A DYNAMIC OBJECT-ORIENTED ARCHITECTURE APPROACH TO ECOSYSTEM MODELING AND SIMULATION*

Prepared by

Pamela J. Sydelko, J.E. Dolph, and T.N. Taxon
Argonne National Laboratory
Argonne, IL

and

K.A. Majerus
U.S. Army Construction Engineering Research Laboratory
Champaign, IL

for

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.

A DYNAMIC OBJECT-ORIENTED ARCHITECTURE APPROACH TO ECOSYSTEM MODELING AND SIMULATION

Pamela J. Sydelko
Environmental Scientist
Argonne National Laboratory
9700 S. Cass Avenue, Bldg 900
Argonne, Il 60439
sydelko@dis.anl.gov

Kimberly A. Majerus
Forester
U.S. Army Construction Engineering Research Laboratories
P.O. Box 9005
Champaign, IL 61826
k-majerus@cecer.army.mil

Jayne E. Dolph
Environmental Computer Scientist
Argonne National Laboratory
9700 S. Cass Avenue, Bldg 900
Argonne, Il 60439
dolph@dis.anl.gov

Thomas N. Taxon
Software Engineer
Argonne National Laboratory
9700 S. Cass Avenue, Bldg 900
Argonne, Il 60439
taxon@dis.anl.gov

ABSTRACT

Modeling and simulation in support of adaptive ecosystem management can be better accomplished through a dynamic, integrated, and flexible approach that incorporates scientific and technological components into a comprehensive ecosystem-modeling framework. The Integrated Dynamic Landscape Analysis and Modeling System (IDLAMS) integrates ecological models and decision support techniques, through a geographic information system (GIS)-based framework. The Strategic Environmental Research and Development Program (SERDP) sponsored the development of IDLAMS. Initially built upon a GIS framework, IDLAMS is migrating to an object-oriented (OO) architectural framework. An object-oriented architecture is more flexible and modular. It allows disparate applications and dynamic models to be integrated in a manner that minimizes (or eliminates) the need to rework or recreate the system as new models are added to the suite. In addition, an object-oriented design makes it easier to provide run-time feedback among models, thereby making it a more dynamic tool for exploring and providing insight into the interactions among ecosystem processes. Finally, an object-oriented design encourages the reuse of existing technology because OO-IDLAMS is able to integrate disparate models, databases, or applications executed in their native languages. Reuse is also accomplished through a structured approach to building a consistent and reusable object library. This reusability can substantially reduce the time and effort needed to develop future integrated ecosystem simulations.

BACKGROUND

Military land managers and decision makers are recognizing the need for more strategic and adaptive approaches to integrated natural resources and ecosystem management. Decision makers face an ever-increasing challenge to balance maximum flexibility for the mission with a multitude of other land use, social, political, and economic goals. In addition, these goals encompass environmental requirements for maintaining ecosystem health and sustainability over the long term. To meet these challenges, simulation models, geographic information systems (GISs), remote sensing applications, scientific visualization, and decision analysis techniques are fast becoming tools of the trade. However, when used individually, these tools are often limited because they evaluate potential impacts for only one ecosystem characteristic at a time, while holding the remainder of the system static. Effective ecosystem management relies upon assessing several components of the system simultaneously, in terms of their relationship to each other and the way that they are affected by land use and land management decisions. The premise of our research is that integrated natural resource and ecosystem-based management can be better accomplished through the use of a more dynamic, integrated, and flexible approach that incorporates the appropriate scientific and technological components into a comprehensive ecosystem-modeling and technology framework.

Other integrated modeling systems have been developed or are being developed for use in land use planning and natural resource management (Berry, et al., 1996; Maxwell and Costanza, 1995; Bennett, et al., 1996; Frysinger, et al., 1996; and Fedra, 1996).

IDLAMS

In 1994, the Strategic Environmental Research and Development Program (SERDP) sponsored the development of the Integrated Dynamic Landscape Analysis and Modeling System (IDLAMS) to address the military's need for a model-integration framework (Li, et al., 1998). Built upon an effort funded by Fort Riley, Kansas, IDLAMS was developed as a multiagency partnership between the U.S. Department of Energy's (DOE's) Argonne National Laboratory (ANL) and the U.S. Army's Construction Engineering Research Laboratories (USACERL). IDLAMS is a prototype computer decision support system that integrates ecological models and decision support techniques and is based upon a GIS framework. IDLAMS predicts land conditions (e.g., vegetation, wildlife habitats, and erosion status) by simulating changes in military land ecosystems for given training intensities and land management practices. Military land managers can use IDLAMS to help predict the future ecological condition for a given land use on the basis of various land management scenarios and levels of training intensity. Land managers can also use IDLAMS as a tool to compare different land management practices and to further determine a set of land management activities and prescriptions that best suits the needs of a specific military installation.

In the initial GIS-based IDLAMS, four major models were developed and integrated; a vegetation dynamics model, a set of wildlife habitat suitability models, an erosion model, and a scenario evaluation module. Brief descriptions of these components and models follow.

> Vegetation Dynamics Model: The Vegetation Dynamics Model is the core model for IDLAMS because the output from this model is the input for all other connected IDLAMS models. The Vegetation Dynamics model is a spatially explicit model that incorporates vegetation changes due to (1) natural succession, (2) land use impacts, and (3) land management actions.

> Wildlife Models: These six submodels represent individual wildlife species and are based on U.S. Fish and Wildlife Service Habitat Suitability Indices (HSI). Each submodel requires the user to input either a vegetation/land cover map representing the current condition or a simulated landcover map generated by the Vegetation Dynamics Model. In some submodels, additional input maps may be required.

Erosion Model: IDLAMS currently integrates the Revised Universal Soil Loss Equation (RUSLE) to generate an erosion status map for each current condition or simulated vegetation/land cover map input by the user. RUSLE also requires other spatial data representing various factors affecting erosion.

Scenario Evaluation Module: IDLAMS uses a value-based decision-analysis process to link the ecological models with the management needs and user requirements of the resource manager. This component of IDLAMS requires (1) developing operational definitions for multiple land use and management objectives, (2) determining and prioritizing a list of relevant objectives to be used, and (3) defining measures to estimate the extent to which those objectives have been reached. This information is then used to perform trade-off analysis for land management alternatives on the basis of the results from the spatially explicit modeling and to rank the alternatives according to how well they meet the specified objectives.

In 1998, SERDP sponsored an IDLAMS workshop to discuss lessons learned from the IDLAMS development and to identify future research priorities. Workshop participants represented the military user community, Army research laboratories, and other SERDP model developers. Some of the workshop's resultant goals for future research and enhancements to IDLAMS include the following:

- More flexible and adaptive mechanisms for integrating disparate existing software applications;
- Improved ability to reflect the dynamics of living ecosystems, land uses, and land management practices;
- Capability to support software applications that can operate at multiple spatial and temporal scales; and
- Reduction in the long-term cost of modeling technology by use and reuse of existing data, models, and system components.

OO-IDLAMS

To accomplish these goals, a new SERDP-funded effort has been undertaken by ANL and USACERL to evaluate a prototype of a next-generation integration architecture using an object-oriented approach. This prototype is called Object-Oriented (OO)-IDLAMS. The flexibility, dynamics, and interoperability gained through an object-oriented approach have the potential to provide key technology solutions for addressing many of the military's goals and needs for integrated natural resource planning and ecosystem management.

OO-IDLAMS is built upon an object-oriented architecture called the Dynamic Information Architecture System (DIAS) (Figure 1). This generic architecture was designed and developed at ANL. DIAS supports distributed, dynamic representation of interlinked environmental processes and behaviors at variable scales (spatial and temporal) of resolution and aggregation. The generic DIAS development effort has been sponsored by several organizations and partners so that all users serve to benefit from resources leveraged through cooperation and DIAS contributions.

The modeling domain of OO-IDLAMS is flexible and is determined by available objects and by the suite of models and other data processing applications that have been "registered" within OO-IDLAMS to address specific information processing needs. OO-IDLAMS also employs an object-oriented GeoViewer module that is designed to navigate within an OO-IDLAMS study area/frame to create, query, view, and manipulate objects. The key elements of the OO-IDLAMS architecture are described below.
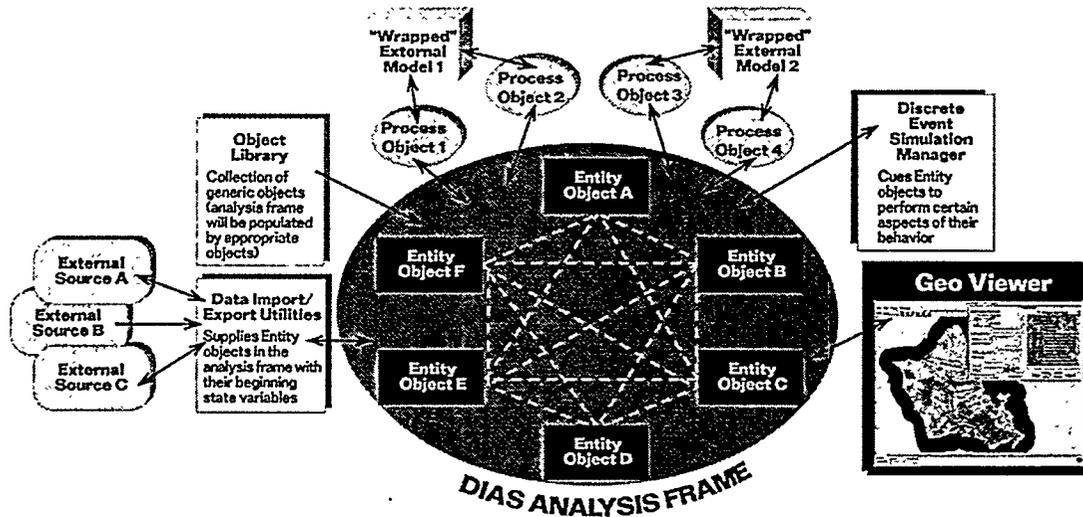
Figure 1. OO-IDLAMS Conceptual (based upon the Dynamic
Information Architecture System (DIAS))

Key Design Elements of the OO-IDLAMS Architecture

Object-oriented design elements offer the key to flexibility and dynamics in bringing together disparate data and computer applications for integrated resource management to support the military mission over the long term.

Entity Object: A domain (discipline-specific) object (e.g., grassland, watershed, stream, atmospheric layer, etc.) encapsulating state/properties and many possible aspects of dynamic object behavior. Entity objects carry both state variables and behavior:

> *State variable*: Properties of an Entity object and the current value of these properties
> *Behavior*: How an Entity object acts and reacts in terms of state changes to an external event.

Analysis Frame: The object representing the area of interest in a simulation. In many cases, the analysis frame is a geographic boundary of an area.

Object Library: A permanent collection of Entity objects. These Entity objects are brought into the analysis frame for use during simulations. When new applications are registered into OO-IDLAMS, new objects may need to be added or existing library objects may need to be edited. In this way, the object library will grow, and the potential for reuse increases over time.

Data Import/Export Utilities: A mechanism that supplies Entity objects with the current values for their state variables. These data-ingestion utilities have been developed to supply the object state variables from a variety of external data sources.

Process Object: An object providing the means of addressing one or more particular Entity object behaviors. This object includes a formal description of the software application that represents the behavior. For the Grassland Entity object example, Process objects can be created to connect the Entity object behaviors with the software application that can address them. The Process object must also include any code necessary to deal with how the time scale should be aggregated or disaggregated on the basis of the context of the simulation (see discussion of Context object).

Discrete Event Simulation Manager: The Discrete Event Simulation Manager processes events in a time-ordered queue. Events are created principally by Entity behavior and by user interactions.

Events have a time stamp and may have data associated with them. They are selectively broadcast to objects that have registered interest in certain Event types at simulation initialization.

GeoViewer: A module that provides object-oriented GIS functionality to navigate within an Analysis Frame and to create, query, view, and manipulate objects within that frame.

Context Object: The object that specifically lays out the context of the simulation. This object contains a method that links up the behavior of Entity objects in the Analysis Frame with their associated Process objects and specifies any dependencies between them. These dependencies include any code that needs to be implemented to deal with any discrepancies in the time scale. For example, if two models in the current suite operate on different time steps, the Context object would include a pointer to the algorithm (coded in the Process object) that aggregates or disaggregates the time scales.

Model Controller Object: The Model Controller object provides an explicit linkage to external application source codes and data structures. It is the "wrapper" used to communicate between the Process object and an external model (or process/subroutine of that model).

Course-of-Action (COA): A COA object is essentially a flowchart of individual steps constituting a specific plan or action. COAs have many uses and can be used to model procedural or sequential processes. COAs, like Process objects, are used to represent specific behaviors of Entity objects.

Spatial Data Set: A Spatial Data Set is an object that carries a complete geometric specification for a two-dimensional or three dimensional (2-D or 3-D) spatial partitioning scheme that divides a region into 2-D or 3-D cells. It also contains a collection of data elements (attributes) for each cell. The Spatial Data Set is tied to a Projection object (see below) if aggregation or disaggregation of spatial scale is needed for the context of the simulation.

Projection Object: This object is associated with a spatial data set object and is responsible for implementing any algorithm necessary to deal with multiple scales. The domain expert should be responsible for providing the correct mechanisms for implementing this disaggregation or aggregation within the context of the simulation.

Registering External Applications

Fortunately, a wealth of existing (legacy) applications are available. These applications have been developed and have often been validated by scientists with extensive knowledge of their specific subject domains. It is important to allow for the reuse of these models/applications without expensive reworking or recoding. Application registration through OO-IDLAMS provides for easy integration of legacy-type applications (e.g., simulation models, GIS applications, database management systems). Once an application has been registered with OO-IDLAMS, it can interact, as appropriate, with OO-IDLAMS Entity objects within a simulation. Applications used within OO-IDLAMS do not directly communicate among themselves but interact indirectly with one another via Entity objects. This interaction provides for a more stable environment because the disruption to the suite of models is limited when a model is added or removed. A further advantage is that OO-IDLAMS external applications are executed in their native languages (e.g., FORTRAN, C, etc.). OO-IDLAMS can even be used to integrate disparate models, databases, or applications that can reside on multiple machines across the Internet. Model registration within OO-IDLAMS is a formalized process that allows model developers to register their models with OO-IDLAMS while retaining full control of their source codes. The model registration procedure consists of the following five steps:

1. Formally defining the application in terms of
   - Attributes/states of domain objects
   - Processes represented as behaviors of the domain objects
   - Input/output parameters to the processes,
2. Instrumenting a model source code (flag/delineate individual processes),
3. Writing a model controller (or wrapper) for the model,

4.   Building (or editing existing) domain Entity objects, and

5.   Building Process objects (one for each process delineated in the source code).

OO-IDLAMS Prototype

In the original IDLAMS framework (Li, et al., 1998), integration is accomplished through heavy reliance on a static GIS structure, which limits flexibility and the potential for truly dynamic interaction among the models being integrated. To overcome some of these limitations, ANL and USACERL have undertaken a move toward an object-oriented approach (OO-IDLAMS). An OO-IDLAMS prototype is being developed to demonstrate the key benefits of an object-oriented approach. The study area for the prototype will be Fort Riley, Kansas, which is the same study area used in the original IDLAMS application.

The OO-IDLAMS prototype integrates a subset of the original IDLAMS, including the Vegetation Dynamics Model and the Henslow's Sparrow Habitat Model (reimplemented as an Environmental Systems Research Institute (ESRI®) application). For the OO-IDLAMS implementation, the Military Training and Land Management components of the original Vegetation Dynamics Model were broken out into three COA objects within OO-IDLAMS (Training, Burning, and Planting). These COAs represent the Fort Riley land use and land management plans. These plans are inherently procedural in nature and readily lend themselves to COA implementation. COAs are considered to be models within OO-IDLAMS. The natural succession processes remain part of the external Vegetation Dynamics Model and are registered with OO-IDLAMS as the external Vegetation Dynamics Model. Conceptually, the natural functions of vegetation dynamics will not need to be frequently changed over time and should remain as part of the external model. On the other hand, the aspects of land use and land management may change frequently over time; this should not pose any difficulties because COAs can be easily edited.

Figure 2 illustrates a very important object-oriented design principle. The models do not talk directly to one another but are only integrated through their relationships with the Entity objects. This makes it much easier to replace a model or integrate additional models without disrupting any other models in the suite. This modular design principle makes OO-IDLAMS much more flexible than the earlier GIS-based version.
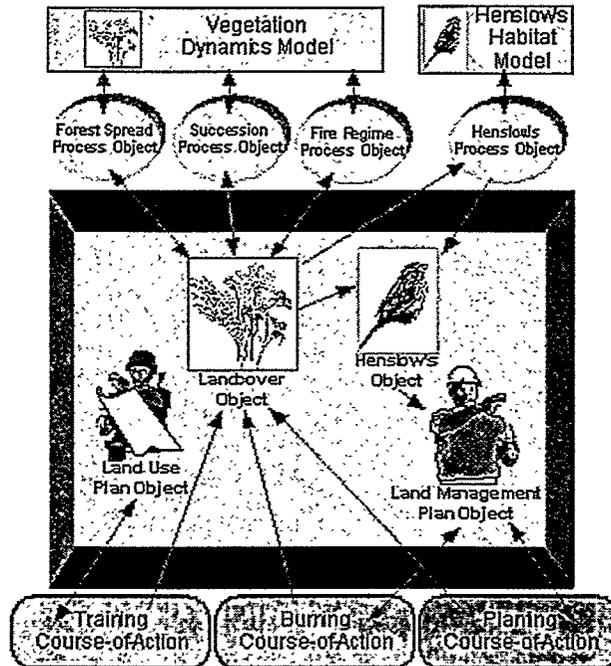

Figure 2. OO-IDLAMS Prototype Conceptual Design

Four Entity objects are needed for the currently designed implementation of OO-IDLAMS: Landcover, Land Management Plan, Land Use Plan, and Henslow's Sparrow. These objects include state variables and behavior needed for the models in the OO-IDLAMS suite. More Entity objects can be added if additional models or applications are added to the OO-IDLAMS suite. Some model input and output parameters are not held as Entity state variables. Only input or output that will be needed by more than one model must be state variables. If, however, other models were to be registered that require this input or produce these variables as output, Entity objects could be easily edited to include them.

This ability to utilize existing objects illustrates the object-oriented design principle of reusability. As the object library for OO-IDLAMS grows, the potential for reuse grows. As new models are added, it may be as easy as pulling an existing Entity object from the library and simply editing it to include new variables and behavior specific to the new model. Another benefit of object-oriented design is the improved ability to reflect the dynamics among living ecosystems, land uses, and land management practices, because the current state of these objects is always used by any of the models in the suite.

For the purposes of this paper, details about the individual models are not discussed. Rather, only model input and output parameters that directly correspond to Entity state variables or behavior are discussed.

The Landcover Entity object holds six state variables and three behaviors for use in the OO-IDLAMS suite of models (Figure 3).
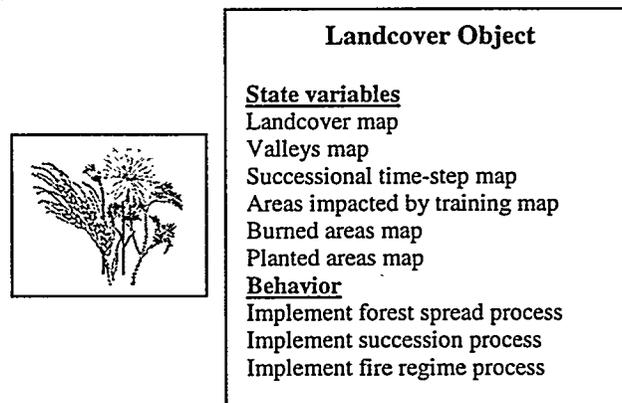

Landcover Object

State variables
Landcover map
Valleys map
Successional time-step map
Areas impacted by training map
Burned areas map
Planted areas map
Behavior
Implement forest spread process
Implement succession process
Implement fire regime process

Figure 3. Landcover Object State Variables and Behaviors

State Variables:

1. *Landcover map* – Spatial Data Set object that holds the attributes for landcover categories. This state variable is used as an input to all the models and is updated by the "update landcover map" behavior invoked by several models in the following ways:

    • The Vegetation Dynamics Model updates this state variable in several ways. It can change prairie to forest by the forest spread process. The succession process can change vegetation categories by allowing them to change from one category to another. The fire regime process controls whether prescribed burning will change a forested area to prairie.

    • The Training COA can update this state variable by setting vegetation categories to damaged due to training impact.

- The Planting COA can update this state variable by changing damaged areas to early or late successional grasslands, depending on what is planted (erosion control or native species).

- The Burning COA can update the *landcover map* state variable because burning changes forest to late successional prairie. In addition, because burning also accelerates the successional time-step for an area (see below), it can change the landcover state if the time-step of an area is almost ready to change to the next category.

2. *Valleys map* – Spatial Data Set object that defines where valleys exist. This variable is used in the forest spread process of the Vegetation Dynamics Model and the Burning COA.

3. *Successional time-step map* – Spatial Data Set object that holds the successional time-step attribute. Succession within the Vegetation Dynamics Model proceeds from damaged grasslands, to early successional grasslands, to late successional grasslands, to forest. The number of years spent in each stage is a Vegetation Dynamic Model input parameter. For example, for the Fort Riley implementation, the time needed to change from early to late successional may be 15 years. Because each cell in the spatial data set has an initial successional time-step assigned to it, if an early successional cell has a successional time-step of 14 years, it will change the next year into late successional. Processes within the Vegetation Dynamics model as well as all three COAs can update the *successional time-step map* state variable either by moving it forward or setting it back.

4. *Areas impacted by training map* – Spatial Data Set object that holds the output from the Training COA (see below). These areas are used to determine where to plant in the Planting COA.

5. *Burned areas map* – Spatial Data Set object of burned areas. This map is an output of the Burning COA. The *burned areas map* is used by the Henslow's Sparrow object, because recently burned areas are considered poor habitat in the Henslow's Sparrow habitat model. In addition, the *burned areas map* for each year can be saved for viewing after the simulations. It can also be used to create animated visualizations that show the burning regime over time.

6. *Planted areas map* – Spatial Data Set object of planted areas. This map is an output from the Planting COA. Currently, the *planted areas map* is not used directly by another model. However, because the history of areas planted may be of interest to land managers, it is saved as a state variable for visualization purposes. As for the *burned areas map*, the *planted areas map* can also be viewed and used in animations.

Behavior:

1. "Implement Forest Spread Process" – This behavior will implement the forest spread process of the Vegetation Dynamics model. The three processes of the Vegetation Dynamics model have been registered as separate processes in OO-IDLAMS. This demonstrates the modularity of the system. If at some time, the Vegetation Model needs to be updated with a new forest spread algorithm, but all other processes of the model remain valid, only this process will need to be changed to call another algorithm.

2. "Implement Succession Process" – This behavior implements the succession process of the Vegetation Dynamics model.

3. "Implement Fire Regime" – The fire regime process of the Vegetation Dynamics Model is invoked by this behavior.

The Land Management Plan Entity object holds one state variable and two behaviors (Figure 4).
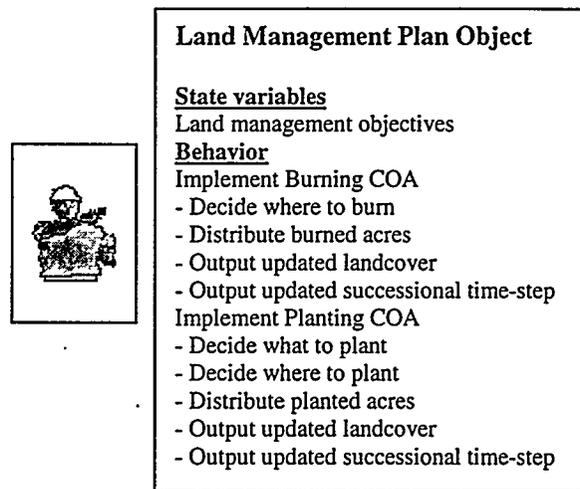

Figure 4. Land Management Plan Object State Variables and Behaviors

State Variables:

Land management objectives – This state variable is a collection of all land management objectives the land user cares about and a ranking of their importance. Currently the only objective available in this prototype is managing the Henslow's Sparrow. As new models are added to the suite, it may include management of other species or management for erosion control. Activating or deactivating these objectives can cause different COAs to be invoked.

Behavior:

1. "Implement Burning COA" – This behavior simply initiates the Burning COA. The Burning COA acts as a proxy for the land manager's burning plan. The Burning COA will decide where to burn (based on a burn rotation map and hectares to be burned) and will burn it, and it will update the *landcover map* and *successional time-step map* (held as state variables of the Landcover object).

2. "Implement Planting COA" – This behavior initiates the Planting COA. This COA acts as a proxy for the land manager's planting plan. It is slightly more complex than the Burning COA because the Henslow's Sparrow is a management objective (if selected). Once the COA has established what to plant (erosion control or native species), it then decides where to plant it. If native species are to be planted, the Planting COA will "ask" the Henslow's Sparrow object what the current state of the *patches and additional hectares needed* state variable is. This will invoke a process of the Henslow's Sparrow Habitat Model that creates patches of excellent habitat. To be considered excellent Henslow's Sparrow habitat, patches need to be at least 65 hectares in size. Planting native species next to a patch and therefore fulfilling the patch size requirement could greatly benefit the species. The planting COA will perform an analysis that will check for adjacent damaged grassland areas that can be planted to fulfill the 65-hectare requirement. If native species can be planted in such a way as to fulfill the requirement, the COA will plant to fulfill the 65-hectare patch size requirement. Viewing the *planted areas map* after the simulation can be very helpful to a land manager interested in knowing where to plant native species in order to manage the Henslow's Sparrow. In addition to the behavior discussed here, similar types of COAs can be created to perform

)

multiple species management or for balancing wildlife objectives with other objectives such as erosion control or training realism.

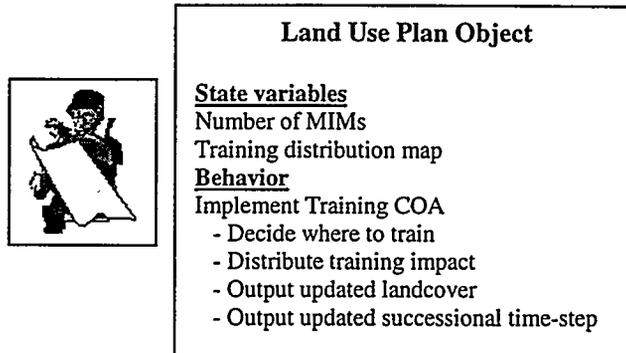The Land Use Plan Entity object holds two state variables and one behavior (Figure 5).

Land Use Plan Object

State variables
Number of MIMs
Training distribution map
Behavior
Implement Training COA
 - Decide where to train
 - Distribute training impact
 - Output updated landcover
 - Output updated successional time-step

Figure 5. Land Use Plan Object State Variables and Behaviors

State Variables: Except for the Training COA, no other models currently implemented in the OO-IDLAMS prototype need the variables held by the Land Use Plan object. However, because existing models and applications that could provide these variables have been identified, they have been included as state variables.

1. *Number of Manuever Impact Miles (MIMs)* – MIMs are calculated as part of the Army Training and Testing Area Carrying Capacity (ATTACC) program (previously known as the Evaluation of Land Value Study (ELVS) (Concepts Analysis Agency, 1996). One MIM has the equivalent impact to the land base as an M1A2 tank driving one mile in an Armor battalion field training exercise (FTX). The MIM value for each mission activity is derived from the number and types of vehicles used, the miles that the vehicles travel, and other characteristics of the activity that affect the land base, such as digging or demolition.

2. *Training distribution map* – This state variable is a Spatial Data Set object that represents the spatial distribution of land use activities. Land use activities are not distributed uniformly across the installation. The *training distribution map* should reflect the effects of topography, vegetation, and other environmental influences on the distribution of land uses such as training and testing. It should also reflect the doctrinal requirements of training and historic land use patterns.

Behavior:

1. "Implement Training COA" – The Land Use Plan object can implement the Training COA, which uses the state variables *number of MIMs* and *training distribution map* as input and outputs an updated *landcover map* and *successional time-step map* (state variables held by the Landcover object).

The Henslow's Sparrow Entity object holds two state variables and two behaviors (Figure 6).
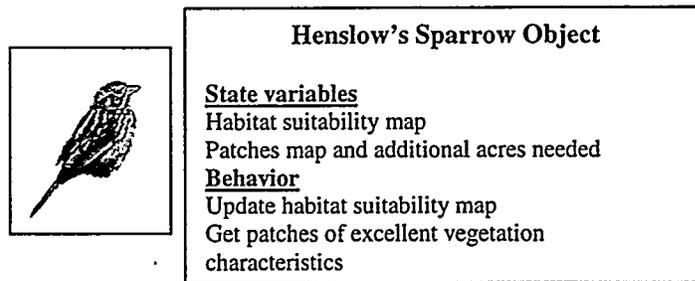

Henslow's Sparrow Object

<u>State variables</u>
Habitat suitability map
Patches map and additional acres needed
<u>Behavior</u>
Update habitat suitability map
Get patches of excellent vegetation
characteristics

Figure 6. Henslow's Sparrow Object State Variables and Behaviors

State Variables:

1. *Habitat suitability map* – This state variable is the Spatial Data Set object that results from implementing the Henslow's Sparrow Habitat Model registered to OO-IDLAMS.

2. *Patches map and additional acres needed* – This state variable combines the Spatial Data Set object calculated within the Henslow's Sparrow Habitat model and the number of hectares that are needed to fulfill the 65-hectare patch size requirement.

Behavior:

1. "Update habitat suitability map" – This behavior implements the Henslow's Sparrow Habitat model, which uses the current states of the *landcover map* and *burned areas map* (held by the Landcover object) as input.

2. "Get patches of excellent vegetation characteristics" – This behavior is implemented when the Planting COA asks for an updated state of the *patches map and additional hectares needed* state variable.

SUMMARY

Effective ecosystem management relies upon assessing the components of the system simultaneously. Modeling and simulation in support of adaptive ecosystem management can be better accomplished through a dynamic, integrated, and flexible approach that incorporates scientific and technological components into a comprehensive ecosystem-modeling framework. This framework should be flexible, allowing new data, concepts, and technologies to be incorporated without significant reworking of the system. In addition, this approach should provide run-time feedback among models, thereby making it a more dynamic tool for exploring and providing insight into the interactions among ecosystem processes. Finally, the approach should be cost effective by encouraging the reuse of existing technology and by allowing applications to be executed in their native languages. An object-oriented approach provides the flexibility, dynamics, and reusability needed for the next-generation ecosystem-modeling framework.

This research will result in an object-oriented prototype (OO-IDLAMS) to test and evaluate a next-generation software architecture. This architecture will allow for a comprehensive landscape or ecosystem approach to integrated natural resources planning and ecosystem management in support of the long-term military mission. OO-IDLAMS offers enhanced capabilities to:

- Allow for the integration of existing legacy–type models without extensive reworking, thus capitalizing upon previous investments in already available models and applications;
- Encourage the development of object libraries that contain a large number of reusable objects to represent a wide variety of natural and artificial elements of the environment and therefore reduce the long-term cost of redeveloping objects and technologies;

- Provide an integrated architecture that allows for the ability to reflect the dynamics of living ecosystems, land uses, and land management practices;
- Support software applications that can operate at multiple spatial and temporal scales; and
- Incorporate new data, concepts, and technologies that will bring the best available knowledge, science, and technology to bear on decision-making processes important to the military mission.

REFERENCES

Bennett D.A., M.P. Armstrong, and F. Weirich. (1996). An Object-Oriented Model Base Management System for Environmental Simulation. In: *GIS and Environmental Modeling: Progress and Research Issues*. Goodchild et al., eds. GIS World Books, Fort Collins, CO, pp. 439–444.

Berry, M.W., R.O. Flamm, B.C. Hazen, and R.L. MacIntyre. (1996). A System for Modeling Land-Use Change. *IEEE Computational Science & Engineering* 3(1): pp. 24–35.

Concepts Analysis Agency. (1996). Evaluation of Land Value Study (ELVS), Study Report CAA–SR–96–5, U.S. Army Concepts Analysis Agency, Bethesda, MD.

Frysinger, S.P., D.A. Copperman, and J. P. Levantino. (1996). Environmental Decision Support Systems (EDSS): An Open Architecture Integrating Modeling and GIS. In: *GIS and Environmental Modeling: Progress and Research Issues*. Goodchild et al. Eds. GIS World Books, Fort Collins, CO, pp. 357–362.

Fedra, K. (1996). Distributed Models and Embedded GIS: Integration Strategies and Case Studies. In: *GIS and Environmental Modeling: Progress and Research Issues*. Goodchild et al., eds. GIS World Books, Fort Collins, CO, pp. 413–418.

Li, Z., P.J. Sydelko, M.C. Vogt, C.M. Klaus, K.A. Majerus, and R.C. Sundell. (1998). Integrated Dynamic Landscape Analysis and Modeling System (IDLAMS): Programmer's Manual. ANL/ESD/TM – 146, Argonne National Laboratory, Argonne, IL, and USACERL–98/127, U.S. Construction Engineering Research Laboratories, Champaign, IL.

Maxwell, T., and R. Costanza. (1995). Distributed Modular Spatial Ecosystem Modelling. *International Journal of Computer Simulation, Special Issue on Advanced Simulation Methodologies*, 5(3): 247–262.