Advances in Water Resources Technology and Management, Tsakiris & Santos (eds)
© 1994 Balkema, Rotterdam, ISBN 90 5410 389 2

Opsis: An intelligent tool for hydrologic data processing and visualisation

George Tsakalias & Demetris Koutsoyiannis
Division of Water Resources, Hydraulic and Maritime Engineering, National Technical University of Athens, Greece

ABSTRACT: Opsis is a hydrologic computer application that performs database administration, data visualisation and hydrologic and statistical computations. It is developed in an object oriented environment with a wide use of graphical tools. It establishes a representation of hydrologic data series as objects with standard characteristics, and a protocol that hydrologic procedures should follow when acting on objects. It is connected to a large hydrologic database and has been used by several Greek organisations.

1. INTRODUCTION

The power-generality tradeoff problem in hydrologic computer applications is usually resolved by a commitment either to powerful but rigid hydrologic applications (e.g. HYDRA-PC, Koutsoyiannis et al., 1991; Data Base Management Software for Hydrological Data, RMIB, 1986, etc.) or to general but weak (from a hydrologic point of view) commercial spreadsheets. Recently, to provide an alternative power-generality tradeoff we have developed Opsis, a powerful hydrologic computer programme, general inside its own scope. As shown in Table 1, Opsis can perform tasks belonging to three categories, i.e. database administration, hydrologic computations-statistics and data visualisation. It borrows its key characteristics from concepts of object oriented programming (Stabler 1986), pattern directed programming (Bratko, 1991, p. 553-560), expert systems (Bratko, 1990, p. 331-357) and blackboard systems (Hayes-Roth,1985; Engelmore and Morgan, 1988).

The Opsis components (objects and procedures) and their internal connections as well as its connections with external sources are shown in Figure 1.

2. REPRESENTATION OF HYDROLOGIC DATA SERIES

The key issue in a computer programme design is the representation of the program's world. A program's world is the set of all factors (i.e. data types, objects and procedures) that concern the tasks performed by the particular programme. The basic idea of Opsis is the uniform representation of hydrologic task factors. Uniformity comprises declarative and procedural aspects of the tasks. Declarative uniformity is accomplished by considering the hydrologic data series as objects with uniform private and visible attributes called *series objects* (or simply *series*). The series and their attributes are independent of the data physical meaning and adequately general to represent effectively almost any type of data series. In every series there are attached numerous ancillary objects that hold information about the relation of the series

Table 1: Categories of tasks performed by Opsis

Database administration	*Hydrologic applications*	*Data visualisation*
SQL procedures	Rating curves	Graphs - Charts
Data transferring	Frequency analysis	Reports
Aggregation	Regression	Graphical user interface
Interpolation	Data computations	
Quality controls	Series administration	

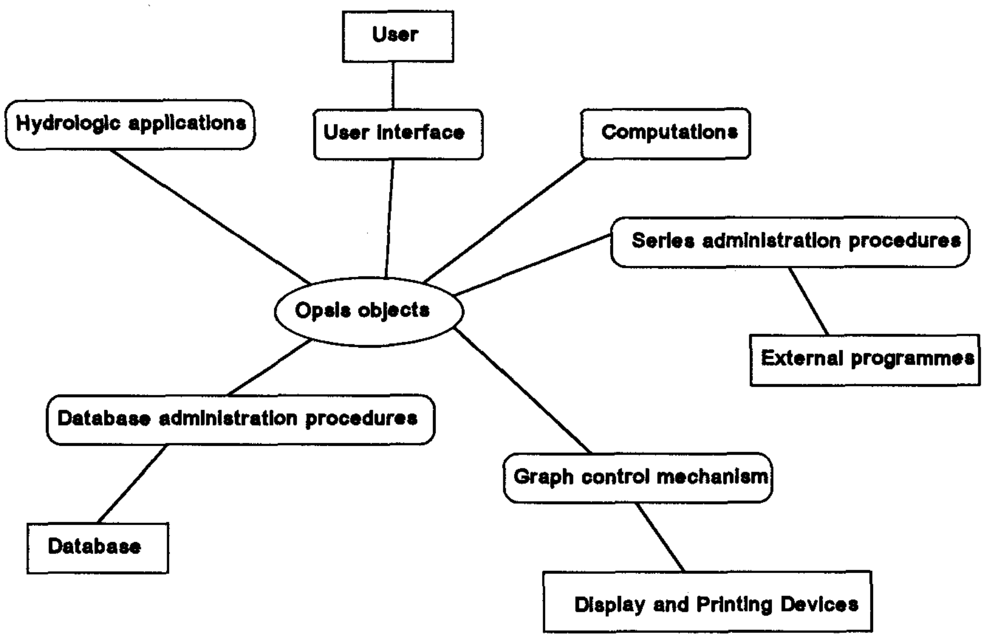

Figure 1: Opsis components

and the database, the visualisation characteristics of the series, and the parameters required for the computations that might regard the particular series. Opsis objects are accessible from all system procedures and from the user interface.

3. HYDROLOGIC OBJECTS

During the development of Opsis, special attention has been paid to the successful representation of hydrologic objects. In this section we will focus on this representation, describing the main system objects.

3.1. Series object

Most hydrologic procedures are applied on time series. We can consider the results of most processes as data series as well. For example, the result of a frequency analysis can be a data series that contains the coordinates of the empirical probability density function of the original data series. Thus, the main object of Opsis is the *series object* which contains an array of *row objects* and some ancillary objects associated with the series as described below.

3.2. Row object

Each hydrologic measurement is represented by a row object. Some of the row object attributes are shown in Table 2.

Table 2: Row object description

Attribute	*Description*
Value array	It can express rows of the natural data with one of the following forms: date-value (e.g. date, temperature), date-value 1-value 2 (e.g. date, water discharge, sediment discharge), range-date-value (e.g. 18 days, last date, cumulative rainfall depth) or range-value 1-value 2 (e.g. class range, class starting value, frequency of observations).
Connection	It can be either FALSE or TRUE indicating an isolated or a connected row, respectively. Isolated rows are not visible in plots nor take part in computations (e.g. suspicious measurements in a regression analysis).
Break	When a series must be split (e.g. during double mass analysis), every derived subseries starts at a row declared as a break point (giving to its break attribute the value TRUE).
Action	The action attribute takes an integer value that specifies what sequence of database queries will be performed when it will be necessary.

3.3. *Database object*

When a series is produced by a database query, a database object is created and associated with the series. Some of the database object attributes are shown in Table 3.

3.4. *Variable type object*

This object holds specific information on the variable type that the series represents. For example, in case of monthly rainfall this object includes attributes like the default step (1 month), the default maximum value (e.g. 500 mm) and the default minimum value (0 mm).

The variable type object is a typical *parameter object* of Opsis. All Opsis procedures require some parameters. These are attributes of parameter objects. Procedures that create series attach parameter objects to them. For example, all series produced by an SQL query hold in their variable type object default values for range checks and temporal consistency controls.

Table 3: Database object description

Attribute	*Description*
Source table	It holds the name of the database table from which the series is selected.
Destination table	It contains the name of the database table where the series will be saved (perhaps after some modifications).
Instrument	It is a unique key for the instrument that measured the data of the series.
Storage structure	It defines the storage structure of the particular series.

3.5. *Curve object*

When a series represents coordinates of a computed curve (e.g. a rating curve) we probably want to know the way the series was computed. This information is provided by the curve object with attributes such as interpolation type (e.g. linear, logarithmic etc.), offset array (constant offsets on non linear curves), etc.

3.6. *Visualisation object*

The visualisation object holds information about the visual characteristics of the series when it is plotted on a diagram or displayed on a spreadsheet. The object's attributes concern the type, size and colour of displayed marks, lines and labels in the series diagram, and the format of the displayed values in the series spreadsheet.

3.7. *Axis object*

The most important attribute of the axis object is the type attribute. Opsis currently provides decimal, logarithmic, normal, exponential, Gumbel, Weibull and time axis types. The time axis can provide a minute, hour, day, month or year scale. In addition, the axis object holds information about the diagram ticks, minimum and maximum values, etc. It is noted that the Opsis graph control mechanism is authorised to modify the axis attributes consulting a rule base, when they are not consistent to the series displayed.

3.8. *Configuration object*

It holds information about the specific configuration

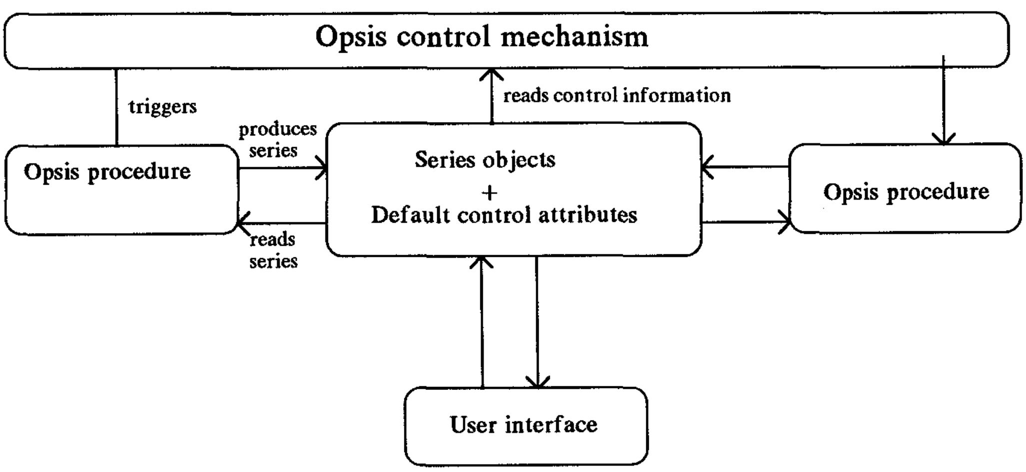

Figure 2: Opsis control architecture

of the value array of each row (see Table 2) and the way a sorting of rows will be performed if demanded.

3.9. *Statistics object*

When Opsis performs a statistical computation on a series, an ancillary statistics object is attached to the series. This object can contain attributes like sums, statistical moments, correlation coefficients, regression parameters, extreme values, etc.

4. HYDROLOGIC PROCEDURES SCHEME

Opsis procedures act on the series objects in a standard and transparent way, thus providing a procedural uniformity of the system. All Opsis procedures are independent with each other, which means that there is no procedure calling another procedure. This scheme might seem strange but it follows one of the most successful intelligent systems architecture, that is the blackboard control architecture (Hayes-Roth, 1985). Some benefits of this functional independence are the simplicity of the system control, expandability and modifiability.

The procedures communicate only indirectly, through their products. They act on series objects and produce series objects. The Opsis control mechanism decides what procedure will be executed next, triggers that procedure and lets it decide for the feasibility of its action. The only control competence granted to the Opsis procedures is to record default control directives to the ancillary objects of their products. The scheme of the Opsis control architecture is shown in Figure 2.

Each Opsis procedure is applied to an *input base* and produces an *output base* considering for this production a *parameter object*. The base (input or output) is an array of pointers to the series held by the specific Opsis session. A sequence of procedures acting on a set of series objects is called an Opsis task. As it actually revealed, most of hydrologic tasks can follow the uniform Opsis task representation, and with a certain number of steps they can naturally achieve their goal. Some examples of such tasks are the analysis and visualisation of single hydrologic series (e.g. hyetographs, hydrographs), the comparison and processing of multiple data series (e.g. double mass curve analysis, regression etc.), the interpretation of hydrologic data (e.g. construction of stage-discharge relationships), etc.

Each output base inherits ancillary objects from the input base series, depending on the particular procedure applied, which can selectively modify certain objects. In every Opsis procedure, the programmer declares the ancillary objects that are transported from the input base series to the output base series. The declarative programming of Opsis procedures enhance their readability and expandability.

4.1. *Built-in procedures*

The most useful built-in procedures are shown in Table 4. They are currently divided into five major categories: database administration, series administration, transformations, frequency analysis and regression.

4.2. *Opsis macros*

The system provides a user and programmer interface to all its built-in procedures via high level macros. The input and output uniformity and the declarative encoding of the Opsis procedures, facilitate the use of the Opsis macros. Every Opsis task (a sequence of procedures) can easily form a macro. During a macro execution, each subsequent procedure acts on an input base that is the output base of the previous procedure.

The built-in procedures are designed considering their use inside macro operations. For example, there is no procedure that produces a double series object in one step. Instead of this, the programme provides a procedure that produces double series (taking columns and ancillary objects from two parent series) and a procedure that produces mass series. So, the user can set up a macro with these two procedures, apply the macro to an input base and receive the desired result. The user can save the macro for later use and he or she can use the mass or the double series procedure in other tasks.

Owing to the procedural and declarative uniformity, every new procedure added to the system (in order to perform a specific task) is not a top end of the programme but rather a dynamic language element that can be used in subsequent phases by other procedures to accomplish other tasks.

4.3. *Transparency of Opsis procedures.*

Another important feature of the system is the transparency of procedures. All tasks that are performed automatically by the programme can be also performed step by step, and in each step the hydrologist has direct access to the process parameters, input and output through the friendly user interface.

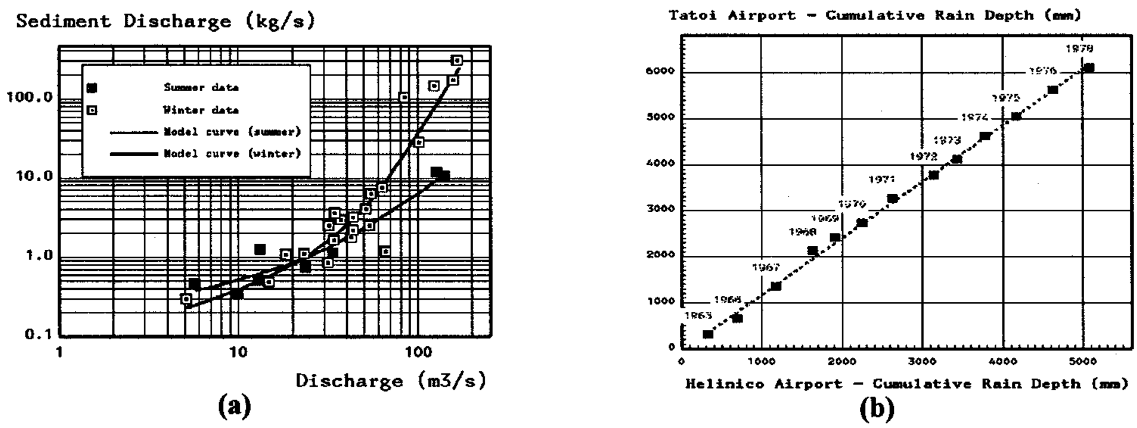

Figure 3: Typical graphs produced by the Opsis graph module: (a) sediment discharge rating curve and (b) double mass curve of annual rainfall data

5. THE GRAPH CONTROL MECHANISM

The uniform representation of objects inside Opsis facilitates the use of an intelligent rule based graphics module, the graph control mechanism. The graph control mechanism receives the output of the performed tasks and feeds back any changes in data, which might have been done by using the series diagram as a graphical interface (e.g. by moving a mark with the mouse etc.). The graph control mechanism takes care of all tasks required, to insure that data visualisation will always be in the proper form and level of abstraction. This level provides the hydrologist all and only these aspects of data needed for his or her job. The user is not concerned with cumbersome tasks which are all accomplished by the rule based graph control mechanism, though he or she always keeps the privilege of direct access to all diagram components. Typical examples of graphs produced by the graph module of Opsis are shown in Figure 3. It should be noted that such graphs are also used as an integral part of the user interface, so that the user can graphically modify them (e.g. the rating curve of Figure 3a) and the modifications are directly reflected to the series objects.

6. CURRENT STATE

Opsis is currently implemented in the X-Windows environment under UNIX. It is connected to Ingres relational database, providing direct access to HYDROSCOPE (the Greek national hydrologic data bank). It is also connected to external hydrologic programmes - models. A typical session of Opsis is depicted in Figure 4. (The reader may notice that the

Table 4: Categories of Opsis built-in procedures

Category	*Specific procedures*
Database administration	SQL procedures performing sequences of SQL queries (Select, Insert, Delete, Update) Data manipulation procedures that use various storage structures Procedures for aggregation, interpolation, extreme values extraction, mean values extraction, etc. Quality control procedures, such as range check, temporal, spatial and internal consistency control, etc. Data transferring procedures, such as importing data from other programmes to the database and vice versa
Series administration	Generation of empty series Importing data from ASCII files Series splitting Connected and isolated rows separation Reports writing Double series generation Column swapping
Transformations	Polynomial transformations Logarithmic transformations Exponential transformations Cumulative series production Column computations Curve interpolation
Frequency analysis	Box plots Density plots Frequency and probability plots Duration curves
Regression	Linear Exponential Polynomial Regression with smoothing

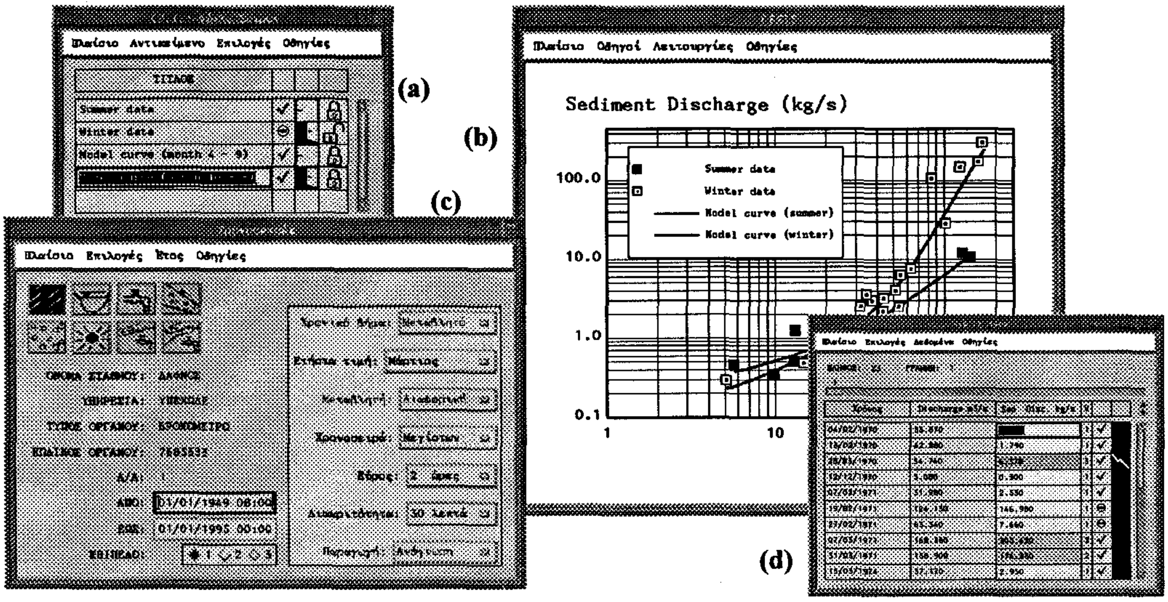

Figure 4: View of a typical Opsis session on a X-terminal: (a) session adminisration frame (b) diagram frame, (c) time series frame and (d) series spreadsheet

user interface currently communicates with the user in Greek, although specific labels can be written in other languages such as English)

ACKNOWLEDGEMENTS

Opsis has been developed in the framework of the HYDROSCOPE project. HYDROSCOPE was approved by the Greek General Secretariat of Research and Technology and was incorporated into the STRIDE HELLAS research programme funded by the European Union. Additional funding was provided by Greek Organisations, i.e., the Ministry of Industry, Energy and Technology, the Ministry of Environment, Physical Planning and Public Works, the Ministry of Agriculture, the Ministry of Education, the Water Supply and Sewage Corporation of Athens, the National Meteorological Service, the Public Power Corporation and the General Secretariat of Research and Technology.

The authors wish to thank A. Christofides N. Mamassis, A. Manetas, I. Nalbantis, N. Papakostas and A. Sakellariou for their contructive comments and their contribution on related HYDROSCOPE components.

REFERENCES

Bratko, I., *Prolog, Programming for Artificial Intelligence*, Addison Wesley, 1990.

Engelmore, R. and Morgan, T. (eds.) *Blackboard Systems*, Addison Wesley, 1988.

Hayes-Roth, B., A blackboard architecture for control, *Artificial Intelligence* 26(2), 251-321, 1985.

Koutsoyiannis, D., Tsolakidis, K. and Mamassis, N., HYDRA-PC, A data base system for regional hydrological data management, in G. Tsakiris (ed.) *Advances in water resources technology*, Balkema, Rotterdam, 1991.

RMIB (Royal Meteorological Institute of Belgium), *Data Base Management Software for Hydrological Data on Microcomputer, Explanatory notice*, G06.3.01 (micro), Brussels, Belgium, 1986.

Stabler, E.P., Object-oriented programming in Prolog, *AI Expert*, 46-57, Oct. 1986.