RRB 89-40

A MICROCOMPUTER-BASED SYSTEM FOR ENVIRONMENTAL MODELLING

by

D.A. Swayne*, J. Storey*,
A.S. Fraser and D.C.L. Lam

National Water Research Institute
Canada Centre for Inland Waters
Burlington, Ontario, L7R 4A6

*Department of Computing and
Information Sciences
University of Guelph
Guelph, Ontario N1G 2W1

May 1989
NWRI Contribution #89-111

MANAGEMENT PERSPECTIVE

With the advent of the microcomputer, the development of mathematical models has been supported by software with graphical and data interfaces. The RAISON system has made use of microcomputer technologies such as spreadsheets, geographical information system (G.I.S.) and database. This report summarizes the development in these aspects. It points to a new level of computer simulation, with fast interactive graphical presentation of the results and effective use of database linkage. It can also be used to improve the efficiency in data storage, analysis and archives.

PERSPECTIVE-GESTION

Depuis l'arrivée des micro-ordinateurs, des logiciels comportant des interfaces pour le graphisme et les données ont été utilisés pour la création des modèles mathématiques. Le système RAISON fait appel à plusieurs des ressources de la micro-informatique telles que les tableurs, les systèmes d'informations géographiques (SIG) et les bases de données. Ce rapport fait le point en ce domaine et laisse entrevoir un nouveau type de simulation informatique caractérisé par une présentation graphique interactive rapide des résultats et une utilisation efficace des liaisons avec les bases de données. Le système peut aussi servir à augmenter l'efficacité en matière de stockage, d'analyse et d'archivage des données.

RÉSUMÉ

Le projet RAISON de l'Institut national de recherche sur les eaux du Canada a été mis sur pied pour la modélisation, fondée sur une base de connaissances, des effets des pluies acides et autres effluents industriels dans les milieux aquatiques.

Le logiciel a été créé dans le langage de programmation C pour les ordinateurs MS-DOS tels que les IBM PC et les clones. L'affichage graphique est conforme à plusieurs normes relatives aux systèmes PC. Le projet RAISON comporte quatre sous-systèmes en plus de certains produits auxiliaires dont une bibliothèque mathématique et statistique. Ces quatre sous-systèmes sont : un tableur, un système d'informations géographiques, un système de gestion de base de données, de même qu'un langage et un environnement de programmation.

L'interconnexion entre les sous-systèmes est forte. Le tableur et le langage de programmation comportent des fonctions permettant de tracer des cartes directement sur l'écran. De plus, le langage de programmation reconnaît les structures du tableur et ses pages de données sont organisées selon ces structures.

On a pu réaliser, à partir de ce logiciel, des applications importantes dans le domaine de la modélisation et de la surveillance de la qualité de l'eau dans des contextes variés. La classification des eaux des lacs selon leur acidité, la modélisation, sous forme de systèmes experts, de l'acidité de l'eau des lacs et des effluents miniers et l'étude de la qualité des approvisionnements publics en eau sont des domaines d'application dans lesquels le système RAISON s'avère fructueux.

A MICROCOMPUTER-BASED SYSTEM FOR ENVIRONMENTAL MODELING

D. A. Swayne [1], *John Storey* [1]
A. S. Fraser [2], *and D. C.-L. Lam* [2]

[1]Department of Computing and Information Science,
University of Guelph,
Guelph, Ontario, Canada, N1G2W1

[2]National Water Research Institute,
Burlington, Ontario, Canada, L7R4A6

ABSTRACT

The RAISON project at the (Canadian) National Water Research Research Institute has been concerned with knowledge-based modeling of the effects of acid rain and other industrial effluent in the aquatic environment.

The software is implemented in the programming language C, for MS-DOS machines such as IBM PC and clones. Graphical output conforms with with several PC-based standards. There are four subsystems to RAISON in addition to ancillary products such as a mathematical / statistical library. They consist of: a spreadsheet, a geographical information system, a database management system, and a programming language and programming shell.

The interconnection between subsystems is strong. Direct on-screen map functions are available from both the spreadsheet and the programming language. Likewise, the programming language recognizes (and has its data pages organized like) the spreadsheet structures.

The utility of the software is such that significant applications products have been successfully developed for water quality modeling and monitoring in a wide variety of situations. Classification of lake acidity, expert systems modeling of lake acidity and mine effluent, and the quality of public water supplies are examples of successful application domains for RAISON.

KEYWORDS: environmental applications, decision support systems, geographical information systems, PC software.

INTRODUCTION AND BACKGROUND

The RAISON system was originally developed for the National Water Research Institute for analyzing and reporting acid rain data. The prototype was first described in the literature in [Swayne and Fraser, 1986]. It initially consisted of a customized spreadsheet, a map system and a database management system, implemented in the programming language C, for the IBM PC or compatibles.

Data was downloaded to the PC from a large aquatic database, NAQUADAT. Organized by sampling station, the data was retrieved into the spreadsheet for simple model calculations such as soil sensitivity classification, or ion rosettes [Fraser 1986]. Maps were input as line drawings, creating closed polygons on the screen (which was standard IBM CGA color). Direct output from the spreadsheet calculations to color the screen using color fill functions at the coordinates of the sampling station became an integral part of the acid rain reporting. More standard histogram and line-drawing software was created to fulfil the need for graphical reportage.

Since the spreadsheet was a special purpose product, compromises were struck to make it faster at the expense of the generality found in the commercial products. Full floating-point arithmetic was adopted, with floating decimal output and an extensive scientific function library. Database units (gram-molecular or SI units optionally) were input from the NAQUADAT by the utility programs, and automatic conversion routines incorporated to minimize the chance for error from inappropriate choice of units.

The system was written in a two-screen version (CGA and monochrome, later to be supplanted by EGA and monochrome), but it automatically reverts to single-screen mode when only one screen is present. The reasons for two screens rather than one were: (especially in earlier versions of the software) the monochrome screen provided faster output, and a larger information content with the display; and windowing software was (and remains) too heavy a load both in computation and memory consumption to be a viable option.

In this paper we describe some of the features of the system, as it has developed since the early version. It is quite different from the marriage of three or four commercial products communicating through the host's file system. As a generator of applications it has allowed quick prototyping in a number of complex systems involving reporting, geographical distribution of data, and rudimentary inference capability.

To date we have implemented several prototype expert systems for the synthesis of large-scale (in a geographical sense) models of lake acidification, a reporting system for analyzing and modelling mine effluent which includes analysis of text of reports in compiling its results, a monitoring system for public water supplies for use in developing countries which incorporates human and agricultural activity and health data as well as water source and well type when reporting results from field analyses.

OVERVIEW

In order to understand the structure of the system we distinguish between a user and an implementor. When we refer to a user of a RAISON system, we mean a person working with an existing set of maps and data who is manipulating by means of existing programs and the spreadsheet data from the database, and is using existing tools developed and implemented in RAISON. Conversely, an implementor is one who is developing a particular database map-set and collection of RAISON programs. An implementor requires a toolkit to build applications, a set of line-drawn maps, and a large database from which relevant data collections may be abstracted for a particular problem. In some cases, map digitizing software or other tools normally required for implementation may be used also. Gen-

erally, a hierarchy of regional data and maps, customized analysis programs, and simple methods for extracting knowledge from the data are all that a user requires. Much of the time in our projects has been spent in implementation Use of the results occupies a minor portion of the effort. Therefore the implementor's collection of tools is most important to our application.

A user typically sees a top level map (Figure 1), which contains a set of small boxes (icons) which may be selected to proceed to lower level maps. Applications are selected from menus or simple typed inputs, and results displayed graphically or in spreadsheet form. "Pretty-print" text may be added to either pictures (maps and/or graphs) or to spreadsheets. Graphical "windows" allow overlays of line or bar charts upon other images of graphs or of maps. The windows are not representative of new execution environments, however, but are rather designed to complete the presentation of results. Simple sets of rules may be input or modified in certain execution environments involving interaction of the programming language interpreter with the database. As well, spreadsheet macros may be inserted and calculated as with any commercial spreadsheet product.

THE DATABASE SYSTEM

The database management system (called RDBS or RAISON Data Base System) allows data collections to be designed, installed and updated using existing (or new) file formats. Character, floating-point and date fields are allowed, as well as text strings. Collections may be joined using matching fields. As an extension to this idea, near-matches may be defined (e.g., same month, Spring, neighboring sampling station). Many other utilitarian functions have been developed. New fields may easily be added, and the database restructured. See Figure 2 for database interaction.

THE SPREADSHEET

The spreadsheet subsystem is the primary vehicle for system manipulation. Database retrieval loads the joined data into a spreadsheet, and a large selection of menu options are available, including manipulation of the other subsystems (maps, database, statistics, and system maintenance/configuration).

Compromises to maximize performance have been struck. The number of spreadsheet columns is fixed at compilation time. The maximum size of the spreadsheet can either be determined from available memory, or allowed to grow as a virtual entity (limited by disk capacity but possibly slower in computational speed). This second option has been largely ignored in practice.

Even when incomplete RAISON's are generated (missing perhaps the graphics routines or whatever), the spreadsheet subsystem is always present. It forms the backbone of the interpretive programming language, RPL (RAISON Programming Language).

RPL has a similarity to BASIC in most of its control structures, except for its internal memory which is organized like a spreadsheet. Memory spreadsheet cells, and database (RDBS) entities are declarable in RPL. SI and gram-molecular units, map and stastistical functions, and other spreadsheet/database attributes are known to RPL. RPL can execute source code stored in the database ("runprog" or "runsub" for macros or closed subroutines), spawn other RPL programs, or even a new application (using the shell described next). Trace and interaction windows are optionally opened as RPL programs execute. Many applications consist of only one or two RPL programs, which generate spreasheets linked together with the maps and chemistry database [Lam et al. 1988]. In Figure 3 we see a sample spreadsheet.

RAISON'S SHELL AND PROGRAMMING LANGUAGE

RAISON's primary control language, mirroring the menu options in the spreadsheet with significant extensions, is called RSH (RAISON SHELL). RSH macros may be constructed and invoked to give a particular "flavor" to RAISON. The typical introductory user interface and long sessions with the expert system prototypes are written in RSH. For instance, a RSH macro called STARTUP.RAS is invoked to set up the current system, establish directory paths, and display initial maps and text information.

The programming language is called RAISON Programming Language or RPL. RPL has the usual simple structure for assignment statements as found in BASIC or FORTRAN. Constants may be numeric or string. RPL has an extensive set of operators, for scientific functions, database operations, spreadsheet operations, and subprogram execution. It is basically a character string interpreter. Declarations are based on first use and modified dynamically by most recent use (MRU). Rudimentary list processing, memory allocation and deallocation (with automatic garbage collection), graphics primitives, map primitives,and mechanisms for closed subroutines are all language features which have been implemented.

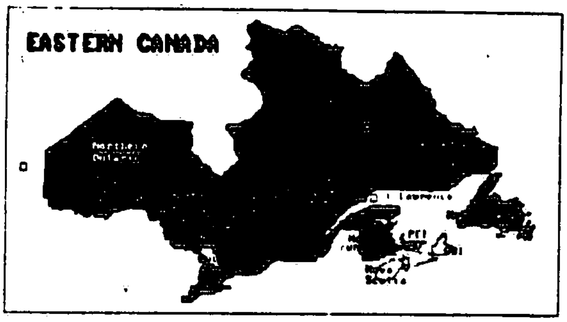
The organization of memory into spreadsheets is most useful for organizing the data analysis. Spreadsheets may be created or loaded, and data variables and constants may be placed at specific locations within them. Key locations like the end of data loaded into the spreadsheet are reserved variables, and the variables generated by program are stored in a page of scalar data attached to the spreadsheet. Simple rules (frames) for calculation or map coloring can be attached to spreasheet cells an executed in parallel (conceptually, at least).

THE MAP SUBSYSTEM

The map sybsystem includes map creation and editing functions. Input maps (in point form) are stored in RDBS files. Edit functions (from the spreadsheet) allow fixups and alterations to the maps and associated icon data (also stored in the database). The map subsystem includes the ability to create and store "snapshots". This can be done interactively, from RPL or from RSH. Snapshots are bit-mapped images of screens. Data from contour plots can be accessed either on the current screen or in a snapshot (*.SNP) file. Likewise, contour or map lines may be extracted using a search algorithm designed to perform curve tracing on the bit-mapped image. This, and an algorithm for recognizing the interior of a polygon, form the primitive operations in RAISON's geographical information system. In the GIS, we have chosen to store the bit-mapped images rather than more complex hierarchical data structures. Since we are building applications more interested in developing and accessing a knowledge-base, and even some of our time-dependent simulations are image-oriented [Wong et al. 1980], the image-oriented approach has proven adequate. Set-theoretic operations (**and**, **or**, etc.) have been implemented to allow overlay operations and to identify image segments. Again, these operations are both interactive and available in the programming language RPL.

Given the uneven distribution of the sampling stations, the contouring algorithm developed for RAISON is in fact a potential map. The scaled value at each point, weighted by the square of the distance from the data points is calculated and normalized. While this algorithm has obvious shortcomings, other methods such as tessellations and spline fitting are completely unusable in most of our circumstances (because of the uneven distribution of data sampling points). For contour sketching, we optionally provide a description in the knowledge base to modify the spherical influence of data points (such as cutoffs by shoreline in the case of lake or river contours).

Figure 1. Top level map and simultaneous display on monochrome screen.




```
                    R A I S O N   M I C R O

                          CONTROL KEYS
<F2>-Spreadsheet                 <Alt><F2>-Redraw map
<F3>-Database                    <Alt><F3>-Back out and redraw
<F4>-Display Stations            <Alt><F4>-Zoom in and redraw
<F5>-Display Top Map             <Alt><F5>-Set scale factor
<F6>-Polygon                     <Alt><F6>-Set level
<F7>-Identify Icon
<F8>-Identify Station            <Alt><F8>-Add Icon
<F9>-Up One Level                <Alt><F9>-Snapshot
                                 <Alt><F10>-Action and Snapshot
<Enter>-Select Icon


<Ctrl><X>-Exit Raison Micro


LL 5212 7423     Base 4102 5038     XY 320 196     Scale 15.00     Level 0
```

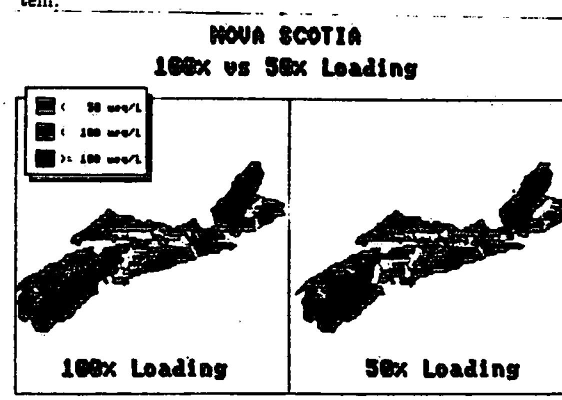
Figure 2. Database interaction. Action selection, format selection and a typical format.

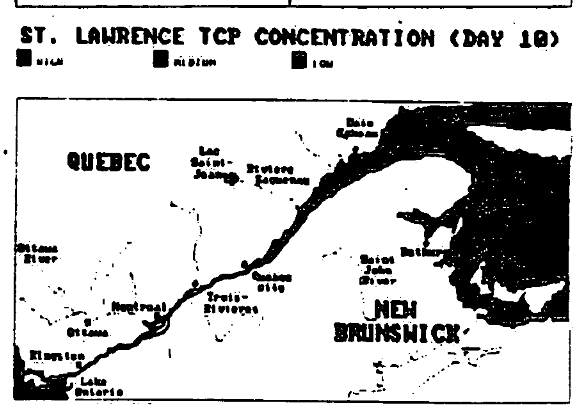
```
                         Raison Database System
        Create     Update    Link       Import    Options           Quit
                  Database File
                  Alternate File
                  Delete Records

                          CREATE FILE FORMAT
                          AVAILABLE LAYOUTS
*1) Watershed location file with Action.
 2) Water quality station data (Na,K,Ca,Mg,Cl,So4,Alk,pH). Time series.
 3) Terrestrial data (watershed area and area weighted sensitivity).
 4) Lake classification data.
 5) Discharge data.
 6) Water quality station samples(Na,K,Ca,Mg,Cl,So4,Alk,pH,Alum,Doc)
 7) Model error
 8) Water quality samples(Na,K,Ca,Mg,Cl,SO4,ANC,pH,Al,DOC) Time series.
 9) Location of stations (water quality, discharge).
 10) Number of lakes and water quality stations for each watershed.
 Choose desired layout ...

                             Message

Layout file:layouts.rdl.        Number of Layouts: 10.


                          Enter/Change Data
F1-Help      E/F2-Edit Rec.           A/F3-Append New Rec.        F10-Quit
F4-Mark      F5-App. to Spreadsheet   F9-Search
                         Add/Update Database
Station:

Date:

Na:              K:               Ca:              Mg:

Cl:              So4:             Alk:             pH:

Aluminum:           Doc:

                             Message
Water quality station samples(Na,K,Ca,Mg,Cl,So4,Alk,pH,Alum,Doc)

 Is this the desired layout? (y/n)
```

Figure 3. Sample spreadsheet, and some of the menu options.

```
A1: 'AG

          A          B        C        D        E         F        G
 ▶ 1◂   AG        CDR100   ED100    CDR50    ED50
   2    AG1       0.012135 0.005824 0.027702 0.028580
   3    AG2       0.029539 0.032286 0.044704 0.061708
   4    AG3       0.109691 0.091021 0.128514 0.128466
   5    AG4      -0.01867 -0.03070 0.023872 0.022882
   6    AG5      -0.05358 -0.05358 -0.02232 -0.02232
   7    AG6       0.059778 0.055500 0.077172 0.086758
   8    AG7       0.021804 0.015376 0.043841 0.046634
   9    AG8       0.080078 0.072897 0.094909 0.104154
  10    AG9       0.241661 0.240299 0.241661 0.240299
  11    AG10      0.042858 0.049896 0.045841 0.052258
  12    AG11      0.092914 0.086601 0.092914 0.086601
  13    AG12      0.604497 0.676928 0.604718 0.704543
  14    AG13      0.096957 0.100362 0.104296 0.118391
  15    AG14      0.075834 0.082380 0.090242 0.119262
  16    AG15      0.209924 0.153543 0.210145 0.176773
  17    AG16      0.238421 0.172498 0.238864 0.223877
  18    AG17      0.208716 0.144210 0.208735 0.164121
</>Menu <F1>Help <F2>Edit <^I>Ins <^T>Del


 rEcalc  rAnge  BackColour  Worksheet  Retrieve  DataBase  aNalysis  Files
Re-calculate the worksheet

A ---------------------------- HELP -----------------------------
 Topic:CALCULATE
 CALCULATION HELP
      These options allow built in functions to be applied
 to the rows of data in the spreadsheet.
      Each function appies itself to the rows of data which
 have a date. The date column identifies the rows of data
 which are obtained from the database.

  CDR: calculates the  Cat-ion denudation rate using:

       NA, CA, K, MG, discharge rate and watershed area

  EXCESS: calculates the Excess SO4 using:

       SO4, discharge and watershed area

  RUNOFF: calculates the runoff:

       discharge / basin area
 -- F1-Display list of available topics ---------------------------
```

Figure 4. Some of the varied reports using the map system.





Other innovative tools have been designed and implemented. For instance, if an X-Y or scatter plot of pH or similar observations is made, the line-drawing and GIS software may be invoked to determine the spreadsheet entries which generated a particular point or set of points on the screen.

Sample map outputs (including contouring) are shown in Figure 4.

SYSTEM DESCRIPTION

The structure of a RAISON implementation is manipulated by an implementor interacting with RSH and various special-purpose routines written in RPL. Startup scripts are generated and snapshots of key maps along with icons and action directives are built in the database. Water chemistry or other data are loaded and linked into the system through icons on the various levels of maps ("0" ≡ TOPLEVEL, "1" ≡ NEXTLEVEL, etc.). Data files containing valid RPL representations of rules are loaded, and file directives ("FILELIST.PC") created to specify paths and actions. The whole package is then reviewed by a potential user in order to incorporate necessary or desired features.

System organization is clearly completely dependent on the application, and RAISON is only the toolkit for that implementation. It has become identified with the end-products, however, because of its successful application in many diverse projects: acid rain analysis, the safety of water supplies, and the distribution/effects of pesticides and mine effluents are current examples.

In the hands of moderately skilled implementors, applications can be built and tested quickly. The tight integration of subsystems through the database / spreadsheet (instead of the use of several different commercial products with file transfer mechanisms) has resulted in efficient implementations and also in the ability to generate animated presentations of findings. Originally, we fought with the much slower earlier PC's and less-capable software products, which led us to create our own product. Now, we find it simpler to continue refining RAISON, rather than moving to commercial packages, and to push the limits on the software whose architecture we know intimately.

ACKNOWLEGEMENTS

Swayne and Storey are supported through Research Contracts from Environment Canada: contracts DSS#03SE.KW405-8-0540 and 03SE.KW405-8-0050. A loyal and patient user community have contributed unnumerable suggestions for improvements to RAISON.

REFERENCES

1 A. S. Fraser, "Aquatic characterization for Resources at Risk in Eastern Canada", Water, Air and Soil Pollution, 1986.

2 D. C. L. Lam, D. A. Swayne, John Storey, and A. S. Fraser, "Regional Acidification Models using the Expert Systems Approach", to appear 1989.

3. D. C. L. Lam, D. A. Swayne, John Storey, A. S. Fraser, and I. Wong, "Regional Analysis of Watershed Acidification using the Expert Systems Approach", Environmental Software, 3 (3), September 1988, pp. 127-134. Computational Mechanics Publications, U.K.

4 D. Lam, D. Swayne, A. El-Shaarawi and John Storey, "Water Quality Data Management (Malaysia/Canada): From RAISON/Acid Rain to RAISON/Water Quality", Interim Report to International Development Research Centre Coliphage Meeting, Banff, Alberta, September 5-7, 1988. (IDRC File No. 3-P-86-1051-01)

5 D. A. Swayne, D. C. L. Lam, A. S. Fraser, and John Storey, "Regional Acidification Models using the Expert Systems Approach", Expert Systems Theory and Applications IASTED International Conference, Geneva, Switzerland, (June 14-16, 1988), pp.19-22.

6 D. A. Swayne and A. S. Fraser, "Development of an Expert System/Intelligent Interface for Acid Rain Analysis", Microcomputers in Civil Engineering, (1), December 1986, pp. 181-185.

7 I. Wong, C. R. Murthy, D. A. Swayne, and D. C.-L. Lam, "Fast Graphical Simulations of Spills and Plumes for Application to the Great Lakes", to appear, 1989.