BN-SA-1842

Cover Sheet for a Hanford Historical Document Released for Public Availability

Released 1994

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute

Battelle

BN-SA-1842

MODELING SURVIVAL OF JUVENILE SALMON DURING DOWNRIVER MIGRATION IN THE COLUMBIA RIVER ON A MICROCOMPUTER*

R. A. Peloquin and D. H. McKenzie
Battelle, Pacific Northwest Laboratories
Richland, Washington

ABSTRACT

A compartmental model has been implemented on a microcomputer as an aid in the analysis of alternative solutions to a problem. The model, entitled Smolt Survival Simulator, simulates the survival of juvenile salmon during their downstream migration and passage of hydroelectric dams in the Columbia River. The model is designed to function in a workshop environment where resource managers and fisheries biologists can study alternative measures that may potentially increase juvenile anadromous fish survival during downriver migration. The potential application of the model has placed several requirements on the implementing software. It must be available for use in workshop settings. The software must be easy to use with minimal computer knowledge. Scenarios must be created and executed quickly and efficiently. Results must be immediately available. Software design emphasis was placed on the user interface because of these requirements. The discussion focuses on methods used in the development of the SSS software user interface. These methods should reduce user stress and allow thorough and easy parameter modification.

INTRODUCTION

Staff at Battelle's Pacific Northwest Laboratories have developed a model to simulate salmon survival during downriver migration. The model, entitled Smolt Survival Simulator (SSS), represents the mid- and lower-Columbia River along with the associated hydroelectric dam system. The model will aid fisheries biologists and resource management personnel by providing a mechanism for analysis of alternative methods to increase fish survival. The model is designed for a workshop environment where it can be exercised with parameter values arrived at by participants' consensus as well as with individual values and ranges of values. The intended use of the model places a strong emphasis on the interaction between the software and the user. The methodology used to develop that interface is the focus of this paper. The techniques employed are divided into methods that reduce user stress and methods that allow thorough and easy parameter modification. The discussion is broken into five major areas: background, model description, implementation description, methodology of the user interface, and a brief conclusion.

BACKGROUND

The decision, made three generations ago, to harness the power of the Columbia River has provided the Pacific Northwest with a major source of low-cost electricity. When the dams were built, it was assumed that providing adequate upstream adult passage over the dams was sufficient to sustain Pacific salmon runs. All dams below the Grand Coulee on the Columbia River were constructed with fish ladders to allow salmon to migrate upriver. The migrating fish would return to spawning grounds or to hatcheries built to mitigate impacts from dam construction. Fishery statistics from these years of electrical power system development, (the mid-1930's to the mid-1970's), indicate that upriver migration was not the only factor in salmon survival. During that period, the commercial Columbia salmon catch declined by two-thirds.[1]

In 1980, the Pacific Northwest Power Planning and Conservation Act (Northwest Power Act)[2] was signed into law, establishing the Northwest Power Planning Council. One of the principle mandates of the Council was to develop a program "to protect, mitigate, and enhance fish and wildlife, including related spawning grounds and habitat, on the Columbia River and its tributaries."[3] The Council established the Fish and Wildlife Program "to ensure that fish and wildlife resources are accorded co-equal status with other uses in the management and operation of hydroelectric projects in the region."[4]

The Fish and Wildlife program has been focused on the salmon species and Steelhead trout because of their tremendous economic and cultural importance in the Pacific Northwest. Sections of the program address significant stages in the life cycle of these anadromous fish; natural (freshwater) and artificial (hatchery) production, downstream migration as juveniles (smolts), ocean survival, and upstream migration.

One section of the report, downstream migration, requires that mid-Columbia Public Utilities Districts (PUDs) develop and test plans to achieve 90% smolt survival during downriver migration at each dam project.[5] At a hydroelectric dam project, smolts may pass downstream through the spillways or turbines. It is generally thought that the mortality rate associated with passage through the turbines is higher than the mortality rate associated with spillway passage.

Several plans have been recommended to increase smolt survival. One plan is to increase the amount of water through the spillway during spring migration thus decreasing the percentage of fish drawn through the turbines. Another proposal calls for the construction of collection and bypass facilities in the dams. The bypass facilities would divert smolts away from the turbines and through an alternate conduit constructed at the dams. Other plans under study include collecting and transporting smolts around a dam project (short-haul), transporting collected smolts past several dams (long-haul), and various combinations of increased flow over the spillways, bypass facilities, and transportation.

* Work supported by Chelan County PUD, Grant County PUD, and Douglas County PUD under Contract Nos. 231-1204080, 231-1204052, and 231-1204258.

DISCLAIMER

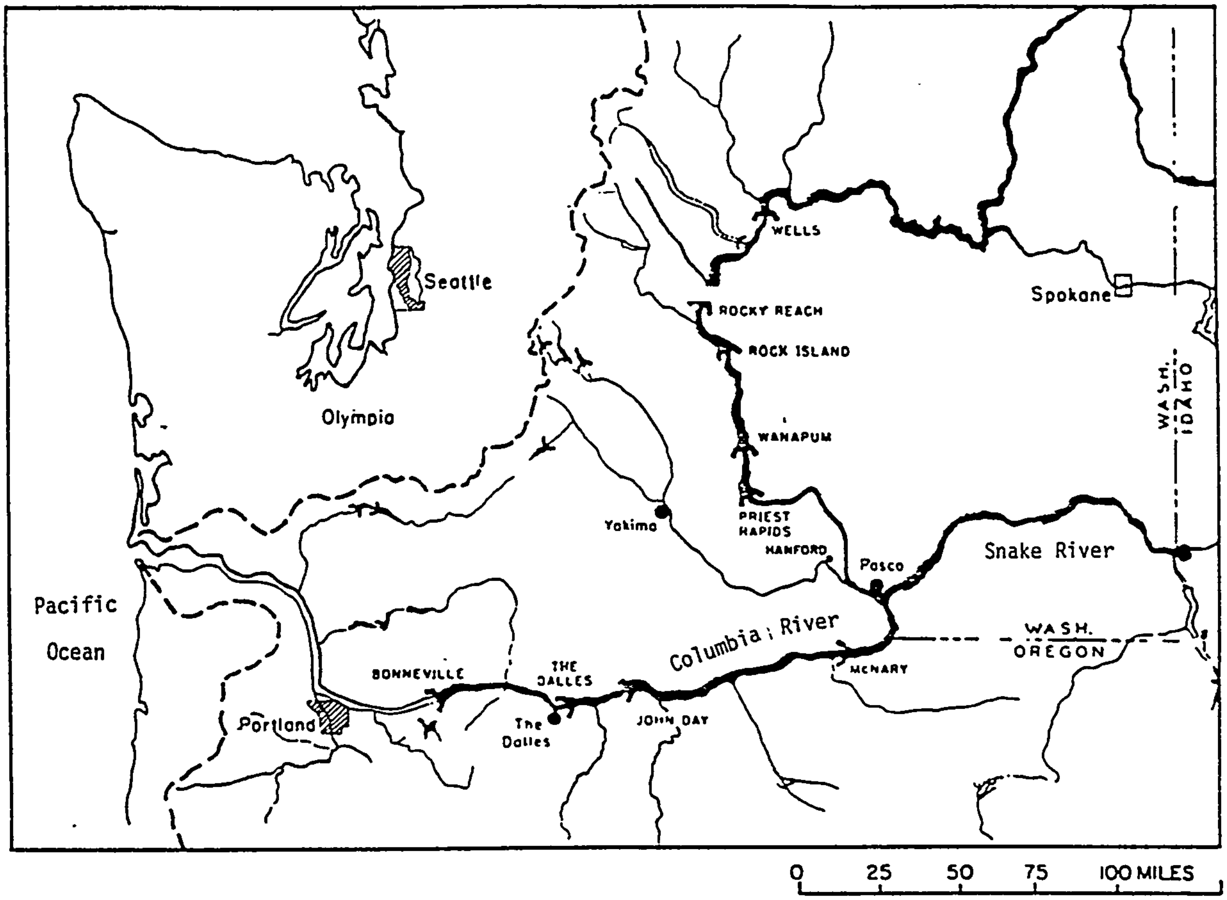

Figure 1. Mid- and Lower-Columbia River and Associated Hydroelectric Dam System modeled by SSS

With an abundance of alternative solutions, the value of a model that would allow comparison of methods was recognized by the mid-Columbia PUDs. They sponsored this study to develop and implement a model to simulate effects of the mid-Columbia hydroelectric facilities on salmon and steelhead smolt survival.

THE SMOLT SURVIVAL SIMULATOR (SSS) MODEL

The Columbia River and the associated hydroelectric dam system is modeled as a series of twelve reaches or pools that usually terminate with a dam. The first five (Wells, Rocky Reach, Rock Island, Wanapum, and Priest Rapids), represent the mid-Columbia. The sixth, Hanford Reach, from below Priest Rapids dam to the head of the McNary pool, is a free flowing reach. The next four reaches (McNary, John Day, The Dalles, and Bonneville) represent the lower-Columbia. The last two reaches represent the Columbia River Estuary and the Pacific Ocean. Figure 1 shows the river model geographically.

Each of the first ten reach/dam complexes in the river model is compartmentalized as depicted in Figure 2. Fish enter each reach by successful passage past the preceding dam, or from hatcheries or tributaries located within that reach. The river reach is divided into a number of travel compartments to simulate the number of days the smolts will spend in each reach. Each dam is composed of three compartments; spillway, turbine, and bypass facility. Smolt passage through the various dam compartments is based on a percentage of the population reaching that dam. Smolts that enter the bypass facilities may be transported past one or several of the dams. Smolts leaving hatcheries may also be transported around the dams. A survival rate is associated with each compartment of the system.

Steelhead trout and four salmon species, Spring Chinook, Summer Chinook, Coho, and Sockeye, are modeled. Species-specific survival rates are applied to the fish as they pass through each compartment of the model.

IMPLEMENTATION DESCRIPTION

The SSS model was implemented in May 1982. The Apple II microcomputer was selected because of its availability and prevalence at that time. The Apple II was configured with 64 kilobytes of memory and a single 5.25" disk drive. The software was written in interpretive AppleSoft BASIC.

To accommodate the software on the Apple II, the program was segmented into six modules. The modules are executed consecutively using the BASIC CHAIN statement as depicted in Figure 3. Variable values are preserved by reserving a data area in memory with the COMMON statement. Four of the modules form the user interface, (SSS, EDIT1, EDIT2, and MENU). SSS introduces the model to the user and provides general instructions on use of the program. SSS also defines and initializes data structures. The edit modules

* Trademark/brand name - no endorsement

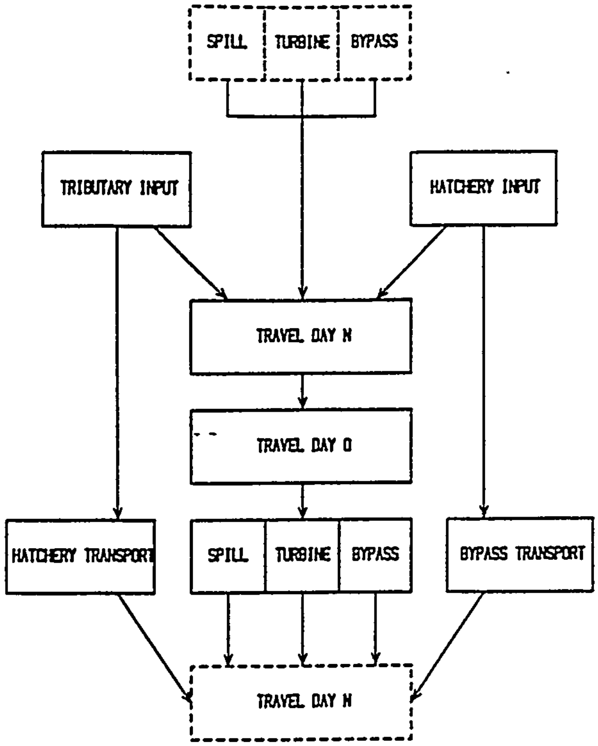

Figure 2. SSS Reach/Dam Compartment Model

control scenario creation and modification. EDIT1 allows the user to modify river and dam-specific parameters. EDIT2 controls selection of species-specific parameters. MENU handles program control. The simulation is performed by the module FISH. FISH generates screen reports during the simulation and summary screen reports of the scenario. REPORT generates the printer report and plot file. Two additional programs, WATER and RAND, control editing of data files. The random-access data files contains dam-specific and species-specific default values.

USER INTERFACE METHODOLOGY

The intended use of the model places a strong emphasis on the interaction between the software and the user. The methodology used to develop that interface is presented in this section. The techniques employed are divided into methods that reduce user stress and methods that allow thorough and easy parameter modification.

User Stress Reduction.

Stress will be reduced if the user is comfortable with the computer. This should be accomplished as early in the computer session as possible. Careful attention was given to this first meeting of the SSS software and the user. The program introduces itself to the user and gives a brief summary of its function. The user is given general instructions on use of the program. For example, a "yes or no" question can be answered with a "Y or N" respectively. By the time the inexperienced user has read through these pages he or she is likely to be be more comfortable with the program for several reasons. Information about what the program can do has been gained. Those responsible for the software have been identified thus making the program seem a little less impersonal. The user has performed four or five successful depressions of the "return" key.

A dialogue has been established between the user and the computer. The interaction should proceed in a predictable manner. The arrangement of questions should be tested on potential users to ensure that the presentation is in logical order. Using similar screen formats or templates for all questions will increase the continuity of the dialogue.

Thorough and Easy Parameter Modification.

Several guidelines control the parameter modification process in the SSS software:

- The screen is erased prior to presentation on each parameter or set of parameters.
- Queries are either phrased as questions or as sentences containing the default value.
- Mnemonics are avoided. Parameters are described in term familiar to the user.
- Menus of selected options or allowable ranges are displayed when appropriate.
- Default parameters are displayed with each query.
- A null entry (depressing the "return" key) will invoke the default.
- User input is tested to establish that it is within the allowable range.
- Edited parameters are redisplayed for verification.
- Queries are worded so that a scenario with default values can be created using the null entry only.
- The user is allowed to bypass sets of parameters.

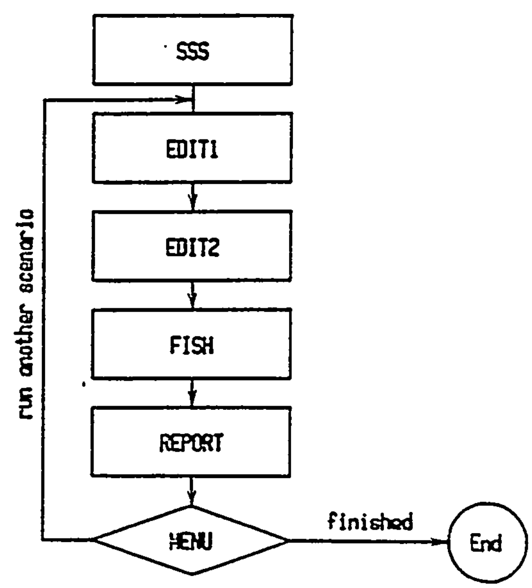

Figure 3. SSS Code Modularization Logic

Assuming that introductions have been made, the conversation between the user and the SSS software begins with questions of a general nature. The first user queries are for the date, the user's initials, and a title to identify the scenario. The program accepts responses in any format. The next two queries to be considered in the SSS program are the beginning and ending locations of the scenario. Figures 4-6 depict this dialogue between the computer and the user. The default locations are identified. The user is asked if the parameter requires a change. If the user responds positively, a list of the available options is displayed as depicted in Figure 5. The user is given instructions on how to change the value; in this case, to enter the index of the selected option. After the edit, the selected option is displayed for review. (See Figure 6.)

```
######################################

THE SMOLTS ARE FOLLOWED FROM
WELLS THROUGH PRIEST RAPIDS.


DO YOU WISH TO MODIFY (DEFAULT=NO)? Y
```

Figure 4. Example A - Initial Question

```
######################################

   1 - WELLS
   2 - ROCKY REACH
   3 - ROCK ISLAND
   4 - WANAPUM
   5 - PRIESTS RAPIDS
   6 - HANFORD REACH
   7 - MCNARY
   8 - JOHN DAY
   9 - DALLES
  10 - BONNEVILLE
  11 - COLUMBIA ESTUARY
  12 - OCEAN TO COLUMBIA MOUTH

   ENTER INDEX OF LAST POOL/DAM TO
   CONSIDER? 12
```

Figure 5. Example A - Menu

```
######################################

THE SMOLTS ARE FOLLOWED FROM
WELLS THROUGH OCEAN TO COLUMBIA MOUTH.


DO YOU WISH TO MODIFY (DEFAULT=NO)?
```

Figure 6. Example A - Verification

Although each parameter is handled individually, all use the same logic structure when questioning the user. The screen is cleared prior to the presentation of each question. Either a single parameter or a group of similar parameters may be displayed along with their default values. The user is asked if any changes need to be made to this parameter or group of parameters. Throughout the running of the programs, depressing the "return" key is all that is necessary to instate the default condition. This substantially increases the ease and speed of the scenario creation/edit process. If no change is indicated by the user, the program proceeds to the next parameter. If the user wishes to change a parameter, the program will ask the user to enter a new value. If this particular edit concerns a set of parameters, the user will be queried about each parameter in succession; is that parameter to be modified, and if so, what is the new value. New values are then displayed and the user is again queried as to whether any of the parameters in the set need to be modified. The looping logic in the review/edit programs enables the user to verify the input data and to correct any mistakes made while inputting the data. The review/edit loop continues until the user indicates that no more changes are necessary.

Figures 7 and 8 illustrate how SSS questions the user on the set of river flow questions. In this case the user has indicated that some of the values need to be changed. A text page of monthly flow values is presented for each reach/dam complex in the user's scenario. The user is asked if any changes are to be

```
######################################
DO YOU WISH TO MODIFY RIVER FLOW
VALUES (DEFAULT=NO)? Y
```

Figure 7. Example B - Initial Question

```
######################################
        FLOW (KCFS) AT WELLS

            JAN      101
            FEB      115
            MAR      134
            APR      118
            MAY      153
            JUN      147
            JUL      111
            AUG       95
            SEP       68
            OCT       79
            NOV       97
            DEC      101

DO YOU WISH.TO MODIFY ANY OF THE ABOVE
(DEFAULT=NO)?Y
######################################
      JAN - MODIFY (DEFAULT=NO) ?
```

Figure 8. Example B - Menu and Edit

made at this complex. If the user responses positively, each month is displayed at the bottom of the screen and the user is asked whether that month's flow value requires modification. If the users responds with a "yes" or "Y". A new value can then be entered. By responding with null entries to these questions, the user can quickly "page" through the scenario parameter values.

The SSS software allows for efficient scenario constructions both when the user views all reach/dam complexes generically and when the user specifies parameters for each reach/dam complex. This is accomplished with two editing modes; specific and generic. In the specific mode, the user is shown a group of parameters and asked if any changes are to be made. If the user makes a positive response, a query will be made about each parameter in the set. The generic mode assumes that the user wishes to set up a scenario with the same values for most of the parameters in the set. In this mode the user is asked to supply a value that will then be stored for each parameter in the set. New values are then displayed and the user queried if additional changes are to be made for this set. If the user's response is positive, individual changes can be made during successive iterations of that editing loop.

Scenarios can be created quickly and efficiently using the techniques discussed above. A substantial amount of computer code will be required. Subroutines can be written to handle many of these tasks thereby reducing the volume of code.

CONCLUSION

The software audience has changed in the past five years. As microcomputers have become more prevalent, end users are communicating directly with software. A specially trained person, usually a programmer, no longer acts as a buffer between the computer and the end user. Designers and writers of software must respond to this change by creating user interfaces that reduce user stress and allow easy and thorough parameter manipulation.

REFERENCES

(1), (3),(4), (5) Mid-Columbia Fish and Wildlife Program. 1982. Northwest Power Planning Council, 700 S.W. Taylor, Portland, Oregon.

(2) Public Law 96-501.