

Web-based educational use of hydraulic software in FORTRAN

Apostolos Syropoulos

E-mail: apostolo@obelix.ee.duth.gr

Angelos L. Protopapas

*Associate Professor, Civil Engineering, Democritus University of Thrace
Vas. Sofias 12 – Xanthi 67100 – Greece*

E-mail: aproto@civil.duth.gr

Abstract

This paper reports on the development of educational software to facilitate on-demand availability of traditional hydraulic programs written in FORTRAN (or any other high-level computer language) through the Web. Students may use such tools to learn, practice and consolidate knowledge inside and outside classrooms, even from remote places. This software has been successfully class-tested within an undergraduate civil engineering course and students reported a positive experience with its use. Methods are also proposed to extend the application so that it will be able to return XML and/or SVG content.

Keywords: hydraulic engineering software, web-based education, CGI scripts, XML

1. Introduction

Since the era of mainframe scientific computing in the 1970's significant computer programs have been developed in FORTRAN for the solution of environmental problems, such as water flow and pollutant transport, simulation of biochemical and hydrothermal processes, estimation of hydrologic parameters etc. These enormous volumes of computer code have been the intellectual product of hundreds or even thousands of engineers and scientists and, despite the technological advances and the efforts made, have not always been integrated into educational curricula or disseminated to the broader professional practice.

The transition from the mainframe to the personal desktop computer and then to workstations and portable computers was overwhelmed in the last decade by the emergence of the World Wide Web and the new possibilities it can offer. The application of client-server architecture has led to computer assisted learning facilities, which may complement human teaching in environmental studies.

The goal of this on-going project is to develop educational software to facilitate on-demand availability of traditional hydraulic programs written in FORTRAN (or any other high-level computer language) through the Web. This way students may use these tools to learn, practice and consolidate knowledge inside and outside classrooms, even from remote places.

Benefits of Web-based education include independence of teaching and learning with respect to time and space [1]. Another important advantage of Web-based educational software is platform independence of the application and easy access to it. These advantages are particularly appropriate for water resources studies since they allow distributing the expertise of the instructors involved in the development of a Web-based computer assisted learning environment.

A rather different approach to Web-based educational applications is the so-called Intelligent Tutoring Systems (ITS) which were originally developed as stand-alone, student-adapted educational applications. However, lately, following the rapid growth of Web-based technologies the ITSs have been transformed so that they can be used over the Network.

However, a serious drawback of ITSs is that they are designed from scratch with cooperation of domain experts with knowledge engineers in many iterative cycles. This is why they require a huge amount of man-hours to develop. By some estimates it takes more than 200 man-hours to produce an hour of instructional material for an ITS [2].

2. Motivation for Web-based Application

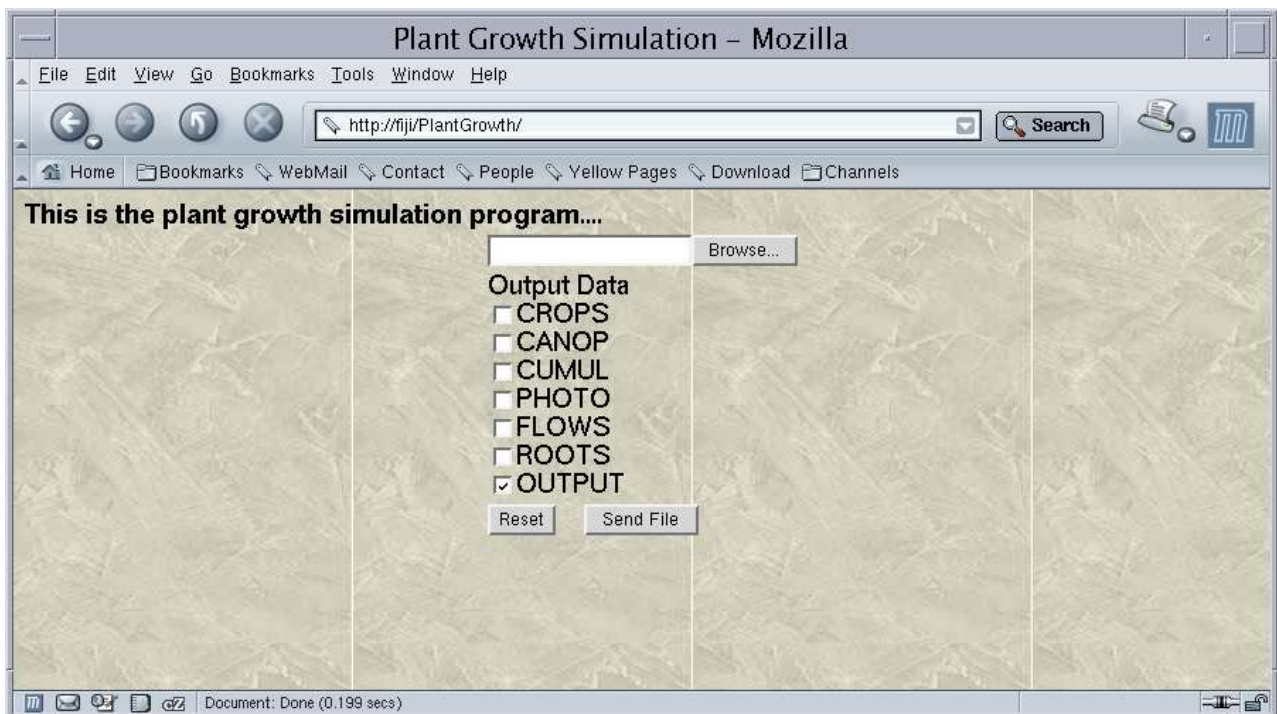
An assignment for a required course at Democritus University of Thrace, Greece, titled "Land Reclamation and Irrigation Engineering" involves the estimation of crop water needs. Rather than using typical empirical formulas, it was decided to familiarize the students with a biohydrological model of plant growth. The model simulates the underlying physiological, chemical and physical processes, deals with arbitrary climatic inputs, and takes into account the transport of moisture and solute in a soil column [3]. The computer program is called HYNEW and is written in FORTRAN. A detailed description of the operational aspects of the program can be found in the program manual [4].

This emerged as one of many situations where computer programs developed on no longer widely available computer platforms are still considered to be valuable educational material. Although it is not difficult to port these programs in a particular computer platform, it is not practical to do this in many different common platforms. For example we have ported one such program to Solaris 8 x86 but it would not be trivial to port it to Windows, Linux, etc. The option to provide students with the Solaris binary was not feasible because each student may use one of the many different operating systems on campus. This situation motivated the development of a Web-based application.

3. Description of the Application

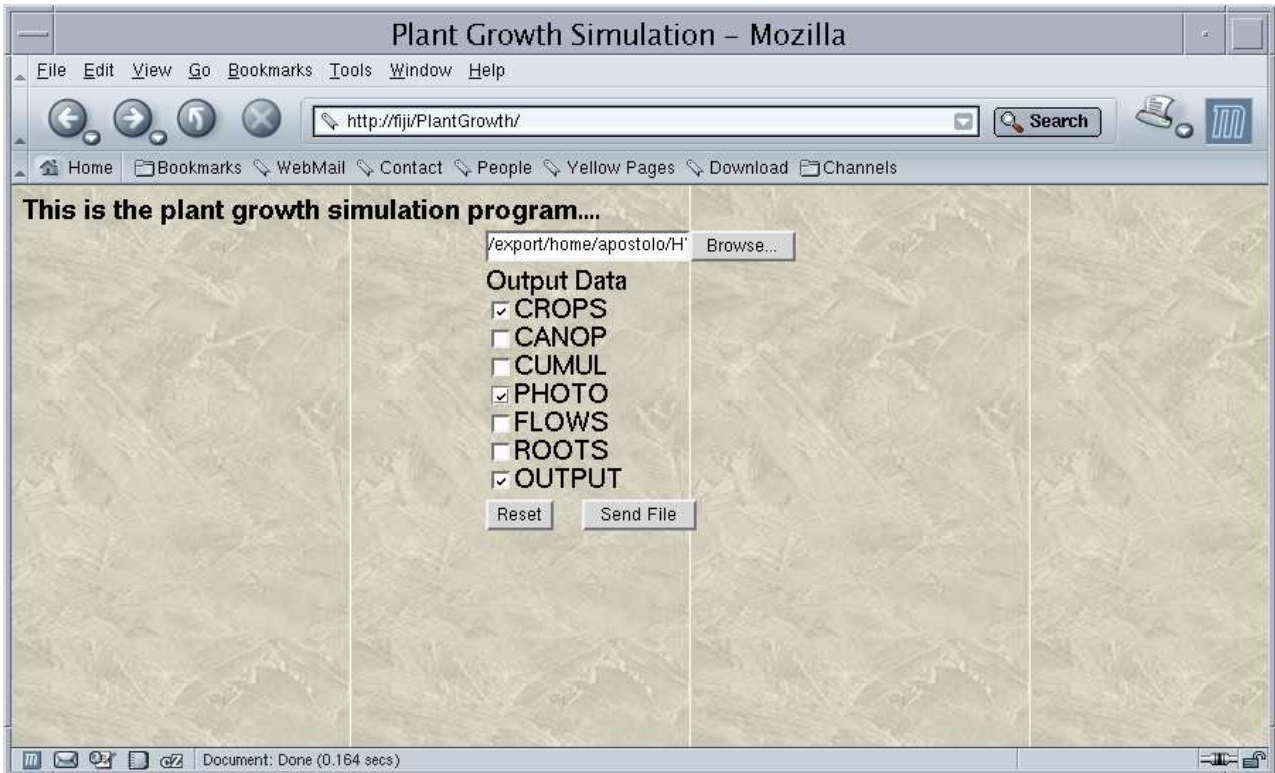
Currently, the user interface of our application is simply an HTML page that can be used to invoke a CGI script. A Common Gateway Interface (CGI) script is a program that is designed to accept and return data in a standardized way (CGI specification). The CGI is a specification for transferring information between a World Wide Web server and a CGI program. Hence our CGI script is used to upload the contents of an input file, which contains data that are “comprehensible” by the FORTRAN program, and returns data that this program produces. In Figure 1 a preliminary version of the Web page is shown.

Figure 1. Preliminary version of the Web page



By pressing the “Browse” button, one is able to choose a data input file (that resides on local storage media, e.g. a hard disk) the contents of which will be transferred to the Web Server. The user also selects the output data that he/she wishes to receive by ticking the appropriate boxes. In Figure 2 the same Web page is shown after certain selections are made.

Figure 2. Web page with certain selections

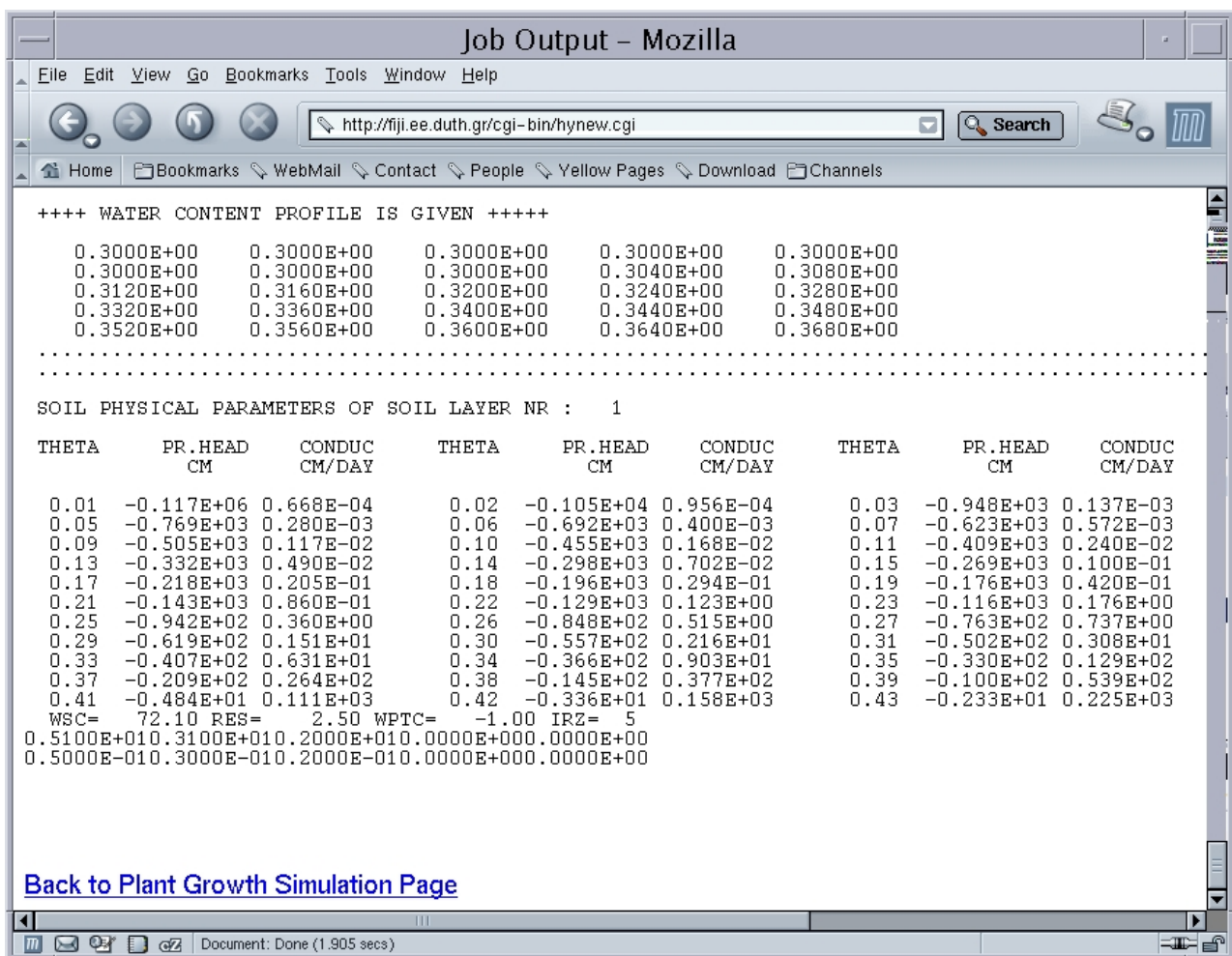


At this point the “Send File” button should be pressed so that the data are delivered to the Web Server. In Figure 3 a glimpse of the output data is presented. The output can be saved as a plain text file and used to print or produce graphs and tables with widely available applications. At the moment there are plans to extend the application so that it will be able to return XML and/or SVG content.

XML (the eXtensible Markup Language) is W3C-endorsed standard for document markup. It defines a generic syntax used to mark up data with simple, human readable tags [5]. Therefore, XMLised data can be transferred through the Web without any additional effort. However, this implies that the CGI script should be able to XMLise the output of the FORTRAN program. Though this seems to be a difficult task, it can be easily accomplished once the proper Document Type Definition (or DTD, for short) is created. A DTD lists all elements, attributes and entities a document should use and specifies the context in which they should be used. The DTD is essential for the validation of the resulting XML file. This means that users have at their disposal a tool that can be used to check the integrity of the data file. One of the great advantages of using XMLised data is that one can “display” them in various ways. For example, it is possible to generate a PDF file (i.e., to some printer-friendly format) or some other file. This can be achieved by using a XSLT transformation. The eXtensible Stylesheet Language Transformations is a W3C standard for transforming XML documents into other XML documents or other formats [6]. This was conceived as part of XSL but has been found to have wider applications. Although it is possible to transform our output data to SVG, we believe it is better to directly generate SVG content in order to have complete control of certain aspects of the generation process, such as the size of the final file, etc.

SVG is a W3C standard for vector graphics, based on XML [7]. The term vector graphics refers to the representation of separate shapes such as lines, polygons and text, and groups of such objects, as opposed to bitmaps. The main advantage of vector graphics formats (e.g., SVG, PostScript, etc.) over bitmap graphics formats (e.g., JPEG, GIF, etc.) is that images can be rendered at different sizes and be transformed in various ways without worrying about image resolution and pixels. This is quite important, as users may not have at their disposal high-resolution printing equipment. In addition, it permits the examination of certain details on the screen without any problem. Our CGI script will be able to generate SVG data by adding more options to the initial screen. For example, one may want to have a plot of the soil physical parameters of a certain soil layer. In addition, we can try to generate images depicting the physical environment, although this can be achieved easier in a VRML world generated by a XSLT transformation.

Figure 3. Sample output data



4. Implementation Details

The Web-based application has been implemented on an Intel P4 machine running Solaris 8 x86. Currently, the system is equipped with IDE hard disks, which are quite adequate for our current limited use (a class of about 20 students). However, when the application will be available in a “productive” environment, the system will be equipped with SCSI disks to allow maximum performance while minimizing the possibility of a disk crash.

We have opted to use Apache as our Web server mainly because it is open source and because the time between bug-tracking and bug-fixing is very very short. Since Perl is the language of choice for CGI programming, we have opted to use Perl to code the CGI scripts. The CGI script accepts data, which are stored to a local input file submitted by the user. The data contained in the local file are processed by the FORTRAN program, which has been slightly modified so that it can cooperate smoothly with the CGI script. For example, error messages are placed in error-log file. The computed output data are post-processed by the CGI script, which brings them to a human-readable form.

In certain situations, it is possible that users may feed the system with really complicated input data files, which will take quite a long time to be processed. In addition, such input data may generate enormous data, which will make it difficult to transport them through the Internet. For this reason, an upper limit for execution time is set. An over-demanding job will generate the typical error screen with a message "Job cannot be executed for more than x seconds". The current limit was estimated from a number of test executions with various input data and applies only to this particular case. Other programs will definitely have a different time limit. Another limit that is imposed in the current application configuration is the length of input data, which can be up to 20 KB. The limit is not imposed by some Internet traffic restrictions, but mainly due to the fact the processing time is proportional to the input data length.

5. Summary and Conclusions

We have developed educational software to facilitate on-demand availability of a hydraulic engineering program written in FORTRAN through the Web. This software has been successfully class-tested within an undergraduate civil engineering course and students reported a positive experience with its use.

It is expected that this approach can be applied to a variety of traditional computer programs written in out-of-date programming languages, yet solving quite important problems. Instructors could no longer depend on the availability of specific computer equipment or system managers and students could be provided with a transparent way to understand simulation results.

References

- [1] Virvou, M., Educational Software Technology for Environmental Studies, Proc. 6th International Conference "Protection and Restoration of the Environment", 1553-1560, 2002.
- [2] Woolf, B. P., and P.A. Cunningham, Multiple Knowledge Sources in Intelligent Teaching Systems, IEEE Expert Systems, Vol. 2(2), 41-54, 1987.
- [3] Protopapas A. L., and R. L. Bras, State-space dynamic hydrological modeling of soil-crop-climate interactions, *Water Resources Research*, Vol. 24(10), p. 1765-1780, 1988.
- [4] Protopapas A. L., User's Manual for Program HYNEW, 1988, available upon request.
- [5] <http://www.w3c.org/XML/>
- [6] <http://www.w3.org/TR/xslt>
- [7] <http://www.w3.org/TR/SVG/>