

Dynamic Modelling and Visualization on the Internet

Bo Huang* and Michael F. Worboys

Department of Computer Science
Keele University
U.K.

Abstract

There is a growing requirement for GIS to incorporate dynamic analytic models. At the same time, there is a need to distribute results of dynamic GIS using the Internet. Therefore, this paper sets out to explore the implementation of dynamic environmental models using Internet-based geocomputation techniques. An overview discusses shortcomings of current Internet GIS techniques for dynamic modelling based on the idea that bidirectional and sustained communication is required between the client and the server sides. Thus an applet-servlet approach is explored to demonstrate the modelling process of a chosen hydrological model, TOPMODEL, which requires frequent and efficient client-server interactions. This approach overcomes the inherent shortcomings of the current Common Gateway Interface (CGI) and more primitive Java applet techniques. We present an effective and generic way to implement dynamic modelling and visualization processes in an Internet environment. This allows users to benefit from Internet-based geocomputation techniques to gain insights into computation and representation of dynamic spatial phenomena.

Key words: Dynamic modelling, visualization, TOPMODEL, Internet, applet-servlet

* Address for correspondence: Bo Huang, Department of Computer Science, Keele University, Keele, Staffs, UK ST5 5BG. E-mail: bohuang@cs.keele.ac.uk

1 Introduction

Most current GIS systems lack sophisticated spatial analytical and modelling capabilities. As an alternative, attempts have been made to integrate various environmental models into GIS in different ways (see for example Goodchild et al 1993; Fotheringham and Rogerson 1994; Fischer et al 1996; Longley and Batty 1996). Various spatial statistical packages have also been developed as add-ons tied to GIS or as stand-alone packages with basic visualization functionality in support of exploratory data analysis. It is envisaged that the next challenge in the coming years will be the transformation of current GIS from static systems to dynamic modelling systems, to help provide more insights into space-time processes (Burrough 1998).

Environmental modelling is made intricate by involving both space and time. Visualization, although not absolutely necessary for such modelling, has been considered an important means to support the modelling process, as it can help users improve the understanding and may even provide insights that would otherwise be missed. To facilitate the development of model code and the visualization of results, a stand-alone package called PCRaster (Deursen 1995; Wesseling et al 1996) has been developed, providing a powerful and flexible modelling language for implementing dynamic environmental models. This frees users from complex low-level programming using Fortran or C/C++. Although the data for, and the results of modelling can be easily exchanged with standard GIS, no current GIS includes dynamic modelling as standard functionality.

Various efforts have been made in facilitating the development of dynamic modelling software, but there are still further difficulties such as easy user access, platform independence and GIS invariance. Thus the use of Internet to overcome these obstacles is suggested (Openshaw et al 1999). Internet technology has developed rapidly in recent years, resulting in the migration of some GIS functionalities to the WWW. The explosive development of the Internet also provides an opportunity to allow easy and wider access

to an environmental model as well as data, and to create platform-independent modelling software.

This paper focuses upon the use of Internet for dynamic modelling and visualization. The placing of dynamic models on the Internet has various positive implications. Through the Internet, users can easily access the model, test the model and finally make a decision whether or not to buy this model or use this model in their particular settings. On the other hand, for software developers, the traditional problems of porting software onto different hardware and for different operating systems may be avoided. While it is a good idea to make environmental models operational across the Internet, the task is non-trivial as there are several difficulties as follows:

- Allocation of different tasks such as modelling and result visualization to the client or server side so as to achieve a balance between both sides.
- Bidirectional and sustained communication between the client and server sides at each time step during the modelling and visualization process. That is, after a modelling result is achieved, the client side then starts to display the result. Only after the display is done on the client side can the server side start the calculations for the next step.
- Update of existing displays (e.g. maps and charts) at each time step instead of data reloading and redrawing.
- Synchronization between different windows (e.g. a map window and a chart window) at each time step for visualization of outputs in the modelling process.

Conventional Internet-based techniques have drawbacks in dealing with the above problems. Thus a Java-based applet-servlet approach, which is able to overcome the barriers, is adopted.

In this study, we take TOPMODEL (TOPography based hydrological MODEL) as an example to illustrate how an environmental model is mounted on the Internet. The reasons for choosing this model are that it has been widely used in hydrological applications, and can easily link to a GIS as this model relies on Digital Elevation Model (DEM) data.

This paper details how TOPMODEL is prototyped on the Internet. Section 2 discusses the shortcomings of current Internet GIS techniques in terms of basic requirements on dynamic modelling and visualization. Section 3 describes TOPMODEL concepts and its current practices without involvement of Internet. In Section 4, the simulation of TOPMODEL on the Internet using the applet-servlet approach is presented. Finally Section 5 draws the conclusion.

2 Current Internet GIS Techniques: an Overview

As the Internet has become an important means for acquiring and disseminating information, a number of Internet GIS products have been brought to market. Underlying these products, there are basically two kinds of applications: server-side and client-side. The former depends on the GIS server to perform all processing tasks and sends back the results to the front-end upon clients' request, while the latter allow the data to be transferred to the local client side and executes all processing at the front-end (Plewe 1997; Peng 1999). Common Gateway Interface (CGI) and Java applet approaches are respectively typical examples of each application.

CGI was one of the first techniques for creating dynamic content in web pages. With CGI, a web server passes certain requests to an external program. The output of this program is then sent to the client in standard HTML formats that can be viewed in any web browser. The advent of CGI methods made it possible to implement most new functionality in web pages, and CGI quickly became a de facto standard, implemented on dozens of web servers (Huang and Lin 1999). Despite the simplicity and universality of CGI scripts, the CGI approach suffers from drawbacks of limited user interface and poor performance as the server needs to create a process for any request. The result is that it is not suitable for dynamic modelling and visualization because at each time step, existing displays may need to be updated. To send back an image using CGI methods at each of a number of steps is time consuming and will cause an over-loaded server. Instead it is better to send back in a proper format only the data that change.

In contrast to the CGI approach, the Java applet has been considered as a very flexible client-side application which augments a web browser. In general, the strengths of Java applets are the abilities to enhance user interface and improve performance. But an environmental model like TOPMODEL requires complicated computations for modelling and visualization. If all these are accomplished by an applet, it will result in a very fat client side, which, except for taking time for downloading the software and data, may finally result in the paralysis of the client machine. Since both modelling and visualization processes are required, we advocate a comparatively balanced client-server application, in which the client side is dedicated to visualization, while the server side is responsible for complicated modelling computation.

Given that CGI and merely Java applet approaches are not suitable, let us look at the Java servlet techniques.

Recent years have witnessed a strong rise of server-side Java applications, which is one of the latest and most exciting trends in Java programming (Hunter and Crawford 1998). Java servlets are a key component of such a server-side Java development.

A *servlet* is a generic server extension - a Java class that can be loaded dynamically to expand the functionality of a web server. It can be thought of as a server-side applet. That is, a servlet extends the capabilities of a server in the way that an applet extends the capabilities of a browser. A Java servlet runs inside a Java virtual machine (JVM) on the server.

Unlike CGI, which use multiple processes to handle separate programs and/or separate requests, servlets are all handled by separate threads within the web server process. This gives servlets tremendous performance benefits over CGI programs. Also because they are written in Java, servlets have full access to the various Java APIs and to third-party component classes, making them ideal for use in communicating with applets, databases,

and Remote Method Invocation (RMI) servers. Plus, servlets are portable between operating systems and between servers.

As a servlet allows flexible communication between a server program and a client program e.g. a Java applet, this lends itself to dynamic modelling and visualization because frequent bidirectional communication is required in the process of modelling and visualization. In other words, a coordination between the client and the server sides at each time step during the whole process is needed.

The applet and the servlet can communicate in several ways such as HTTP text, HTTP object, socket and RMI. In our case we choose HTTP text because it is simple and suits our purpose, and can be immune from firewall blockage and browser security checks. Of course, other communication methods can also be applicable. For example, a socket connection can be established with a data stream probably in compressed binary format delivered between the applet and the servlet. However, the decoding of the data stream may take time. Moreover, the socket connection may fail for applets running behind firewalls. For more details about the pros and cons of each communication method, see (Hunter and Crawford 1998).

3 TOPMODEL Concepts

TOPMODEL (Beven and Kirby 1979) is a physically based watershed model that simulates the variable-source-area concept of streamflow generation. This model requires Digital Elevation Model (DEM) and a sequence of rainfall and potential evapotranspiration data, and it simulates the resulting stream discharges. TOPMODEL has achieved some popularity as it provides computationally efficient prediction of distributed hydrological responses with a relatively simple framework for the use of DEM.

Since the theoretical basis of TOPMODEL has been clearly reported in (Beven and Kirby 1979; Beven and Wood 1983; Beven 1995), we here provide only a brief description of

the model. TOPMODEL makes use of a topographic index of hydrological similarity based on an analysis of the topographic data (Quinn et al 1997), which can be described as follows:

$$\ln(\alpha / \tan \beta)$$

where α is the cumulative area drained through a unit length of contour line, and $\tan \beta$ is the slope.

In general TOPMODEL can be used to reproduce the hydrological behavior of catchments in a distributed or semi-distributed way, in particular it maintains a continued accounting of the storage deficits allowing the identification of the saturated source areas within the catchment. Under the assumption of an effective water table gradient and saturated flow parallel to the local surface slope $\tan \beta$, then at any point I on a hillslope, the downslope saturated subsurface flow rate q_i per unit contour length (m^2/h) is described by the equation:

$$q_i = T_0 \tan \beta_i e^{-\frac{S_i}{m}} \quad (1)$$

where T_0 is the average soil transmissivity (A homogeneous soil through out the catchment is assumed), β_i is the slope angle, S_i is the local storage deficit, and m describes the change in transmissivity with depth. Based on this exponential approximation, the local deficit is derived as:

$$S_i = \bar{S} + m[\Lambda - \ln(\alpha / \tan \beta)_i] \quad (2)$$

where \bar{S} is the average storage deficit, Λ is the areal average of $\ln(\alpha/\tan \beta)$. Equation (2) is used to predict the saturated contributing areas at each time step. The negative value of S_i indicates that the area is saturated and the saturated overland flow is generated, while the positive value of S_i indicates the area is unsaturated.

Other equations regarding unsaturated zone calculations, vertical flow, evapotranspiration, and outflow from the saturated zone can be found in (Beven et al 1995).

Over the past two decades, TOPMODEL concepts have been implemented with several computer languages like Fortran and Visual Basic upon different computer platforms (see for example Beven 1997). The Visual Basic version of TOPMODEL in the Windows environment allows users to view outputs of the model and interact with the model. These developed tools have been widely used in the application of hydrological modelling in numerous catchments in the world (Beven et al 1995). Meanwhile, attempts have been made to improve the concepts, such as uncertainty evaluation when using the TOPMODEL model for hydrological modelling.

Efforts have also been made using GIS for implementing TOPMODEL. Chairat and Delleur (1993) used GRASS for hydrological parameter determination, hydrological assessments and linked it to TOPMODEL. Another paper from Stuart and Stocks (1993) focused on the integration of a set of generic modelling tools into the SPANS GIS. These tools are then employed for building TOPMODEL. While TOPMODEL has been implemented in different ways, its prototype on the Internet has so far rarely been done especially with new Internet techniques such as applet-servlet. The following reports on our exploration in this regard.

4 Implementation of TOPMODEL on the Internet

The implementation of TOPMODEL mainly consists of two stages (Figure 1). The first stage is to derive α and $\tan \beta$ from the elevation map of the watershed, which results in a topographic index map. The second stage integrates all hydrological parameters and rainfall records for simulation purposes. The topographic index map at the first stage is derived by ArcView. We read the topographic index map data using Java and then move to the simulation process at the second stage.

The modelling process at the second stage are made for areal subdivisions based on the $\ln(\alpha/\tan \beta)$ subdivisions. The saturation deficit for each subdivision is calculated from the mean subcatchment deficit at the start of each time step.

[Figure 1. A generic implementation process of TOPMODEL]

Each increment also has a root zone storage (SRZ) and an unsaturated zone storage (SUZ). SUZ has an upper limit of the local saturation deficit SD. The local contributing area is where SD-SUZ is less than or equal to zero. Finally, channel routing calculations are done. These steps are realized by Java programming inside a servlet at the server side.

The output of the TOPMODEL includes different graphical displays such as a hydrograph showing the curve of predicted streamflow, animation of soil saturation status at each time step, and other maps. A user is encouraged to view this output and think about how the model is working. These are implemented by a Java applet at the client side.

The architecture for the implementation is shown in Figure 2. We here use the datasets of the Slapton Wood catchment to illustrate the implementation primarily due to their availability at the TOPMODEL website (<http://www.es.lancs.ac.uk/hfdg/topmodel.html>).

[Figure 2. Architecture of the implementation]

The Java applet first displays the topographic index map in the web browser (Figure 3). Then the simulation process can start after selecting the 'simulation' button. In this process, the Java applet sends the time step number to the servlet, which receives it and generates the saturated contributing area represented by a string of grid numbers and the predicted streamflow at this time step. These are then sent back to the applet, which updates the topographic index map with the new saturated contributing area in the red color and draws the curve of predicted streamflow at another window. Note that it is only the newly saturated contributing area data, but not all the image data that are sent to the Java applet for visualization. This significantly reduces the network traffic. The following Java source code reflects this:

```
public void doGet (HttpServletRequest req, HttpServletResponse res)
```

```

throws ServletException, IOException
{
    String value;
    Enumeration values = req.getParameterNames();
    while(values.hasMoreElements()) {
        String name = (String)values.nextElement();
        value = req.getParameterValues(name)[0]; //value of timestep
    }
    //handshake at two sides with timestep value
    if (Integer.parseInt(value.trim())== timestep){
        res.setContentType("text/plain");
        //send the topographic index subdivision numbers represented
        //by a string to the Java applet via a text-output stream
        PrintWriter out = res.getWriter();
        out.println(getSubdivisions()); //topographic index subdivisions
        timestep++;
    }
}

```

The doGet method above has two interesting parameters: HttpServletRequest and HttpServletResponse. These two objects give full access to all information about the request and let the developer control the output sent to the client as the response to the request. In this case the data of newly saturated contributing area are streamed to the Java applet.

[Figure 3. Dynamic visualization of saturated contributing areas and predicted streamflow against observed data (URL: 160.5.90.208/grid/topmodel.html)]

In contrast to the background information of observed streamflow and rainfall record, the difference between the predicted flow and the observed data can be easily distinguished.

The prediction of streamflow involves a series of calculations such as local storage deficit, root zone deficit, unsaturated zone storage, and saturation excess. Since the saturated contributing areas play an important role in the final generation of streamflow, they are displayed at each time step. The saturated cells in this case are closely related to the areas with higher topographic index values, which are close to the channel in the catchment. For presentation purpose, only the time steps 300-368 of the whole 950 time steps are displayed at the right window of Figure 3. The displays at the left window and the right window are synchronized by using threads in Java programming.

The saturation status map is updated at each time step, and this effect is visualized through animation. This, in conjunction with dynamic streamflow drawing, assists users in understanding the catchment responses to rainfall with respect to terrain features. Such functionality is made possible by the fact that the results can be mapped back into space and viewed by the user in their spatial context. This satisfies the objective of TOPMODEL development as the model is always suggested as an aid to understanding before it is used as a predictive tool (Beven 1997).

The implementation of TOPMODEL developed here illustrates how it can give some insight into the spatial patterns of hydrological responses in a visual way, both in terms of presenting hydrographs and mapping the predictions of saturated contributing area in the catchment under study based on the distribution of the topographic index.

5 Conclusions

This paper has demonstrated how to implement an environmental model and its visualization process in an Internet environment using the applet-servlet approach. This approach overcomes drawbacks of the conventional Internet GIS techniques approach by providing an effective and efficient mechanism for the two-way and sustained communication between the visualization and the modelling processes so that dynamic nature of environmental modelling is successfully reflected. Only the very relevant data

in response to clients' request are sent back to the Java applet greatly reducing network traffic and augmenting the performance.

This implementation of TOPMODEL in this paper emphasizes the way in which users can explore the model's data and predictions across space and through time. The simultaneous dynamic visualization of the observed and simulated phenomena is a help to both modelers and users. From the perspective of modelers, it helps to assess the rhythms of convergence or divergence between simulated and observed dynamics, and to identify the critical zones where further investigations should be undertaken to improve the model. From the perspective of users, it can improve understanding of relationships or variables inside the model. The use of Internet technology makes it easy to gain the above benefits.

The applet-servlet approach is not only applicable to highly interactive problems as presented in this paper but also to any other problems that CGI scripts could tackle. This indicates that more simulations like distribution of saturation deficit can be done to enhance the implementation.

The implementation of TOPMODEL can also include interactive parameter calibration and sensitivity analysis, as in (Beven 1997). We hope that the implementation described in this paper provides a basis for further implementations of TOPMODEL on the Internet as well as other environmental model simulations to facilitate collaborative spatial decision making.

Acknowledgements

The authors would like to thank Prof. K. J. Beven and his research group at University of Lancaster, UK, for providing various reports, Fortran source code and sample data at their TOPMODEL website. Comments of anonymous reviewers were helpful in improving this paper.

References

- Beven K J, Kirby M J 1979 A physically based variable contributing area model of basin hydrology *Hydrol. Sci. Bull.* **24**: 43-69.
- Beven K J, Wood E F 1983 Catchment geomorphology and the dynamics of runoff contributing areas *J. Hydrol.* **65**: 139-158.
- Beven K J, Lamb R, Quinn P, Romanowicz R, Freer J 1995 TOPMODEL. In Singh V P (ed) *Computer Models of Watershed Hydrology*, Chapter 18, Water Resource Publications: 627-668.
- Beven K J 1997 TOPMODEL user notes (Windows version). Centre for Research on Environmental Systems and Statistics, Lancaster University, UK.
- Burrough P A 1998 Dynamic modelling and geocomputation. In Longley P, Brooks S, McDonnell R, and MacMillan B (eds) *Geocomputation: a primer*. John Wiley & Sons: 165-191.
- Chairat S, Delleur J W 1993 Integrating a physically based hydrological model with GRASS. In Kovar K, Nachtnebel H P (eds) *HydroGIS 93: Application of Geographical Information Systems in Hydrology and Water Resources*, IAHS Publ. No. 211: 143-150.
- Deursen W P A 1995 *Geographical Information Systems and Dynamic Models*. Netherlands Geographical Studies 190, Faculty of Geographical Sciences, Utrecht University, the Netherlands.
- Fischer M, Scholten H J, and Unwin D (eds) 1996 *Spatial Analytical Perspectives on GIS*. London, Taylor & Francis.
- Fotheringham S, Rogerson P (eds) 1994 *Spatial Analysis and GIS*. London, Taylor & Francis.
- Goodchild M F, Parks B O, and Steyaert L T (eds) 1993 *Environmental Modelling with GIS*. New York, Oxford University Press.
- Huang B, and Lin H 1999 GeoVR: a web-based tool for virtual reality presentation from 2D GIS data *Computers And Geosciences* **25**: 1167-1175.
- Hunter J, and Crawford W 1998 *Java Servlet Programming*. Sebastopol, CA, O'Reilly & Associates, Inc.

- Longley P, and Batty M (eds) 1996 *Spatial Analysis: Modeling in a GIS Environment*. Cambridge, Geoinformation International.
- Openshaw S, Turton I, MacGill J, and Davy J 1999 Putting the Geographical Analysis Machine on the Internet. In Gittings B (ed) *Innovations in GIS 6*. Taylor & Francis.
- Peng Z 1999 An assessment framework for the development of Internet GIS *Environment and Planning B: Planning and Design* **26**: 117-132.
- Plewe B 1997 *GIS-Online: Information Retrieval, Mapping, and the Internet*. Santa Fe, USA, OnWord Press.
- Quinn P F, Beven K J, and Lamb R 1997 The $\ln(\alpha/\tan \beta)$ index: how to calculate it and how to use it within the TOPMODEL framework. In Beven K J (ed) *Distributed Hydrological Modelling: Applications of the TOPMODEL Concept*. John Wiley & Sons Ltd, UK.
- Stuart N, and Stocks C 1993 Hydrological modelling within GIS: an integrated approach. In *HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources*, IAHS Publ. No. 211: 319-329.
- Wesseling C G, Karssenberg D -J, Deursen W P A, and Burrough P A 1996 Integrating dynamic environmental models in GIS: the development of a Dynamic Modelling language *Transactions in GIS* **1**: 40-48.

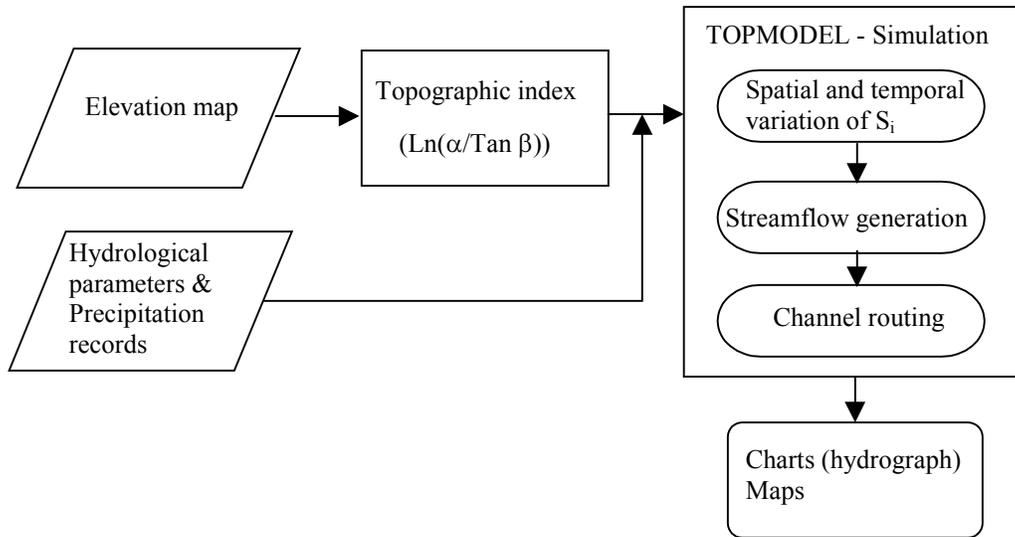


Figure 1. A generic implementation process of TOPMODEL

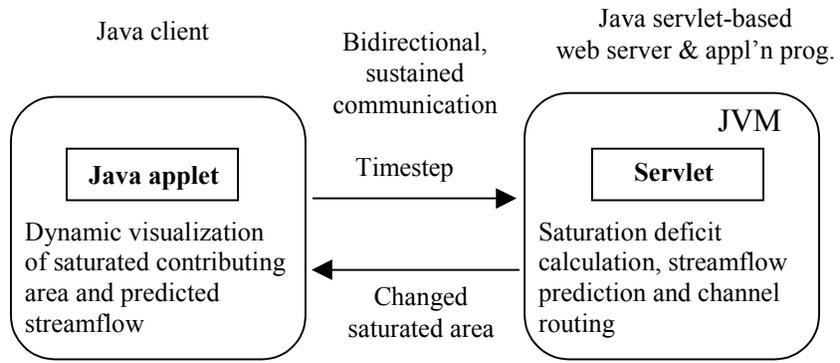


Figure 2. Architecture of the implementation

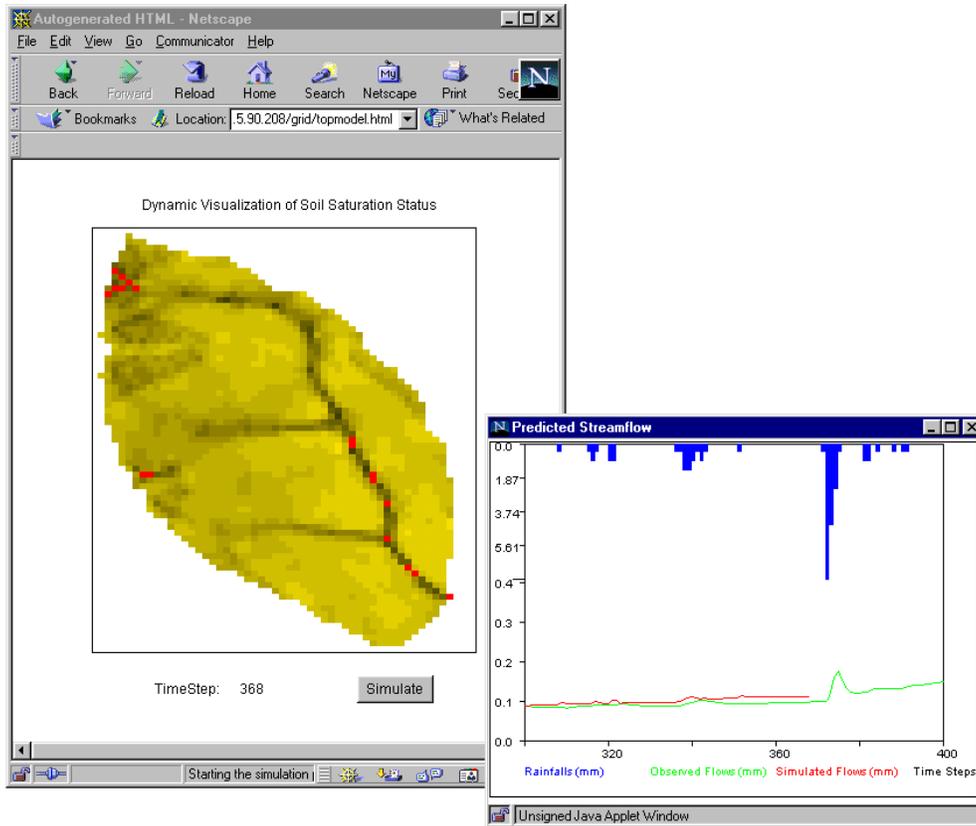


Figure 3. Dynamic visualization of saturated contributing areas and predicted streamflow against observed data (URL: 160.5.90.208/grid/topmodel.html)