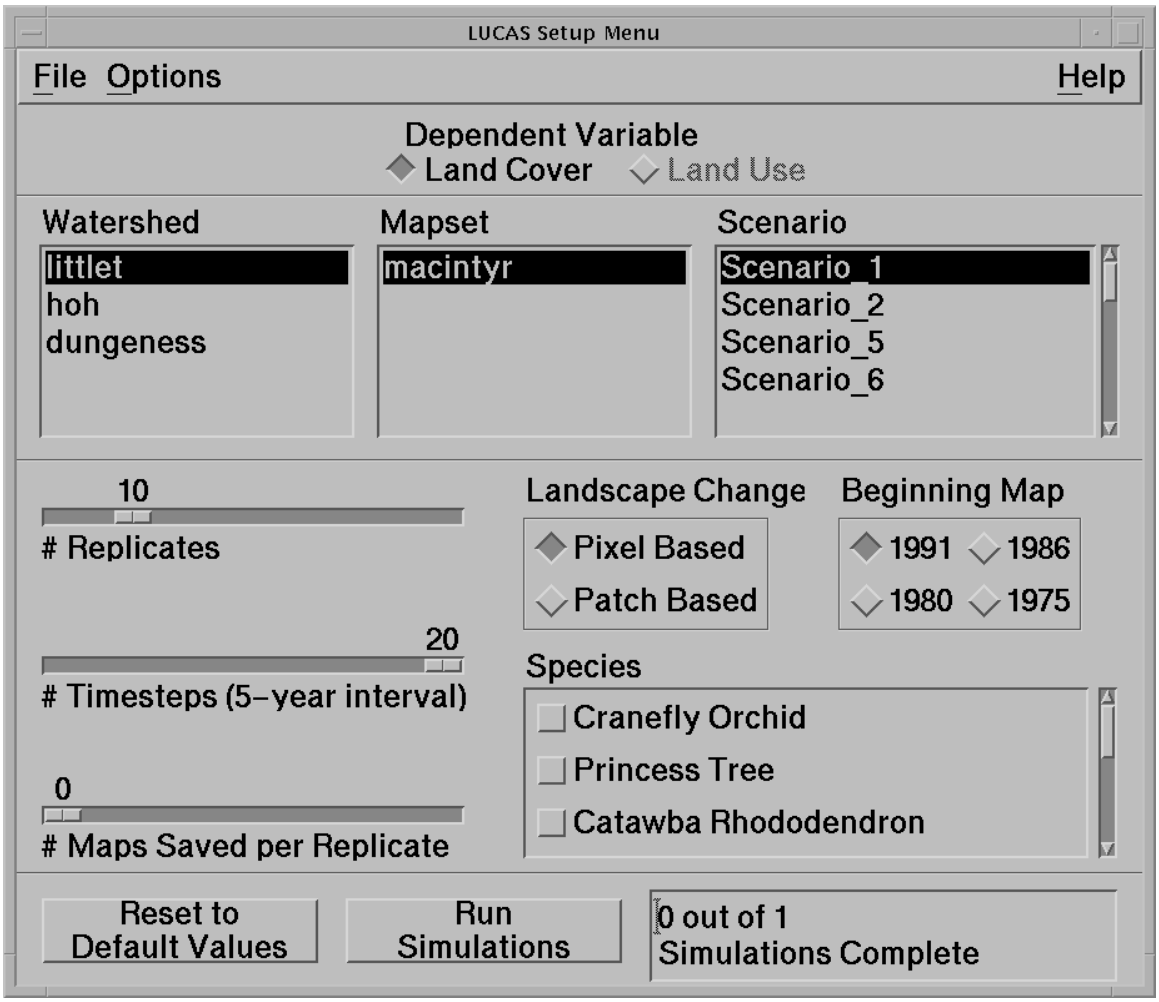

Figure 4.1: LUCAS Setup Menu with values for simulation.

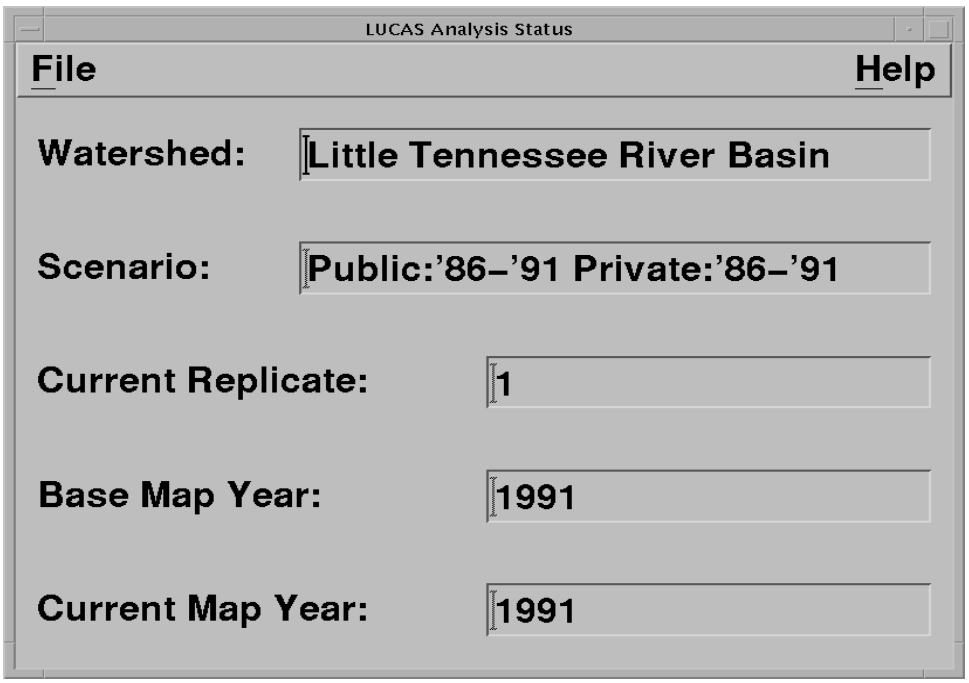

Figure 4.2: LUCAS Analysis Progress Status window.

4.3 Displaying Graphs

This example simulation demonstrates the effects of land-use decisions from the years 1986–1991 on the western North Carolina (Macon County) landscape for 100 years beyond 1991. After the simulation has completed, the GRASS monitor containing both the original land-cover and the final transition land-cover maps is displayed (see Figure 4.3). It is difficult to tell for certain from the final land-cover map how the amount of each landcover type has changed during the simulation. To analyze the result of the simulation, graphs of the statistics are displayed.

A *Graphs* window (see Figure 4.4) is displayed after the simulation has completed in order to allow the user to observe trends in desired statistics. Several graphs are displayed in this example, including:

1. Proportion of unvegetated landscape (see Figure 4.5)
2. Proportion of forest landscape (see Figure 4.6)
3. Area of unvegetated covertype (see Figure 4.7)

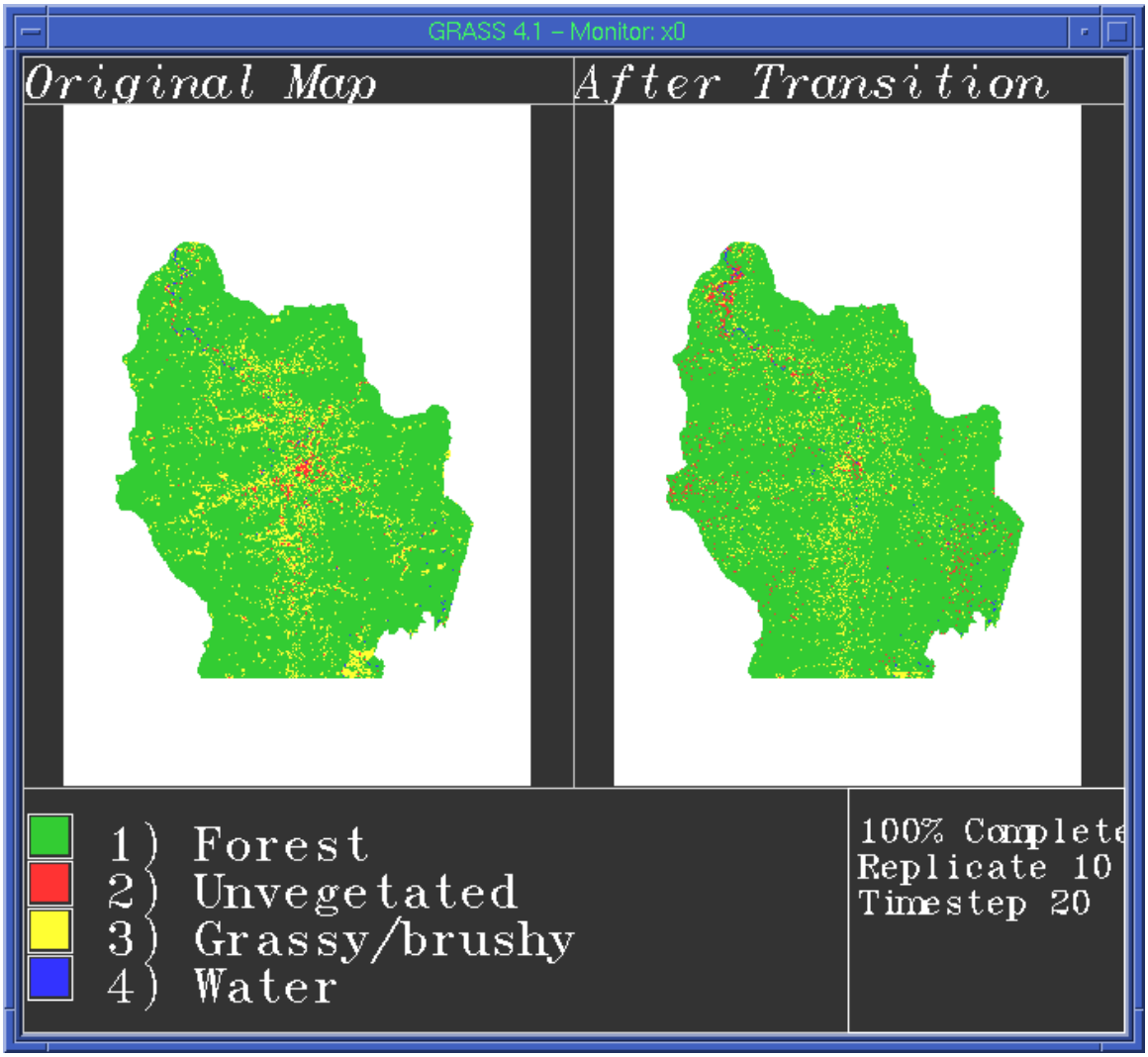

Figure 4.3: GRASS monitor of completed simulation.

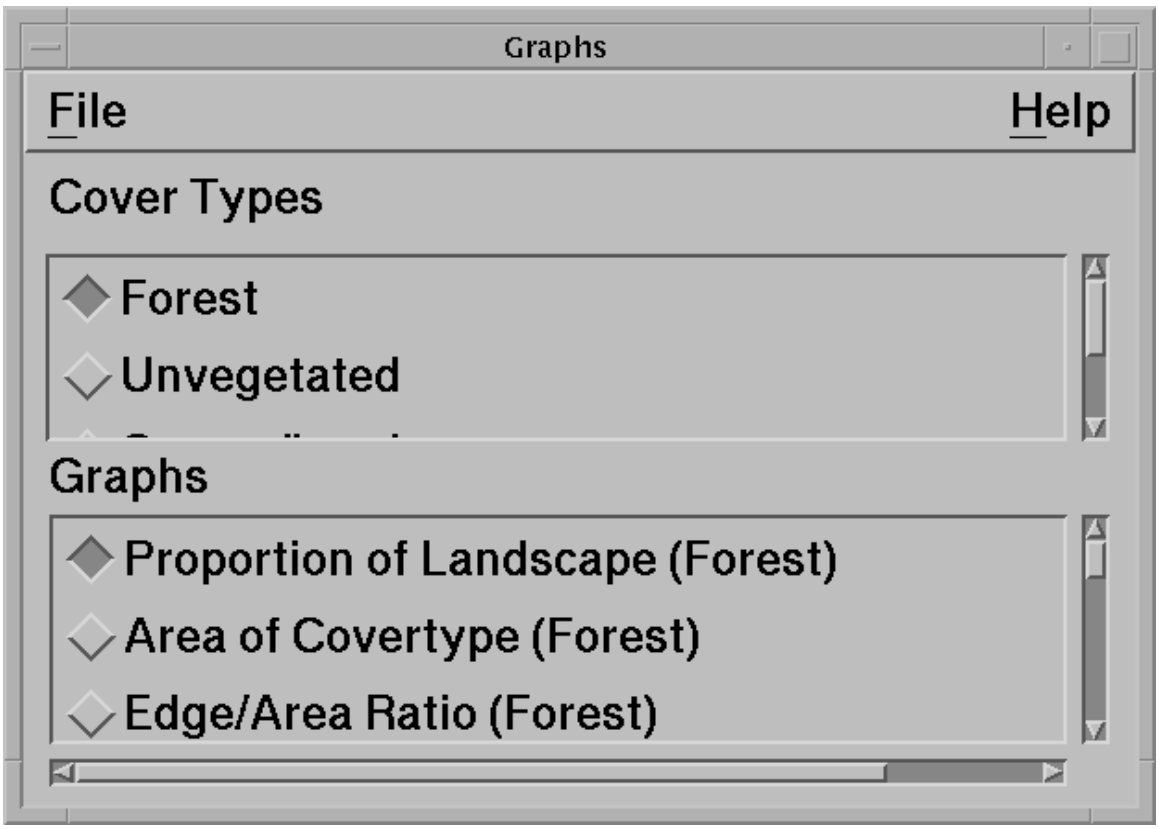

Figure 4.4: LUCAS Graphs window.

From the *proportion of landscape* graphs (see Figures 4.5 and 4.6), it can be observed that the proportion of unvegetated land cover increases from less than 1 percent to nearly 2 percent of the land being unvegetated in the year 2091. The proportion of the landscape covered by forest decreases by 1 percent in the 100 year simulation. The *area of covertype* graph (see Figure 4.7) indicates that approximately 1000 hectares of landscape transitions to unvegetated landcover during the simulation.

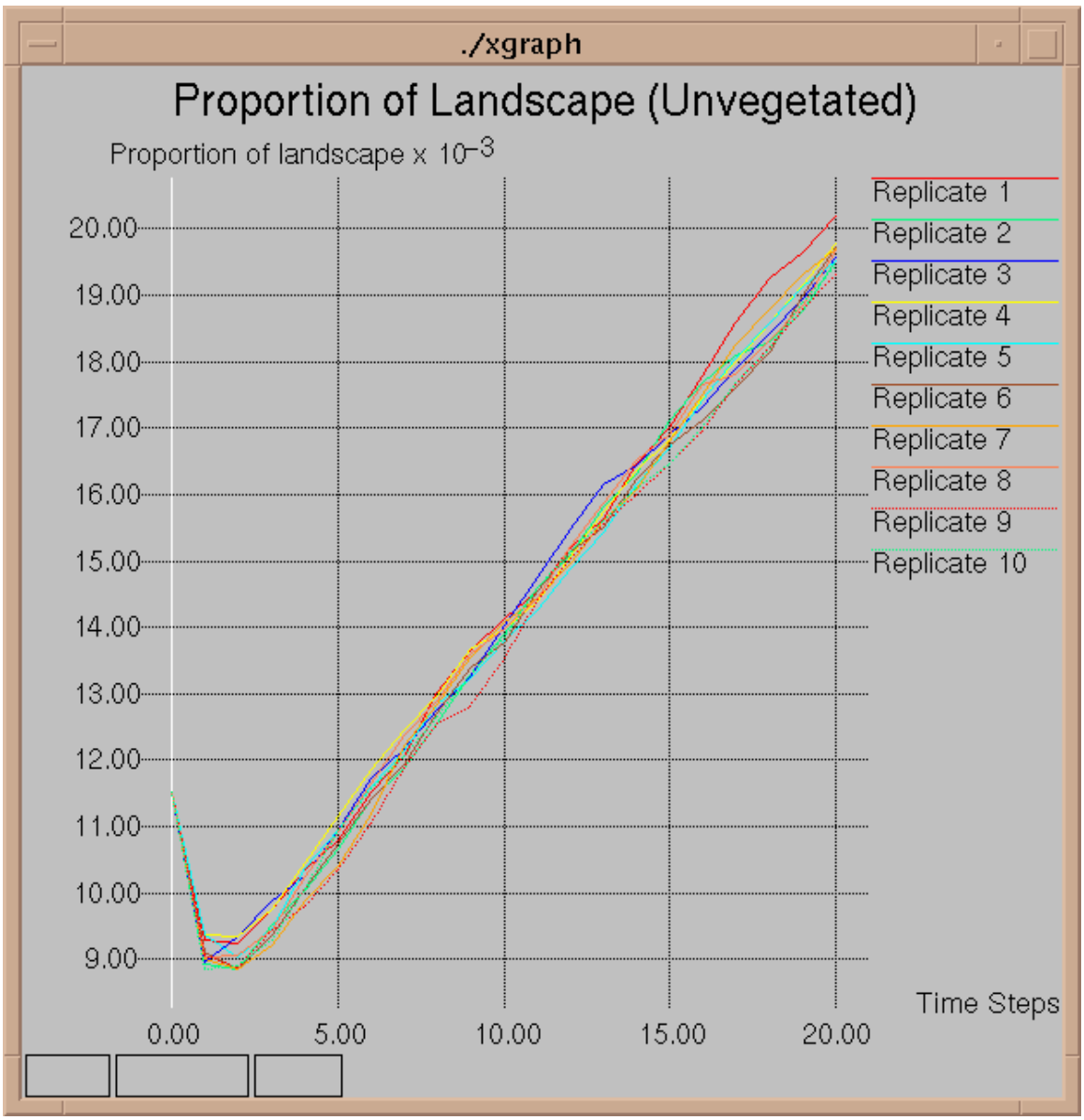

Figure 4.5: Proportion of Unvegetated Landscape.

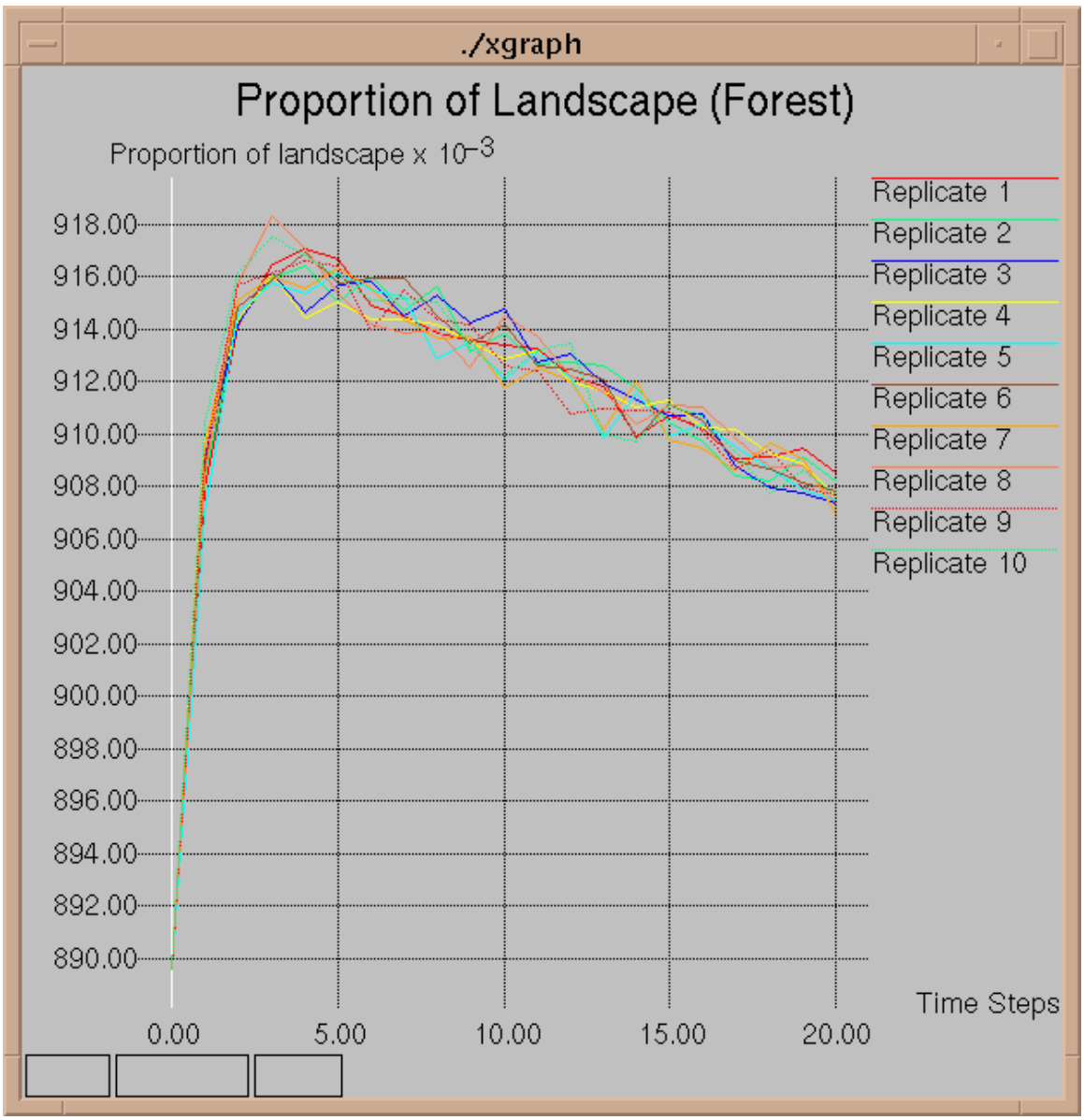

Figure 4.6: Proportion of Forest Landscape.

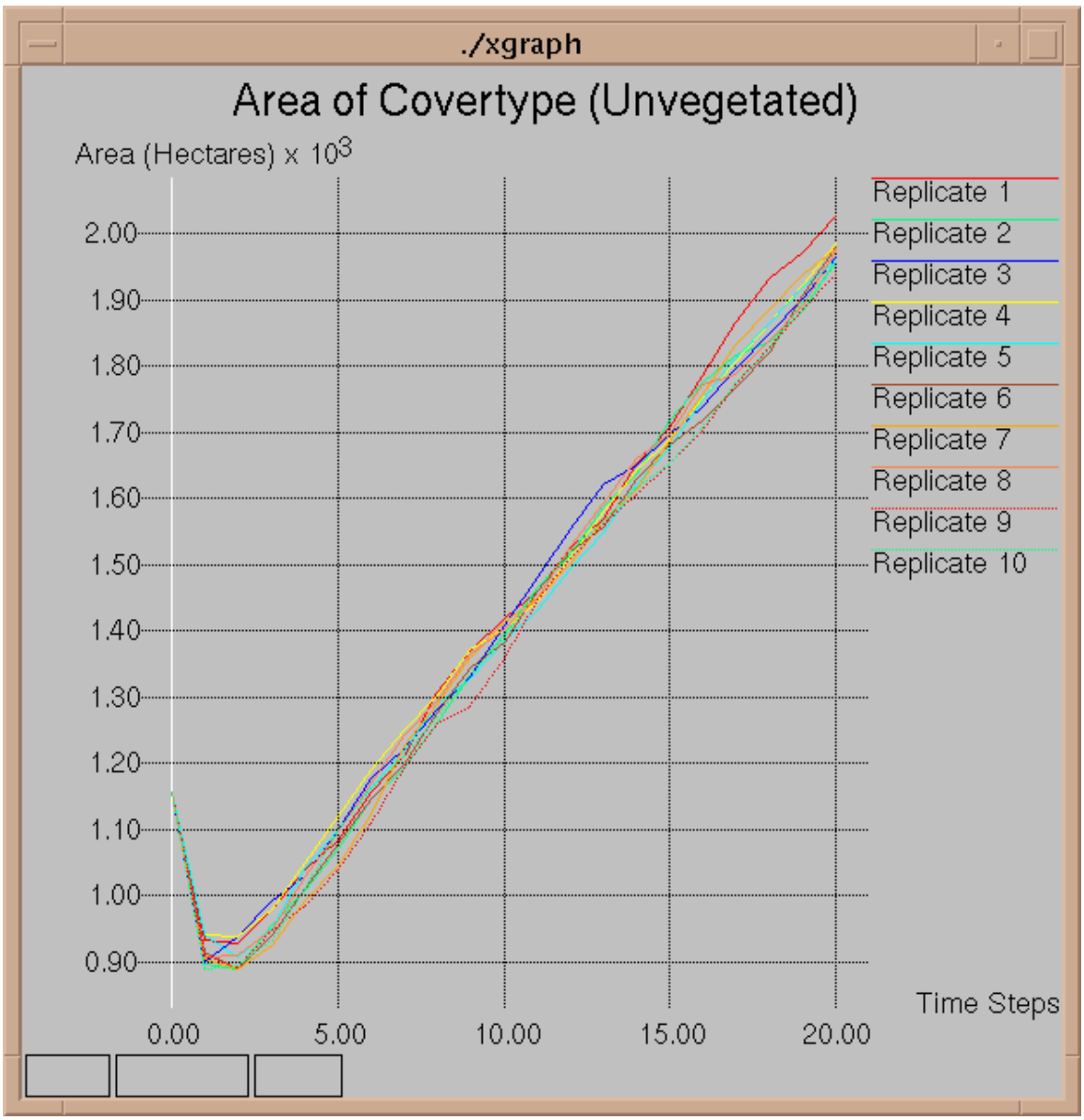

Figure 4.7: Area of Unvegetated Landscape.

Chapter 5

Current Platforms and Future Work

5.1 Current Platforms

One of the initial design goals for the LUCAS graphical user interface was portability across many computing platforms. After considering several commercially-available windowing toolkits, the Motif toolkit by the Open Software Foundation (OSF) was chosen. The OSF/Motif toolkit is available on most hardware platforms, including DEC, Hewlett-Packard, IBM, SGI, and SUN. This allowed development of the interface on a single platform, yet the capability of cross-compilation onto the platform required by users.

The LUCAS GUI was developed on a SUN SparcStation model 10 computer at the University of Tennessee, Knoxville. The configuration included 32MB of main memory and 1 GB of disk space. The software configuration includes: SunOS 4.1.3, X-Windows Version 11 Release 5 and IXI Motif 1.2. Because LUCAS manipulates large amounts of map data, the SparcStation 10 and extra disk space provided an excellent development platform.

5.2 Future Work

Currently, the impacts module (as discussed in Section 1.2) includes habitat suitability models as related to species biodiversity. A water quality impact model has also been developed for future incorporation into LUCAS. This module will facilitate studies in the impact of land-use decisions on the quality of the water supply. The user interface, of course, would be modified to reflect the addition of the water quality impact model.

Sixteen scenarios have been developed for the Little Tennessee River Basin water-

shed (documented in Appendix C). In the near future, a number of scenarios will be defined and developed for the Hoh and the Dungeness watersheds in the Olympic Peninsula of northwestern Washington state. When these scenarios are completed they will be encorporated into LUCAS. By placing these scenario files in the appropriate directory (see Section 3.1.4) and adhering to the scenario file naming convention (see Section 3.1.2), these new scenario files will be automatically included in the *Scenario* scrolled list selection on the LUCAS user interface.

Future enhancements in the underlying LUCAS analysis software include the use of a distributed computing environment such as the Parallel Virtual Machine (or PVM) [BDG+91]. Currently, all of the map data used by LUCAS is stored on a local disk of the workstation on which LUCAS has been installed. Another future enhancement to LUCAS is the ability to access map data distributed across a network of workstations.

Future users may desire access to LUCAS from a PC-based environment. Utilization of LUCAS from a PC running an X-Windows emulation package is straight forward. The user in this case would access a workstation running the LUCAS software and would set the X Windows *DISPLAY* environment variable to the network address of the PC. Porting the LUCAS software to execute on a PC would require a Graphical Information System (such as GRASS) on the PC, the modification of the analysis software to use the GIS, and rewriting the user interface for a PC Windowing environment.

Bibliography

[B+92] M. W. Berry et al. The Land-Use Change Analysis System (LUCAS) For Evaluating Landscape Management Decisions. Technical Report CS–94–238, University of Tennessee, Knoxville, TN, July 1992.

[BDG+91] A Beguelin, J Dongarra, G Geist, R Manchek, and V Sunderam. Solving Computational Grand Challenges Using a Network of Supercomputers. In D Sorensen, editor, *Proceedings of the Fifth SIAM Conference on Parallel Processing*. Philadelphia, PA: SIAM, 1991.

[CFL87] R. N. Coulson, L. J. Folse, and D. K. Loh. Artificial Intelligence and Natural Resource Management. *Science*, 237:262–267, 1987.

[Har89] David Harrison. *Xgraph Version 11.3.2*. University of California Berkeley Electronics Research Laboratory, December 1989.

[Hol78] C. S. Holling. *Adaptive Environmental Assessment and Management*. John Wiley & Sons, New York, 1978.

[LFT+92] R. G. Lee, R. O. Flamm, M. G. Turner, C. Bledsoe, P. Chandler, C. Deferrari, R. Gottfried, and R. J. Naiman. Integrating Sustainable Development and Environmental Vitality: A Landscape Ecology Approach. In R. J. Naiman, editor, *New Perspectives in Watershed Management*, pages 499–521. Sprinter-Verlag, New York, 1992.

[Mac94] R. L. MacIntyre. A Graphical User Interface for the Land-Use Change Analysis System. Master's thesis, The University of Tennessee, Knoxville, TN, December 1994.

[S+94] M. Shapiro et al. *GRASS Version 4.1 Programmer's Manual*. U.S. Army Corp. of Engineer's Construction Engineering Research Laboratory, Champaign, IL, March 1994.

[SAS92] SAS Institute Inc. *SAS User's Guide: Statistics*. Cary, NC, 1992.

[WTF94] D. N. Wear, M. G. Turner, and R. O. Flamm. Ecosystem Management in a Multi-ownership Setting. *Ecological Applications*, 1994. Submitted.

Appendix A

User Defaults

LUCAS provides a default value for each selection available on the *Setup Menu*. These defaults are listed below.

- Watershed: the watershed which is parsed from the first scenario file located in the directory indicated by the `LUCAS_DIR` environment variable concatenated with `SCENARIOS`.
- Mapset: the mapset which is parsed from the first mapset directory located in the directory indicated by the `GISDBASE` environment variable concatenated with the selected watershed and on which the user has ownership.
- Scenario: the scenario title which is parsed from the first scenario file located in the directory indicated by the `LUCAS_DIR` environment variable concatenated with `SCENARIOS`.
- Number of Replicates: 1
- Number of Timesteps: 1
- Number of Maps Saved per Replicate: 0
- Landscape Change Method: pixel-based
- Beginning Map Year: 1991
- Species: none

Appendix B

X Windows Resources

X Window resource resolution occurs as a tiered process where each higher level of the tier can override the lower levels. Following is the order of resource resolution, with the top of the tier being hard-coded values:

1. fallback resources
2. resource file(s)
3. command line values
4. hard-coded values

LUCAS provides a set of fallback resources, a resource file (`LUCAS.rf`), and hard-coded (unmodifiable) resource values. The fallback resources can be overridden by resource values located in resource files and on the command line, but this is not recommended as it may alter the user interface in such a manner as to make it unusable. A list of the fallback resources is located in `LUCAS.c`. The `LUCAS.rf` resource file can be customized by the user to create the required *look-and-feel* of the user interface. A list of the resources in the resource file and the interfaces that they effect is provided below.

```
Resources for all LUCAS Interfaces

LUCAS*IconName: LUCAS
LUCAS*background: midnightblue
LUCAS*foreground: wheat
LUCAS*TroughColor: lightslategrey
LUCAS*highlightColor: green
LUCAS*borderColor: red
LUCAS*bottomShadowColor: orange
LUCAS*topShadowColor: darkorange2

Resources for the Setup Menu Interface

LUCAS*topLevelShell.width:  900
LUCAS*topLevelShell.height: 750
LUCAS*topLevelShell.x:      150
LUCAS*topLevelShell.y:       25
LUCAS*topLevelShell*FontList:
-adobe-helvetica-bold-r-*-*-24-*-*-*-*-*-*-*
```

```
Resources for the Analysis Status Interface

LUCAS*topLevelShell2.x:     150
LUCAS*topLevelShell2.y:     225
LUCAS*topLevelShell2*FontList:
-adobe-helvetica-bold-r-*-*-34-*-*-*-*-*-*-*
LUCAS*watershedstatusL.fontList:
-adobe-helvetica-bold-r-*-*-34-*-*-*-*-*-*-*
LUCAS*scenariostatusL.fontList:
-adobe-helvetica-bold-r-*-*-34-*-*-*-*-*-*-*
LUCAS*currepstatusL.fontList:
-adobe-helvetica-bold-r-*-*-34-*-*-*-*-*-*-*
LUCAS*basemapstatusL.fontList:
-adobe-helvetica-bold-r-*-*-34-*-*-*-*-*-*-*
LUCAS*curmapstatusL.fontList:
-adobe-helvetica-bold-r-*-*-34-*-*-*-*-*-*-*

Resources for the Graphs Selection Interface

LUCAS*topLevelShell5.x:      45
LUCAS*topLevelShell5.y:     345
LUCAS*topLevelShell5*FontList:
-adobe-helvetica-bold-r-*-*-24-*-*-*-*-*-*-*
```

Appendix C

Little Tennessee River Basin Scenario Description

Sixteen scenarios have been defined for the Little Tennessee River Basin watershed. These pre-defined scenarios are described below.

Four Transition Modules (TM) used to define the scenarios:

1. Historical transition probabilities for the period 1986-1991.
2. Historical transition probabilities for the period 1975-1986.
3. Spatial limitations on transition from FOREST cover. For this module, forest area that is adjacent to a stream or on slopes greater than 20 percent cannot change to a NONFOREST cover. All other areas will have historical transition probabilities for 1986-1991 applied.
4. No transition from FOREST cover.

Definition of Scenarios by Transition Module (TM) assignments:

SCENARIO Number	Ownership Type	
	Private	USFS(Public)
1	TM 1	TM 1
2	TM 2	TM 1
3	TM 3	TM 1
4	TM 4	TM 1
5	TM 1	TM 2
6	TM 2	TM 2
7	TM 3	TM 2
8	TM 4	TM 2
9	TM 1	TM 3
10	TM 2	TM 3
11	TM 3	TM 3
12	TM 4	TM 3
13	TM 1	TM 4
14	TM 2	TM 4
15	TM 3	TM 4
16	TM 4	TM 4