GeoFish - Tracking Trajectories of Fish Larvae

Tiffany C. Vance, NOAA/NMFS/AFSC, Seattle, Washington
Nazila Merati and Christopher W. Moore, NOAA/OAR/PMEL-UW/JISAO, Seattle, Washington
Sharon Mesick, NOAA/NCDDC, Stennis Space Center, Mississippi

INTRODUCTION

The recent emphasis on an ecosystem approach to management of commercial fisheries leads to increasing dependence on models and modeling. As models become more complex and increased computational resources allow for the routine use of three-dimensional models, interpreting the ever larger and more complex output becomes a greater challenge. The objective of this project is to investigate how ecological models and a geographic information system (GIS) can be tightly coupled to provide easier manipulation of the model parameters, rapid interaction with the model, and enhanced analysis and 3-D visualization of the results. To do this we take advantage of two existing GIS-based architectures - a particle-tracking model in the Chesapeake Bay Oyster Larvae Tracker (CBOLT) and a visualization tool called GeoModeler. Additionally, a variety of Java-based tools provide other parts of the functionality needed for truly three-dimensional displays and analyses of model output.

GeoModeler [http://gis.esri.com/library/userconf/proc06/papers/abstracts/a1848.html]is used for visualization and analysis of oceanographic and fisheries model results in 2- and 3-D. GeoModeler uses Java and a GUI to set up model display parameters, allows for 3-D display/manipulation, and provides analytical tools via Java, VTK and ArcEngine. GeoModeler is modular and can be expanded to read and analyze the output of a variety of models. The Chesapeake Bay Oyster Larvae Tracker (CBOLT) is an integrated system of components [http://gis.esri.com/library/userconf/proc06/papers/abstracts/a2095.html] that allow users to control a particle tracking model, and then examine the output as geospatial features in a Web-based map interface. CBOLT uses an ArcIMS interface to allow users to input model parameters for a particle tracking model, sends these parameters off to computational engine, returns results to a database, and then plots them in 2-D in an ArcIMS site. While CBOLT is currently a specialized system for showing oyster larvae tracking results, it was written in such a way that it could be broken into reusable components. We use these components for the initial setting of model parameters, the passing of parameters to the model and the storage of the results of model runs in a file-based geodatabase. As an example of integrating CBOLT with GeoModeler, we present results that integrate a particle tracking model for Bering Sea northern rock sole (*Lepidopsetta polyxystra*) with the GIS-based visualization and analysis tools.

BACKGROUND

The use of ecosystem analyses to understand the larger patterns in a fishery or ecosystem leads to increasing dependence on models and modeling. As models become more complex and increased computational resources allow for the routine use of three-dimensional models, interpreting the ever larger and more complex output becomes a greater challenge. Models provide a way to simplify reality to increase understanding of systems such as ecosystems. A complex system is reduced to a set of tractable questions and parameters are adjusted to simulate a variety of situations. Particle tracking models are used to study the dispersion of fish and shellfish larvae. The output is extensive and multidimensional and not amenable to straightforward interpretation. The results of models may be in file formats that are hard to integrate with other types of data and with analysis tools. Ideally, the analytical process would be rapid and iterative, with the user able to test ideas, gain immediate visual feedback and rapidly run another model run or analysis. Important variables and relationships may become apparent during this interactive process. Previous approaches to making the output data more easily analyzed include creating a MATLAB-based tool to analyze and plot two-dimensional slices of the data. But, as these types of models are more frequently producing three-dimensional output, more advanced visualization tools are now needed.

A GIS provides tools for handling disparate types of data, have strong 2-D analytical tools and provide excellent 2- and 3-D visualizations and displays. They also include new capabilities to handle netCDF files, a standard for model output, natively within the GIS. However, they are not powerful enough to run true ecosystem models and have limited abilities to analyze and display data in three-dimensions. This can be a problem as models such as the Regional Ocean Model System (ROMS) models are fully three-dimensional, and much potential information is lost when the results are shown in 2-D. Visualization tools such as Java3D, Visualization ToolKit (VTK) and GeoModeler provide 3-D visualization and analysis and allow scientists to add analyze patterns in model results. CBOLT provides easy setup of model parameters, handoff to a compute engine and storage of results and metadata in a database.

SOFTWARE COMPONENTS

GeoModeler uses Java and a GUI to set up model display parameters, allows for 3-D display/manipulation, and provides analytical tools via Java, VTK and ArcEngine. GeoModeler was developed to exploit the fact that ESRI ArcEngine exposes code and objects to allow closer coupling of core GIS functionality and analytical/modeling tools. With GeoModeler (Figure 1) we are able to directly integrate GIS and modeling capabilities in support of management and decision making. Through the use of Java-based API's and connectors, scientists and managers are able to use a GIS-based graphical interface to select the data to be used in a scenario, set the weights for factors in the model and run the model.

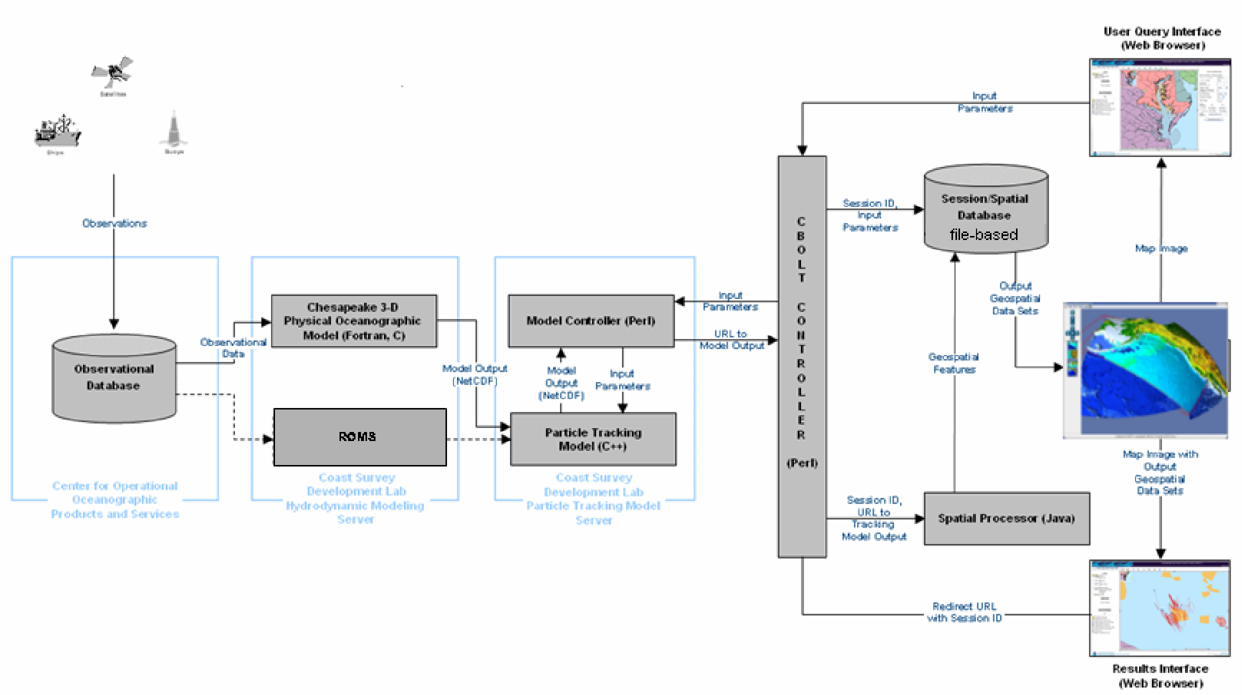

Figure 1 CBOLT/GeoFish architecture

The GeoModeler interface and display uses graphical objects to provide functionality related to the type of data being displayed. As a data layer is added, the relevant tools for analyses are exposed for use. Data can be read directly from an OPeNDAP server by the Java code in GeoModeler. GeoModeler has been used for displaying results from the Regional Ocean Modeling System (ROMS) circulation model. As with CBOLT, GeoModeler was designed in components. We use the GeoModeler Java-based components for I/O with oceanic models such as ROMS, and for 3-D display and analysis. Analysis tools include path on volume and volume on volume intersections. Display components include pan, zoom, slice, and adding and removing data layers.

We use ArcGIS Engine - Java and implementations of ArcObjects to provide spatial analytical tools. ArcEngine is a simple API-neutral cross-platform development environment for ArcObjects - the C++ component technology framework used to build ArcGIS. ArcObjects are the core of the ArcGIS functionality and include tools such as overlay (union, intersect), proximity (buffer, point distance), surface analysis (aspect, hillshade, slope), and data conversion (shapefile, coverage and DEM to geodatabase). ArcEngine's object library makes full GIS functionality available though fine and coarse-grained components that can be implemented in Java and other environments. Using ArcEngine, solutions can be built and deployed to users without requiring the ArcGIS Desktop applications (ArcMap, ArcCatalog) to be present on the same machine. It supports all the standard development environments, including Java, and C++, and all the major operating systems.

CBOLT is an integrated system of components that allow users to control a particle-tracking model, and then examine the output as geospatial features in a web-based map interface. CBOLT allows a user to input model parameters for a particle tracking model, sends these parameters off to compute engine, returns results to a database, and then plots them in 2-D. The CBOLT application (Figure 2) uses an ArcIMS interface to set up model parameters. A three-dimensional circulation model written in FORTRAN and C generates a circulation field for the particle-tracking model. The particle-tracking model is written in C++. The results of the model are in netCDF and are converted to geospatial features using a Java-based processor. They are

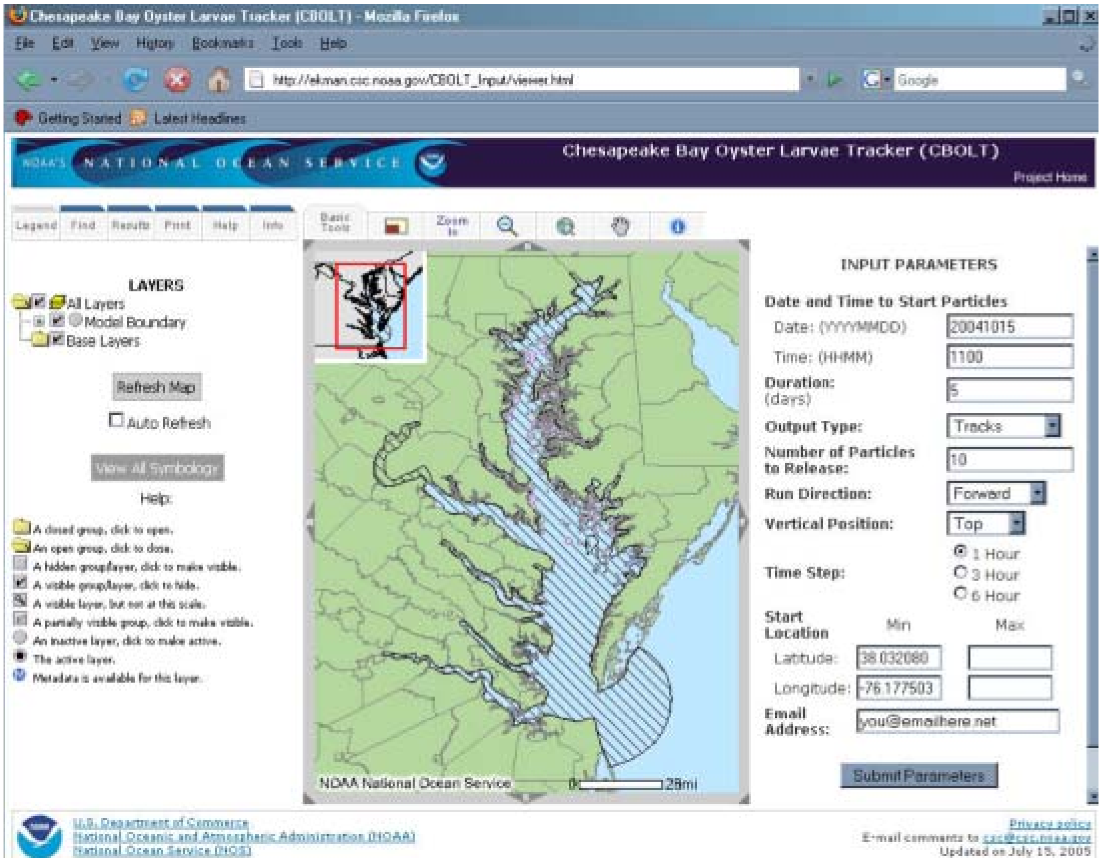

Figure 2 The CBOLT ArcIMS interface

stored in a file-based geodatabase. We also use the 2-D display components with an ArcIMS map server for map-based output.

PROTOTYPES

In creating GeoFish, we provide a fully-functional prototype of how one might integrate a GIS with a number of oceanographic and fisheries models. With this tool, scientists and managers are able to use a graphical interface to display datasets, select the data to be used in a scenario, set the weights for factors in the model and execute the model on a compute server. The results are returned to the GIS for display and spatial analysis. The project creates a framework for linking to other types of back-end fisheries, oceanographic, and ecosystem models written in a variety of programming languages. Current prototype applications include two examples - one for rock sole larvae and the other an example of particle tracking for oyster larvae. The first models the distribution and transport patterns of northern rock sole, *Lepidopsetta polyxystra*, larvae in the southeastern Bering Sea (Lanksbury et al., 2007). This application uses a regional ocean modeling system (ROMS) model, runs an associated particle tracking model and displays results draped over a three-dimensional globe. The second implementation allows the setting of sources for dispersion of oyster larvae in Chesapeake Bay using the triangular grid C3PO model [http://nauticalcharts.noaa.gov/csdl/op/c3po.html] and the display of the resulting dispersion tracks in two and three-dimensions.

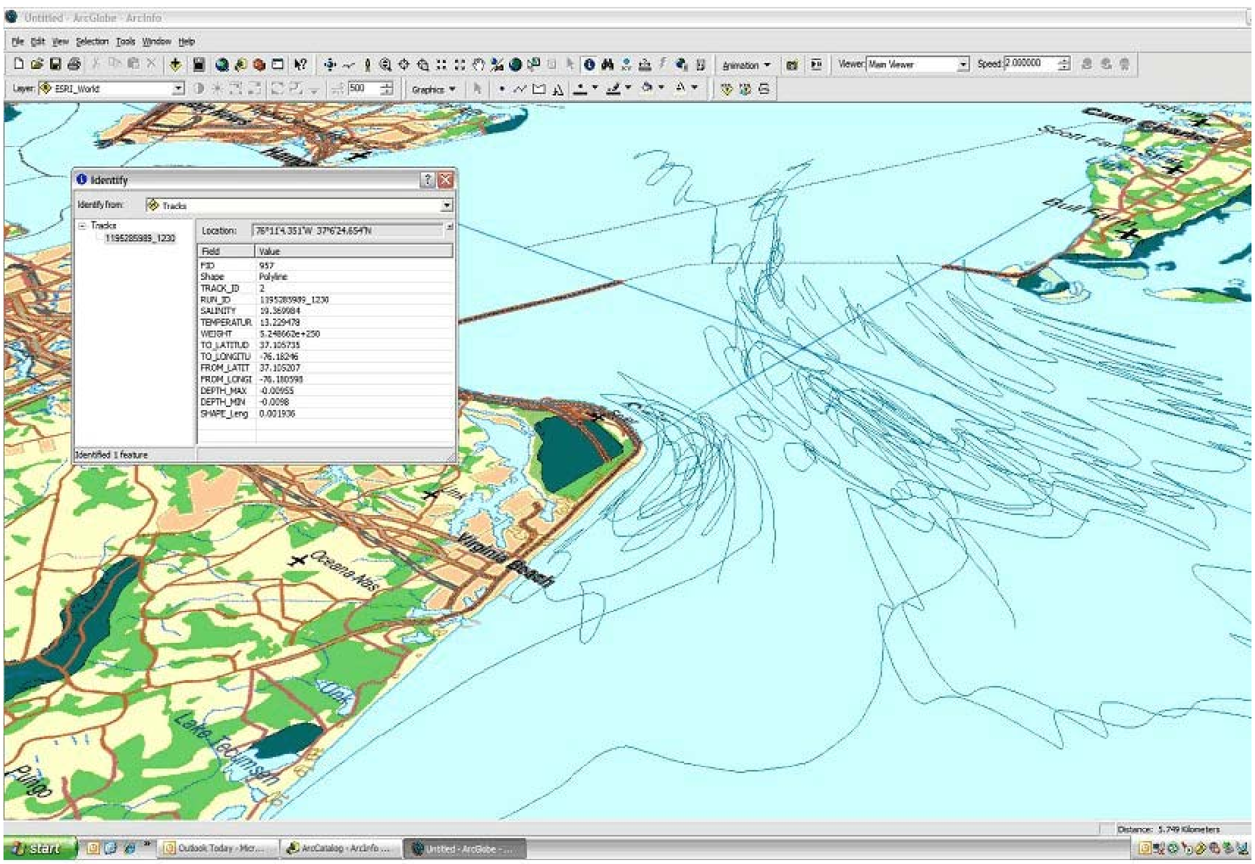

Figure 3 - 2.5-dimensional output of oyster larvae tracks from GeoFish

Both applications implement an ArcIMS and VTK-based interface to a Lagrangian particle tracking models. GeoFish uses an ArcIMS-based front end to set the source for the larvae, the duration of the model run and whether the model should be run forwards or backwards in time.

The user to "seeds" a region or area with larvae and watches how they disperse based upon a hydrodynamic circulation model. The user sets parameters for the model run including the release location for the larvae, the release data and time, the run duration and direction, the number of larvae to release and the type of spatial output. A particle-tracking model is run and the results are returned as a netCDF file, which is translated and stored as geospatial features. The output is a three-dimensional grid of the trajectories of the larvae over time. The features can be shown as a two-dimensional map in an ArcIMS (Figure 3) and as a three-dimensional visualization in GeoFish (Figure 4). The ArcIMS renderer provides analyses

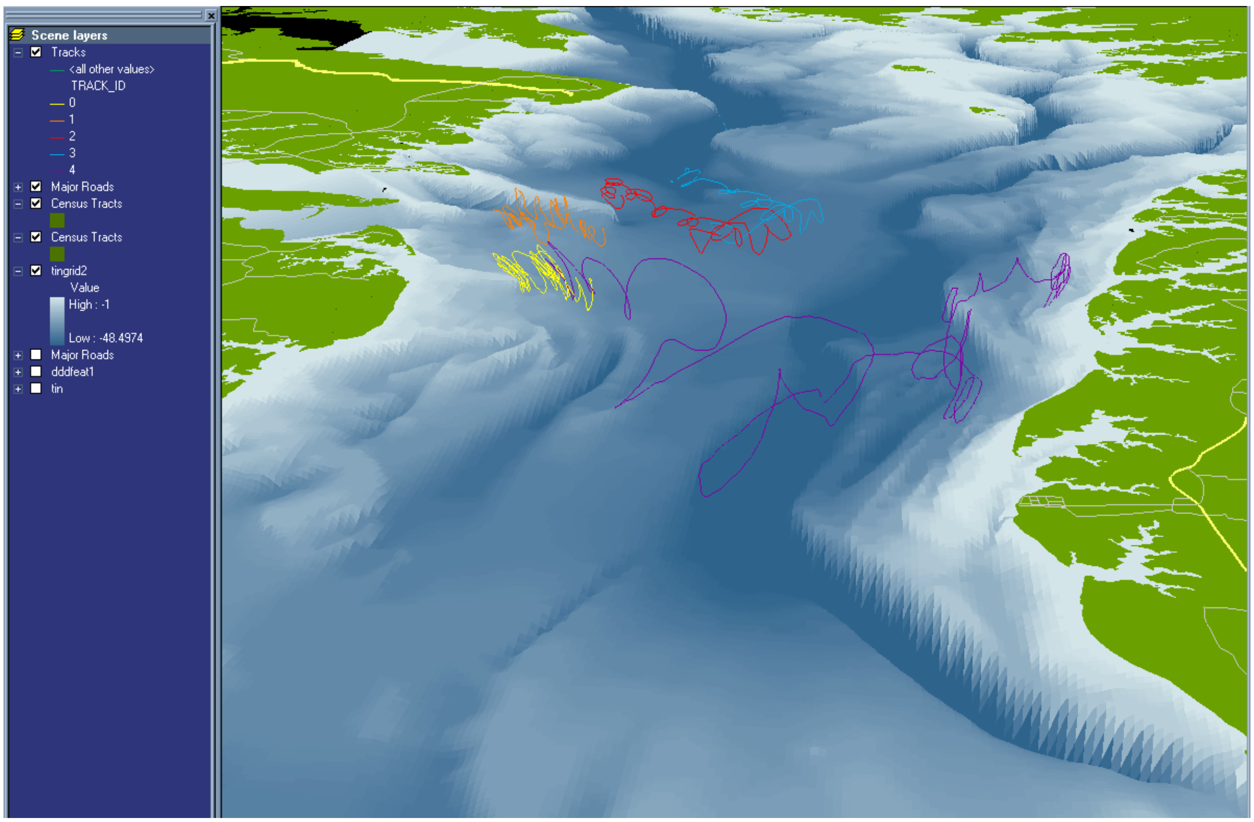

Figure 4 Three-dimensional GeoFish output for oyster larvae tracks

such as spatial queries and buffering. The GeoFish renderer allows for color-coding of the tracks, overlays of other parameters stored in shapefiles and netCDF files, and zoom, pan and rotation of the visualization for user exploration.

CONCLUSIONS

Extending GIS functionality beyond analysis to scientific modeling enhances the ability of NOAA scientists to model the interaction of physical and biological systems. The ability to set parameters for, run and visualize the results of oceanographic and fisheries models improves our ability to support ecosystem based analyses. The project takes advantage of developments both in GIS and modeling technology. ArcEngine is the state-of-the-art for integration of GIS functionality in applications and using ArcEngine for access to the ArcObjects and models enhances existing capabilities. The portability of Java allows for interaction with larger datasets and may allow for distributed computation. Fully three-dimensional visualization and analysis makes the model results more understandable to a variety of users.

ACKNOWLEDGMENTS AND DISCLAIMERS

Funding for this project was provided by the High Performance Computing and Communications (HPCC) project of the NOAA Office of the Chief Information Officer. For more details please see http://nwweb.nwn.noaa.gov/hpcc/nwg/. Additional support for this research was provided by the NOAA Fisheries-Oceanography Coordinated Investigations (FOCI) program and an NSF IGERT graduate fellowship (NSF award 0333257) in the Ecosystem Informatics IGERT program at Oregon State University. This publication was supported by the Joint Institute for the Study of the Atmosphere and Ocean (JISAO) under NOAA Cooperative Agreement #NA17RJ1232. PMEL contribution 3228. This research is contribution EcoFOCI-0664 to NOAA's Ecosystems & Fisheries Oceanography Coordinated Investigations. The views expressed herein are those of the author(s) and do not necessarily reflect the views of NOAA or any of its sub-agencies. Mention of software products does not imply endorsement of these products

REFERENCES

Lanksbury, J.A., J.T. Duffy-Anderson, K.L. Mier, M.S. Busby, and P.J. Stabeno. 2007. Distribution and transport patterns of northern rock sole, Lepidopsetta polyxstra, larvae in the southeastern Bering Sea. Prog. Oceanogr., 72, 39–62.

ESRI Arc Engine, www.esri.com , viewed 5/29/08

GeoTools home page, www.geotools.org, viewed 4/28/08.

OPeNDAP pages, www.opendap.org, viewed 4/18/08.

ROMS model pages, http://ourocean.jpl.nasa.gov, viewed 4/20/08.
http://marine.rutgers.edu/po/index.php?model= roms&page=, viewed 6/12/08.

Vance, T.C., N. Merati, S. Mesick, C.W. Moore, and D. Wright. 2007. *GeoModeler: Tightly linking spatially-explicit models and data with a GIS for analysis and geovisualization*. In 15th ACM International Symposium on Advances in Geographic Information Systems (ACM GIS 2007), Seattle, WA, 7–9 November 2007.

VTK, [www.kitware.com], viewed 5/28/08.

Contact author: Tiffany C. Vance,
NOAA/NMFS, 7600 Sand Point Way NE,
Seattle, Wa. 98115 USA
tel: +01.206.526.6767
email: tiffany.c.vance@noaa.gov