

INCREASING THE LIFESPAN OF PREDICTIVE DISEASE MODELS: THE USE OF A SOFTWARE SOCKET IN A TEMPERO-SPATIAL DISEASE DISTRIBUTION MODEL FOR ARBOVIRUSES AND THEIR VECTORS

Cameron AR

AusVet Animal Health Services, 140 Falls Road, Wentworth Falls, Australia, 2782

A large number of computer models of biological systems have been developed. Some aim to answer specific questions and are then discarded, but others are developed for the purpose of providing ongoing predictions, such as predicting the occurrence of disease in a population. While all models are imperfect simplifications of natural systems, predictive models may, in theory, be of practical value for many years after they are produced. However only a very small proportion of predictive models developed have a functional life-span much beyond the project under which they were created. There are many reasons for the failure of a predictive model to be used on a long-term basis, but one key reason is their inability to grow and adapt to change. A model usually encapsulates the level of understanding of a biological system that was possible based on the data and analysis available at the time of the model's creation. Over time we inevitably gather more data and develop a better understanding of the systems under study and models rapidly become outdated as a result.

The term 'model' may be applied either specifically to the mathematical equations which describe aspects of the biological system, or more broadly to the system implementing these equations (the 'model implementation' which includes procedures for data input, storage, processing and display). Due to the complexity of the equations, models are usually implemented using computers, typically using either a spreadsheet program, or developed in a computer programming language (such as C++, Basic or Fortran).

Spatially predictive models introduce a new layer of complexity. Not only is some aspect of a biological system, such as disease occurrence, predicted, but the spatial distribution is also modelled. As a result, the calculations required become more involved, and it is usually necessary to use specialised software, such as a geographical information system, to implement the model. Examples of such models include a model to predict the spread of screwworm fly in Australia¹, or the spread of foot and mouth disease².

The use of either computer spreadsheets or programming languages for model implementation tends to result in a mixing of the two conceptual components of the implementation of a model: the mathematical equations, and the data management mechanisms surrounding them. This means that the mathematical structure of a

model is usually ‘hard coded’ within the system. Many implementations allow relatively simple modification of a variety of parameters or even more extensive modifications (eg BIOCLIM³), but changing the structure and relationships of the underlying mathematical equations requires advanced programming skills, and can usually only be achieved by the person or team who developed the model in the first place. This paper describes an approach developed to overcome these limitations through the separation of the two parts of a model. This was achieved through the use of a ‘software socket’ which embodies the data management and analysis components of a model, and into which can be plugged the mathematical description of the model.

The system was designed to assist with the implementation of a predictive tempero-spatial distribution model to forecast the risk of bluetongue virus activity in Australia. A number of sources of data were used for model development including data from the Australian National Arbovirus Monitoring Programme (seroconversion data from sentinel herds as well as insect vector trapping data), meteorological observations, livestock distributions, vegetation cover and topography. System development was achieved on a single, stand-alone IBM compatible personal computer. Data sources were combined within ArcView GIS version 3.1⁴ using the Spatial Analyst extension for managing raster-based data. The bulk of the system was programmed using the Avenue programming language which forms a part of ArcView, although some specialised routines were developed in Delphi version 4⁵ and linked to ArcView in the form of DLLs.

The system developed consists of a range of generic and specialised data management, analysis and display tools suitable for the modelling of biological processes, which operate independently of the mathematical definition of a model. These tools are called as required by the software socket. The socket itself is a parsing engine which accepts as input a text file containing a simple definition of the data sources, mathematical equations and relationships which make up the model itself. The format of the input text file containing the model definition can be thought of as a simple high-level programming language designed specifically for tempero-spatial modelling of biological systems. The relationships between components are shown in Figure 1.

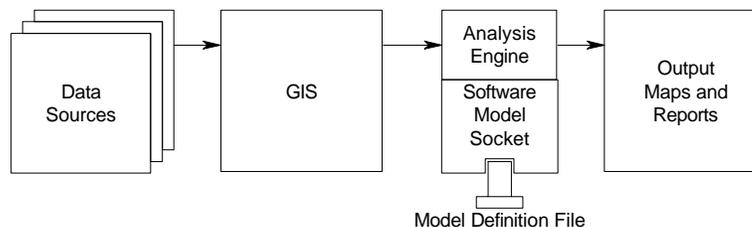


Figure 1: Conceptual relationships within the modelling system.

Each tool is called in response to a corresponding command block in the model definition. A command block consists of the command followed by a number of parameters. Implemented commands include:

- *Define*: Defines a spatial grid data source for later use.

- *Display*: Displays a defined or calculated grid.
- *Summarise*: Calculates any of a number of summary statistics for a temporal sequence of spatial grids, using a variable window and lag. For instance, Summarise may be used to with daily rainfall grids to calculate the average rainfall over a 2 week period, lagged 5 days.
- *Logistic*: Implements a spatial logistic regression model to produce an output grid representing the probability of the outcome. The model can accept any number of input data sources and coefficients.
- *Calc*: performs generic grid-based calculations.
- Other specialised commands such as *Interpolate*, *SQL*, and *Degree Days*.

An example of a simple model definition file is shown in Figure 2. This model calculates the average minimum temperature over the preceding 10 days, defines a grid showing annual rainfall, calculates low rainfall areas, calculates a logistic regression model, and displays the results.

[summarise]	src=c:\data\annualrain	x1=desert
sumtype=average	[calc]	b2=0.1827592
srcdir=c:\data\mintemp\	desert=anrain<=400	x2=avmint
window=10	[logistic]	name=resultsGrid
lag=0	params=3	[display]
name=avmint	b0=-4.590668	grid=resultsGrid
[define]	b1=1.6084073	name=Model Results
name=anrain		

Figure 2: Simple model definition file for use with the software socket

Discussion

While some models allow for the adjustment of certain parameters, this system allows for all data sources, parameters, relationships and structures within the model to be completely redefined without any need for reprogramming.

The use of a generic spatial modelling engine with a software socket simplifies the process of model development and evaluation. Alternative models can be quickly and easily compared by loading different model descriptions into the modelling engine. Sub-components of the model can be developed and tested independently, and then combined into a single text file. The same modelling engine can be used for multiple different spatial models each defined by a small text file.

References

- ¹ Mayer DG, Atzeni MG, Butler DG, Anaman KA, Glanville RJ, Stuart MA, Walthall JC, Douglas IC (1994): Biological simulation of a screwworm fly invasion of Australia. Project Report Series Q094005, Department of Primary Industries, Queensland. 56pp.
- ² Sanson RL, Morris RS, Stern, MW (1993): EpiMAN: a spatial decision support system for use in an exotic disease emergency. Ag. Sys. Info.Tech. 5:20-22.
- ³ Busby. JR (1986) Bioclimatic Prediction System (BIOCLIM) User's Manual Version 2.0 Australian Biological Resources Study Leaflet.
- ⁴ ESRI (Environmental Systems Research Institute), 1998. ArcView GIS 3.1 (software program).
- ⁵ Inprise Corporation 1998. Borland Delphi Professional version 4 (software program)