Design and Implementation of Environmental Decision Support Systems with Object-orientation and Spreadsheets

Ari Jolma
Helsinki University of Technology, Laboratory of Water Resources, 02150 Espoo, Finland.

Abstract Systems analysis techniques -- simulation and optimization models -- have been applied for environmental management for many decades. More recently decision support systems (DSSs) have been designed and implemented to ease and speed up the use of these techniques. In the literature there are numerous reports of DSSs for environmental management. The development effort of these systems has many times been substantial -- even several work years. This paper describes an attempt to improve the design and implementation of environmental DSSs with object-orientation and sophisticated spreadsheets. The main hypothesis is that these tools can be used to form a core of a practical methodology that will result in more resilient and open systems in less time. The principles of object-orientation are presented from the point of view of analysis and conceptualization of environmental systems. A practical set of tools for using spreadsheets as the foundation of DSSs is presented. The use of these tools with an object-oriented analysis is shown to achieve some important goals. A framework for system design is described which attempts to balance the many aspects of environmental management. The described DSS framework is currently being used for several applications and case studies in the water quality management of river basins in Central and East European countries and in operation of Finnish lakes and reservoirs. The first ones of the aforementioned studies are conducted at the International Institute for Applied Systems Analysis (IIASA) in Laxenburg, Austria. The second ones are conducted at the Helsinki University of Technology, Espoo, Finland.

1. INTRODUCTION

The management of natural resources is a task of many disciplines. Sometimes the problem is more technical one; e.g., how to minimize the output of a certain hazardous waste from a certain industrial plant. Sometimes the problem requires a more integrated look; e.g., how to upgrade all waste water treatment facilities of an entire river basin. This study focuses in the integrated management of environment. Integrated management is not just a technical problem, nor it is just a scientific problem, nor institutional or law enforcement problem, but it comprises all of these.

Systems analysis techniques -- simulation and optimization models -- have been applied for environmental management for many decades. However, simulation and optimization models are nowadays regarded more as tools or parts in larger systems or they are used as a part of the underlying structure of more ingenious approaches. In the last ten or twenty years decision support systems (DSSs) have been designed and implemented for environmental management. This work has been reported for instance at many conferences and workshops [Goodchild et al. 1993, Kovar and Nachtnebel 1993, Loucks and da Costa 1991, NCGIA 1993, Tsakiris and Santos 1994]. Also at IIASA there have traditionally been a strong tendency to design and build DSSs [Fedra 1992, Ivanov et al. 1995, Makowski 1994, Orlovski et al. 1986]. The development effort of these systems has many times been substantial -- usually several work years.

This paper describes an attempt to improve the design and implementation of environmental DSSs with object-orientation and sophisticated spreadsheets. The main hypothesis is that these tools can be used to form a core of a practical methodology that will result in more resilient and open systems in less time. The principles of object-orientation are presented from the point of view of analysis and conceptualization of environmental systems. A practical set of tools for using spreadsheets as the foundation of DSSs is presented. The use of these tools with an object-oriented analysis is shown to achieve some important goals. A framework for system design is described which attempts to balance the many aspects of environmental management. The described DSS framework is currently being used for several applications and case studies in the water quality management of river basins in Central and East European countries and in operation of Finnish lakes and reservoirs. The first ones of the aforementioned studies are conducted at the International Institute for Applied Systems Analysis (IIASA) in Laxenburg, Austria. The second ones are conducted at Helsinki University of Technology, Espoo, Finland.

2. INTEGRATED ENVIRONMENTAL MANAGEMENT

Environmental management is a combination of many different and different kinds of activities to use natural resources or to maintain or improve their quality. The success of management is judged against the expectations of different parties involved. In practice environmental management is an art that operates with subjective knowledge of the managed system, databases containing information about it, models for different phenomena, conceptualizations of the managed system, development plans of the society, quality goals, etc. On the other end of this variety there is the scientific order and reasoning, and on the other end there is the political art of possibilities. Between these there is many times hopelessly complex and uncertain real world. It is the understanding of the author of this paper that it is the job of the practical engineering to bring these two together by building an intellectual bridge to connect them.

The word "integrated" has two meanings in the integrated environmental management. First, it means that the managed system is treated as a whole, meaning an integrated look at the joint effect of different actions. Second, it means that the results of science are integrated into the data of the real world. The main benefits of the scientific side of integrated management are the savings of resources that are possible. It might also be possible to prevent situations where actions in one place are foiled by no actions in another place. A third benefit is the chance to check where the biggest problems are and subsequently concentrate on them.

A major manifestation of environmental management is the policy managers adopt. The environmental management policy is a set of rules by which the environmental quality is judged, actions are carried out and the success of the whole management is judged. Usually this policy is in a constant change as the society and institutions develop. Times when policies change more give good opportunities for the research community to introduce new, more advanced models and methods into the policy reformation process. This is for instance the case currently in post-communistic Central and East-European countries. In the case of integrated management these models and methods should ideally form one integrated system.

Scientific research on all the different fields of study in environmental management is well developed and it has produced countless numbers of detailed descriptions and models of their subjects. The engineering task is to select the most suitable models and apply them on the complex real cases. The one most important thing guiding this is the availability of data. Another important thing is to balance the different parts of the system so that their level of detail is approximately the same. The level of detail is of course mainly dictated by the relative importance of the part regarding the overall purpose of the system. The third part of the environmental management, policy formation with decision making, deals then mainly with the utilization of the models or their results.

3. DECISION SUPPORT SYSTEMS

The concept of a Decision Support System (DSS) evolved in 1970's and 80's to describe a system that is an aid to managers in large and unstructured problems. A DSS was seen as a chance to get the tools which research in systems analysis had created to an actual use to those who could benefit from them. A DSS is a rather vague concept but it is principally seen as a computer system with three basic parts: a database, a model base, and a user interface. As a DSS is a complex piece of software, it is usually not just one executable program but a number of them and databases, configuration files, etc. So besides being a scientific and engineering project, building a DSS is also a quite large software engineering project. A recent review of application of Decision Support System methodology in the field of water resources is given by Loucks [1994].

A DSS can assist environmental management in all of its three parts: modeling, application of models/building of applications and utilization of models. All these are major tasks and all of them can benefit from the application of DSS methodology -- that is at least the hypothesis. Many reported decision support systems for environmental management are a result of a significant if not massive amount of programming work. This is usually a result of the required ease-of-use and flexibility that are not simple to realize. In the last ten years there has been much development work done in building general purpose interface tools. This has made the development work easier for the interface part but it has also generated new expectations for system designers.

One way to distinguish two kinds of DSSs is to divide them into generic and case specific. A generic decision support system for integrated environmental management requires a description of the object of the management. In the case of river basin management this implies a need for topological description of the river network, locations of the point sources, etc. It is a dilemma of generic decision support systems whether to model any phenomena in nature or to simplify the description and force the phenomena into something that is modeled.

Another way to distinguish DSSs into two categories is to divide them into model centered ones and data centered ones. A model-centered DSS is according to Makowski [1994] a DSS that uses an underlying mathematical model. A data centered DSS is according to Reitsma [1993] a system that relies and builds largely on and around the data the organization has available.

4. OBJECT-ORIENTED APPROACH

The object-oriented approach is a method for organizing the information the analyst has about the analyzed system. A basic concept in object-oriented analysis is the object and its class. An object is a unit, which binds together data and methods for it. The class of an object describes the data (attributes) an object contains and the actions (methods) the object can be called upon to do. An object is an instance of a class and every object is unique.

Object-oriented approach is mentioned many times with computer programming. However, object-oriented approach has a much broader field of application than just programming. Object-oriented modeling is according to Rumbaugh et al. [1991] "a new way of thinking about problems using models organized around real-world concepts". Rumbaugh et al. [1991] has also identified three kinds of models:

1. Object model: captures the static structure of the system
2. Dynamic model: shows the time-dependent behavior of the system
3. Functional model: shows how values are computed

Each one of these models describes the real-world system from one viewpoint, which is different from the other viewpoints. Rumbaugh et al. [1991] also give practical instruction on using the object-oriented approach for conceptualizing real systems.

Principles of object-oriented programming are according to Meyer [1988]:

- information hiding: objects properties might be visible only to the object itself
- genericity: a generic class is not directly usable; it combines the properties of more usable classes
- inheritance of properties: a class inherits the properties of its superclass
- polymorphism: an object acts according to its class (which is unknown to the user of the object)

An object in an environmental system is an understandable part in the system. An essential feature for an object is that it must be a subject of some action. There might be a couple of possible ways to define objects in the system. The best one is the one that gives the best results; that is, it has the best descriptional powers and still is simple. An object might also contain other objects.

Examples of objects in environmental models are: lakes, river stretches, industrial plants, etc. For instance a river stretch is an object having static attributes like length, cross-section, slope, etc., and dynamic attributes like discharge entering the reach, concentrations of pollutants, etc. A river stretch can also be linked to a mathematical description of the fate of pollutants in river. When a stretch-object has all this information it can calculate by itself the change in the concentrations of the pollutants.

In the case of reservoirs an object representation can make them "smart" [Behrens 1992]. Reservoir objects can be linked to mathematical or logical descriptions of policy issues besides physical descriptions like storage functions. This makes for instance reservoir specific data and calculations local and improves thus the maintainability and flexibility of the system.

Objects in environmental systems can be polymorphic. A waste water treatment plant (WWTP) is an example of such an object. A WWTP can be an existing, old one, or it can be hypothetical; that is, one that might be built in the future. The behavior of a plant depends on its kind: the treatment efficiency of an overloaded old plant is very different from a new plant that is designed for the same waste water discharge.

5. SPREADSHEET TECHNOLOGY

A spreadsheet is a computer program, which was invented and became very popular with the advent of modern personal computers. In fact a large proportion of the success of personal computers can be devoted to spreadsheets. At first a spreadsheet was just the traditional systematic tabular calculation method transferred to a computer. Later more and more features were added to the spreadsheet programs until nowadays they include many kinds of graphics, pre-defined data-analysis procedures, an astonishing number of possible formatting options, possibilities in linking with other programs, (graphical) user interface tools, tools for building database applications, and full-fledged programming languages.

Spreadsheets have always been good for quick ad hoc analysis. With their new features the analyst can now also accomplish a programming task that previously took the labor of a couple of skilled programmers, and still at the same time analyze the problem. This means that more and more problems can be tackled with a computer by more and more case specific conditions taken into account. This puts an emphasis on the analysis and understanding of the field. This seems to call for developments in analysis methodologies rather than in solution methodologies.

Spreadsheets have also grown in size and computational power requirements besides in features. Also the programming languages built into the spreadsheets are usually interpreted and therefore quite slow. Applications are also usually not very transportable from one spreadsheet environment to another. On the other hand, spreadsheets themselves have been transported from one hardware and operating environment to another.

5.1 Spreadsheet basics

Excel 5.0 (Microsoft 1993) will be used as an example of a modern spreadsheet here. However, this does not mean that it is unique in any way; there are also other spreadsheets that deliver same functionality.

An Excel 5.0 document is called a workbook. An application can include a number of workbooks. A workbook is organized into sheets. There are four kinds of sheets in Excel 5.0: work-, graph-, code- and dialogsheets. A worksheet delivers the functionality of a traditional spreadsheet. A worksheet is matrix of cells. A collection of cells is a range. A range can act as a unit: it can have a name and a single function. This makes for instance vector and matrix calculations possible. A cell or range contains data or a formula. A formula is a simple worksheet function or a more complex structure consisting of references to cells and worksheet functions.

A worksheet or a rectangular range can be used as a relational database table. From one perspective a table is a collection of fields (columns) which all have a name and data type. From the other perspective a table is a collection of structurally similar records (rows). Data in a table can be sorted according to one or more fields. Data in a table can also be filtered in Excel 5.0. In-place filtering means that the user can define logical conditions for the records s/he wants to investigate. A table can be dynamic, that is, it contains fields that are conditional functions of other fields, or it contains references to elsewhere in the system. This kind of functionality is achieved with built-in reference, lookup and filtering worksheet functions.

In Excel 5.0 it is also possible to add drawings (circles, lines, rectangles, etc.) to sheets. It is possible to access these drawings from the user-written code for creating, manipulating or studying. This functionality makes the creation of graphical interfaces simple.

Codesheets are for the user-written code. The code is organized into five kinds of units:

- type, constant and variable declarations, which are public to the sheet or to the whole application
- macros, which are procedures the user can directly evoke (call)
- subroutines, which are procedures callable only by another procedures
- custom functions, which are user-defined worksheet functions
- functions, which are procedures returning values

The user-written code is linked to other things in workbooks and in the Excel itself through pre-defined objects. The user can write code to create and read files and access other applications.

6. APPLICATION OF SPREADSHEETS FOR SIMULATION MODELING

This chapter describes a set of practical tools that were designed to be used in building environmental DSSs. How they are used depends on the application domain. Within an application domain it might be possible to build a framework for a DSS using these tools. A framework has then to be customized for each application by adjusting it to manage application specific details.

6.1 A template

By a template is in this paper meant a clonable (usually rectangular) range of cells that contains the definitions of parameters and input, output, and state variables of one object in the model. By a "template" is also already called (in Excel) a special document that can be used as a pattern for new documents. The context in which the term is used here is closer to the "template" of object-oriented (C++) programming. A template is a useful idea for object-oriented modeling with spreadsheets. When a template is copied and pasted, an object is created. This object can then be used to create a piece of a model. This procedure is similar to what happens when an object-oriented program run. In this case it all happens in a spreadsheet and the objects are concrete, visible ranges in the spreadsheet document when they are pasted into it. This makes building an object-oriented model a one-step procedure: a procedure that needs to be run only once per application.

Templates can mix worksheet functions with user-defined functions to give to the whole model a dynamic, user-definable character. Colors can be used to distinguish parameters from decision variables, input from output, etc. Templates can have some pre-defined links to their surrounding cells to give to the template a primitive self-organizing character. Self-organizing means that when templates are copied and pasted in a certain order, they will automatically form a body that starts to function as one system.

The cloning -- copying and pasting -- of templates can be done manually but it is also possible to write a macro that will do it. Normally this macro would use topological and other kinds of information that is stored in a database to create the model. During the execution of this macro it is usually necessary to replace the default values of parameters, etc. with object specific data. The object specific data comes normally also from a database. The macro also has to perform the additional linking of the model. A model may have to be linked with itself; that is, far-apart laying objects may have to be connected, different parts of the model may need to be connected, etc. A model may also have to be connected to a user interface or other part of the system. An input link is a reference to an input field in the user interface and an output link in the user interface is a reference to some output cell in some object in the model.

6.2 A lookup-table

A lookup table denotes in this paper a small table. There is not a lookup table concept in Excel 5.0 but there is a worksheet function LOOKUP, which can be used to look up values from lookup tables. Lookup tables are useful in defining lists and/or tables that have global and static nature. A lookup table is typically composed of a header and data area. An example of a lookup table would be the names of different land-uses that should be available in further modeling. There could be some global parameters, which describe some (stable) characteristics of these different land-uses, included into the land-uses lookup table.

6.3 A model

A model can be implemented into a spreadsheet application as a unit of cells, ranges, worksheets or even workbooks that produce automatically, semi-automatically or manually an output from a given input. A model can be a very simple one, just one or two equations requiring no interaction with the user. This kind of model is most likely implemented simply as a user-defined function. On the other hand a model may be very complex, built from templates of many kinds with a macro and requiring many kinds of user interactions. These are the two extremes and there are countless levels of complexities between them.

Many times one needs a model that has a modest size, requires little user interaction, and which is required for some specific, perhaps one-time-only per object, purpose. Good examples of models of this kind are models for parameter estimation. These kinds of models are convenient to implement within one worksheet that perhaps has an embedded chart for checking the fit, some option buttons for selecting for instance the estimation algorithm, etc. Use of colors as with templates is again helpful for the user. This kind of model can also be used from code simply putting inserting data to the input cells and then reading from the output cells. All calculations are automatically performed by the spreadsheet application and the programming interface becomes almost by default a very clean one.

6.4 A graphical user interface

Graphical user interface (GUI) is a well-established concept that means usually a front-end for a large program or an operating system. In this meaning a GUI is a very large concept. However, in this paper it will be also used to denote a quite restricted and well-confined piece of the whole system. A GUI is a view or window to a part of the system that utilizes mostly non-numeric and non-textual means: drawing and graphing. The concept is restricted because it is useful to think of the user interaction and interface as "screenfuls" or as parts that can be seen and understood in one glimpse.

As the two main means (drawing and graphing) suggest, there are two main kinds of GUIs available in Excel. With both means colors and different shapes can be used available, objects can be in front of or behind other objects, objects can be transparent or opaque, they can be grouped, etc. There are countless different graphing possibilities. In two-way interaction the user can also feed information into the system and not just read the output. Graphical interaction has traditionally often been restricted to being only one-way. For instance in Excel dragging a point in a graph automatically reflects the changes to the respective data in worksheets. For the drawings this kind of functionality is not built-in. However, with built-in programming capabilities they are not difficult to add. One can for instance read the coordinates of a line and store them into a database.

The power of the combination of drawing tools, data storage possibilities and coding is well presented in the creation of GUIs. It is possible to draw in some aspects of the system that is going to be studied and then read that drawing into a database. The drawing data, which may include important spatial and topological information of the system can then be combined with other data of the system. In the third step the original drawing can be reproduced and combined with output data from models.

7. IMPLEMENTATION OF ENVIRONMENTAL MODELS

7.1 The organization of a framework system

The framework system should be built with and around a number of databases and models. Models should act as agents that retrieve input data from databases when evoked and after processing restore it back to the databases. Parts of the system should be separated with interface functions.

A static database is used to store the stationary features of the system. These are mainly topological, hydrological, hydraulic, etc. properties. A scenario database is used to store the data of scenarios, the unstationary data. Other databases should be added according to the needs within the application domain. The access to a database can be open; that is, the application developer can write code or functions that directly read data from the tables. The access to a database can also be restricted; that is, the application developer can read data from the tables only using specific interface functions. The latter form of access might be necessary for polymorphic objects or for information hiding purposes.

There can be several models available in the system for the application developer and the user. However, the most important of these is the management model, which can also include several small models.

7.2 A management model

A management model includes at least a simulation model and an interface to it. A simulation model can be built -- manually or automatically with a macro -- using the information in the databases and the tools described in the previous chapter. There are three major steps that a macro has to go through when creating a management model. In the first step it creates a traditional run-time data structure of the system. In the second step this data structure is utilized when the object templates are cloned for each object in the system. In this step the cloned spreadsheet objects are given their characteristics. Additional linking that is needed after the self-organization is also performed in this step. In the third step the simulation model is linked to the interface.

An optimization model should be subordinate to the simulation model; that is, it should retrieve its input data from it and restores its output data back to it.

8. DISCUSSION

A DSS approach is in the literature targeted towards solving large and unstructured problems. This is indeed the case for water quality management. If the need is for an analysis that can be described to its details before the work starts, there is no need for a DSS. This is particularly true if the analysis is a traditional one. There might be a need for computerized tools but they should not be confused with DSSs. This does not deny the need the user might have at some point might have to perform a such analysis.

The role of DSSs in integrated water quality management of watersheds might still need some clarifications. Their value in organizing the available data and giving an integrated look into the system is undeniable. Their value in giving insight into processes taking place in the aquatic and economic life depends on the quality of the measurement and other data there is concerning these subjects. Clearly there are many different kinds of users that a DSS must cater for. The connection of a DSS to other information systems within an institution might be the weakest link in the whole chain.

The fields within water quality management have each a strong tradition in modeling. A DSS usually can-not have the latest and most detailed models -- often just because of their too big demands for data, is an easy target for criticism. One pitfall for developers of a DSS is to use incompatible models for different things. Incompatible meant here in the sense of the level of detail. It must be well understood that a DSS is not a research tool, it is a tool for analysis that is carried out for decision or policy making.

9. CONCLUSIONS

The conclusions that can be drawn at this stage are:

- The principle of genericity or the hypothetical benefits of a generic system did not hold well in the study. In many cases the work needed to fully implement a feature in a generic way would have been more laborious than manually putting it into the model. The spreadsheet environment allows an easy manipulation of the model to define unusual things.
- it pays to devote time in careful analysis of the application domain i.e. to model the information, later changes to the foundations of the system are time consuming and dangerous also in other ways
- the "framework" approach seems feasible but the work needed to apply the system to a new case may be substantial
- object-oriented approach can be applied also to application development in spreadsheets
- spreadsheet application development and programming environment is already mature enough host larger systems like decision support systems
- benefits from the spreadsheet environment are significant, particularly in the development time and flexibility towards ad hoc analysis
- drawbacks of spreadsheet environments can be circumvented by using external applications to specific tasks

10. REFERENCES

Behrens, J.S. 1992. Simulation of Reservoir Operation using Smart Reservoirs. Computing in Civil Engineering and GIS. ASCE. pp. 606-613.

Fedra, K. 1992. Advanced Computer Applications. *Options, International Institute for Applied Systems Analysis, Laxenburg, Austria* pp. 4-14.

Goodchild, M.F., Parks, B.O. and Steyaert, L.T. 1993. (Editors) Environmental Modeling with GIS. Oxford University Press.

Kovar, K. and Nachtnebel, H.P. (Editors) 1993. Application of Geographic Information Systems in Hydrology and Water Resources Management.

Loucks, D.P. 1994. Developing and Using Decision Support Systems: A Critique and a Challenge. A paper presented at the Second European Conference on Advances in Water Resources Technology and Management. June 14-18, 1994, Lisbon.

Loucks, D.P. and da Costa, J.R. (Editors) 1991. Decision Support Systems. Water Resources Planning. NATO ASI Series Vol. G 26. Springer-Verlag

Makowski, M. Design and Implementation of Model-based Decision Support Systems. *International Institute for Applied Systems Analysis, Laxenburg, Austria* WP-94-86.

Meyer, B. 1988. *Object-oriented Software Construction.* Prentice-Hall International.

Microsoft Corporation. 1993. Excel 5.0 spreadsheet. A commercial product.

NCGIA (National Center for Geographic Information and Analysis). 1993. Proceedings of the Second International Conference/Workshop on Integrating Geographic Information Systems and Environmental Modeling. September 26-30, 1993. Breckenridge, Colorado, USA.

Orlovski, S. Kaden, S. and van Walsum, P.E.V. (Editors) 1986. Decision support systems for the analysis of regional water policies. *International Institute for Applied Systems Analysis, Laxenburg, Austria* WP-86-33

Reitsma, R.F. 1993. Bootstrapping River Basin Models Through Object Orientation and GIS Topology. A paper presented at the Second International Conference/Workshop on Integrating Geographic Information Systems and Environmental Modeling. September 26-30, 1993. Breckenridge, Colorado, USA.

Rumbaugh, J., Blaha, M., Premerlani, W. Eddy, F. and Lorensen, W. 1991. *Object-Oriented Modeling and Design.* Prentice-Hall International.

Tsakiris, G. and Santos, M.A. (Editors) 1994. Advances in Water Resources Technology and Management. Proceedings of the Second European Conference on Advances in Water Resources Technology and Management. June 14-18, 1994, Lisbon. Published for EWRA by A.A. Balkema.