

5/15/02

**User Interface on the World Wide Web:
How to Implement a Multi-Level Program Online**

**Jonathan W. Cranford
Mentor: David B. Yeager
Internal Operation Group
Information Systems Division
Communications and Networks Branch**

Abstract

The objective of this Langley Aerospace Research Summer Scholars (LARSS) research project was to write a user interface that utilizes current World Wide Web (WWW) technologies for an existing computer program written in C, entitled LaRCRisk. The project entailed researching data presentation and script execution on the WWW and then writing input/output procedures for the database management portion of LaRCRisk.

Introduction/Background Information

A few years ago, Robert M. Baker wrote a risk assessment program in DBase IV called TeamRisk that evaluated the security of computer systems at NASA Langley Research Center (LaRC) by comparing a branch's implemented protective measures against a database of NASA and LaRC guidelines. Realizing that other NASA Research Centers might benefit by using such a program to assess their computer security, he then wrote CentRisk, a program which allows a center to create its own database of guideline protective measures.

However, because DBase is a DOS-based software package, both CentRisk and TeamRisk, which together are known as LaRCRisk, were limited in use and availability to IBM PC-compatibles. Consequently, last year Mr. Baker asked Jerry W. Park to write a version of LaRCRisk in C so that it would be portable to various computer platforms (e.g., UNIX, DOS, Macintosh, etc.). Considering that most of the larger computer networks at LaRC are UNIX-based, Mr. Park first concentrated on developing a version of LaRCRisk for a UNIX environment, intending to develop versions for other platforms at a later time. Also, since LaRC's guideline database had already been defined, he decided to focus on the TeamRisk module of LaRCRisk before converting the CentRisk module.

In writing the program, however, he was limited to a crude user interface called "curses" because it was the only C language library available on most other computer platforms. Once the program was completed, he realized that different software packages did not have to be developed for other computer platforms; a user on a platform other than UNIX could simply hook up to the UNIX machine running LaRCRisk via a network connection. Nonetheless, Mr. Baker and Mr. Park decided that the most efficient way to use LaRCRisk, due to the awkwardness of the interface, was to have the branch wanting to do a risk assessment fill out worksheets describing the relevant computer systems and other necessary parameters for the program. Mr. Park could then personally enter the data from the worksheets and run LaRCRisk, generating reports to return to the branch requesting the risk assessment. Until a more efficient system could be created, conversion of CentRisk into C was indefinitely postponed.

This LARSS research project was to create such a system, i.e., if possible, to modify LaRCRisk so that it would run off the World Wide Web portion of the Internet. The advantages of putting LaRCRisk "on the Web" are as follows:

1. user-friendliness.
2. LaRCRisk can be stored on a single machine and still allow the user to enter data from a local computer.
3. Many users can access and use the same guidelines database.

The specific objectives of this project were therefore to 1) determine whether putting a multi-level program such as LaRCRisk on the Web was feasible or not, 2) learn how to manipulate the user interface provided by Web browsers (e.g., Netscape, Mosaic, Lynx, etc.), and 3) using Mr. Park's original code, write an appropriate user interface.

Summary

Before a determination could be made concerning the feasibility of putting LaRCRisk on the Web, more information had to be gathered concerning data presentation and script execution. Data presentation is controlled by Hypertext Markup Language (HTML), which consists of tags of different types embedded into the text of a WWW document; these tags control how the document appears on the screen of a Web browser. Included in HTML is the capability of obtaining input from a user online via a "fill-out form"; browsers which support this feature allow the user to submit the form to a WWW server, which processes the data and generates a document to return to the user as output. How the Web server processes that data and what kind of document it returns are controlled by a "script," a type of computer program. Scripts utilize the Common Gateway Interface (CGI) in communicating with the Web server.

After researching HTML and the CGI and downloading some CGI codes from the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign (NCSA), a trial program entitled LaRCRiskII was written to discover whether or not a multi-level C program could be implemented on the Web. Upon the successful execution of LaRCRiskII and the accompanying conclusion that this research project was indeed feasible, Mr. Baker and Mr. Park were consulted as to the most efficient way to implement LaRCRisk. A modular approach was then adopted: as LaRCRisk was composed of several levels, each of which obtained a certain amount of data from the user and processed it, each level of LaRCRisk would become an individual module, with each module being called in succession. These modules would be scripts separate and distinct from each other that would accept input from the user via a form, save appropriate data into files for later use, and generate a document (usually another form) to return to the user. Using this modular approach, input/output procedures were written for each module as it was converted to CGI format.

LaRC equipment and facilities used in this research project include a SunOS workstation, computer time and space on two separate UNIX network servers, office space, and the technical advice of several employees.

The results were excellent; not only does the WWW user interface add user-friendliness, but members of the same risk assessment team for the branch conducting the risk assessment can run LaRCRisk in the comfort of their own offices. With the necessary documentation, LaRCRisk is self-supporting; it can be implemented so that risk assessments are done without the full-time mediation of a contact person (though one will probably still be necessary to provide user support). Furthermore, the Web version of LaRCRisk (which at this time only includes the TeamRisk portion of the program) provides the necessary environment for the future conversion of CentRisk into C, or another CGI language.