

# Interactive Computer Technology for Planning and Policy Modeling

KURT FEDRA

*International Institute for Applied Systems Analysis, Laxenburg, Austria*

DANIEL P. LOUCKS

*School of Civil and Environmental Engineering, Cornell University, Ithaca, New York*

This paper speculates on the potential impacts of our increasing access to and use of computer technology and communication, especially with respect to planning and policy making. The focus of the discussion is on the interaction between the users of this expanding technology and the technology itself. Those involved in its development, whether it be the hardware or software of this technology, are in a position to make substantial contributions toward a more effective use of the models and their data bases by planners and policy analysts. Specific features of the technology and of environmental planning and policy-making processes are examined to identify where and how interactive computer-based models and associated hardware can best serve individuals, their organizations or institutions. Finally, the necessary conditions for the successful implementation of such tools and methods are identified.

## INTRODUCTION

In the introduction to this series of papers on interactive approaches to environmental and resource management policy analysis the basic question raised was, "How can we develop and apply our models in a manner that will increase their utility to planners and policy makers?" This question is motivated by the obvious and increasing need for more effective planning and policy making leading to "better" plans and policies. This paper is inspired not only by this question, but also by a technology: by the revolutionary development in the electronic data processing and communications industry. This revolution seems to offer improved means and ways to address two major problems in large-scale planning and systems analysis: complexity and communication.

We also make one basic assumption: namely, that formal, i.e., numerical modeling of experimentally unaccessible processes and systems, is a worthwhile exercise. Observed shortcomings are not a problem of modeling per se, but rather of proper usage and effective communication. We are also arguing for a new approach to modeling that can benefit from the new technology. Thus the question is not whether to model, but how, and most importantly, how to interface models with our more traditional ways of planning and decision making.

Computer-assisted methods of planning and design, information management, and process control have proven to be highly successful, and once implemented, indispensable in numerous fields of application. Examples range from worldwide airline reservation information systems and air traffic control to industrial design and manufacturing. Based on the lessons being learned in fields like computer-aided design and manufacturing [Myers, 1982], in electronic or mechanical engineering [Anon, 1982], business-oriented management or decision-support systems [Riley, 1981; Sprague and Carlson, 1982], and data base and information systems, as well as on our own, admittedly limited, experience in the water resources and environmental field [Rogers, 1979; Friedman, 1982; Loucks et al., this issue(a, b)] this paper tries to (1) analyze the specific features of the environmental policy making, planning, and man-

agement process to see where and how computer-assisted methods might best fit; (2) draft a general structure for computer-assisted interactive methods, based on an assessment of desirable features, given the requirements of the planning and policymaking process on one side and the directions of hardware and software development on the other; and (3) identify some necessary conditions for the successful implementation of such tools and methods into existing institutional structures.

At this point it seems appropriate to add a note of caution, a caveat: beware of naive technological optimism. Computers alone are not going to solve anything; and, in fact, much can be said (and justifiably so) against their all too intimate involvement in human affairs [Weizenbaum, 1976]. However, this expanding technology can provide a common language and framework for the necessary multidisciplinary cooperation or stimulus and focus for new approaches to the solution of both old and new problems—eventually.

## THE MESSY PROCESSES OF PLANNING AND POLICY MAKING

Our understanding of the way much of our water resources and environmental planning and policymaking takes place has been outlined in our introductory paper [Loucks et al., this issue(a)]. It seems clear that many basic problems for society involving water and the environment can be understood, at least a little better, through the use of comprehensive systematic modeling. The difficulty in applying such comprehensive formal systematic methods, however, is that the institutions that manage many aspects of those problems are not organized and functioning in the same way. The capability of working with comprehensive policy models and of using the resulting information to help develop more effective comprehensive plans and policies seems to be severely limited in today's very political and often fragmented planning and policy-making environment [Friedman, 1982; Brown, 1982].

Institutional and organizational barriers will always exist. We as model builders have to learn more about how to build our models in a manner that not only makes their output more useful for our planning and policymaking, but that also makes the models more easily usable by those individuals involved in planning and policymaking within these institutions. The human-model interface provided by the computer can play a very important role in accomplishing this.

Copyright 1985 by the American Geophysical Union.

Paper number 4W0975.  
0043-1397/85/004W-0975\$05.00

The very nature of the systems dealt with and the conflicts and problems associated with them calls for subjective evaluation or informed judgment. These are aspects of a political rather than a purely formal and analytical process. The obvious, and perhaps only, way to include these elements of subjective judgment during formalized policy analysis is by directly, i.e., interactively, involving the users or their legitimate representatives or technical staff or elected delegates. Since environmental planning and decision making is a political process (with strong elements of ethics and psychology) as much as a formal analytical one, it deals with subjective perceptions of realities as much as with what scientists would call reality itself. Also, by far the simplest, most straightforward and honest method to acknowledge this is to build environmental analysis right into the decision-making process as has evolved in society.

#### A BLUEPRINT FOR INTERACTIVE ANALYSIS

To build computer-assisted methods of water resources or environmental analysis into the planning and decision-making process (a rather elusive concept), some specific features are required. Easy access and easy use are important, as are reliability and credibility. The often cited user friendliness is very important. User friendliness implies a style and language (i.e., jargon and symbolism) of the interaction between the model user and the model that is familiar and easy to understand. In short, the system has to be attractive and useful for its users. In addition, costs of both the computer hardware and software and the manpower for operation and maintenance must be low relative to the perceived benefits. Finally, the computer-assisted system of models must be compatible with the other tools and methods used in the planning and policy-making process.

To cite from a recent paper on artificial intelligence [Winston and Brown, 1980]

"Situations that require control and action typically involve teams of decision-making experts that must handle vast amounts of data. Unfortunately, ordinary computer technology can make the problem worse, rather than better, since too much information overloads human decision-making procedures. Consequently, there is a need to create intelligent support systems ..."

Also, "... representation is the most important research issue."

With more and cheaper computer power becoming available, more of this power can be used to improve the representation aspect. More and cheaper computer power is available to create an interface engineered to support human planning and decision-making procedures without the introduction of a rigid and demanding formalism. Whereas traditionally, interaction with computers was designed to make things easy and straightforward for the machine, at the expense of the human operator, the advent of abundant and dedicated computer power should allow for a reversal of this approach. Given the vastly increased capabilities of modern hardware, one can afford to be wasteful, from the machine's point of view, to make it easier for people to interact with the machine. For the above problems this requires that the formal methods, or at least the user interface, are cast into the structure and language of the respective institutional framework, as well as the problem's context. The basic principle is to organize information to facilitate judgment.

#### The Man-Machine Interface

To make any projections into the future with regard to computer technology is risky. There have been so many ad-

vances in such a short period of time that it is unlikely too many are aware of what is possible today, let alone what will be likely in the future. However, several workers have summarized some of the trends or advances that have taken place in computer technology over the past few decades [Sebestyen, 1981; Geoffrion, 1983; Branscomb, 1982; Gomory, 1983; Brennan and Molloy, 1983]. Some of these trends are presented in the plots of Figures 1 and 2. Given this progress it seems not at all unreasonable to assume that most of the features of today's supercomputers will soon be available in small, affordable machines.

This provides the starting point for this section: we will assume that all the products that can now be envisioned (largely because they are already announced, although not all commercially available yet) will be available and at a low cost. This list includes fast (i.e., with clock speeds well above 10 MHz) 16- or 32-bit processor chips with virtual memory support; 256 kilobit to 1 megabit random access memory (RAM) chips; fast and high-capacity (several hundred megabyte up to gigabyte dimensions for videodisks [Sebestyen, 1981]) random access mass storage devices of miniature dimensions and suitable for portable units; high-density (i.e.,  $1024 \times 1024$  or more) bit-mapped color graphics devices with high refresh rates and any number of simultaneous colors; reliable voice recognition, high-speed networking with other computers or data bases via public utility lines and/or satellite; and so on. In other words, we will assume that in a truly portable workstation of briefcase dimensions, featuring a large, high-resolution color graphics screen, we can have a fast 32-bit CPU with virtual memory capabilities, supplemented by dedicated processors (input/output array processors, graphics), several megabytes of directly addressable fast RAM, several hundred megabytes of high-performance minidisk drives, alternative operating systems, and a vast array of utility programs, problem-specific, high-level languages, compatible at the object code level, and the necessary interfaces for fast networking or host access (Figures 3 and 4). Given all that, Will it improve environmental planning and decision making? Where are the bottlenecks?

One obvious bottleneck is the man-machine interface in a general sense, i.e., the user's ability (and willingness) to exploit this technology, including the development of the necessary

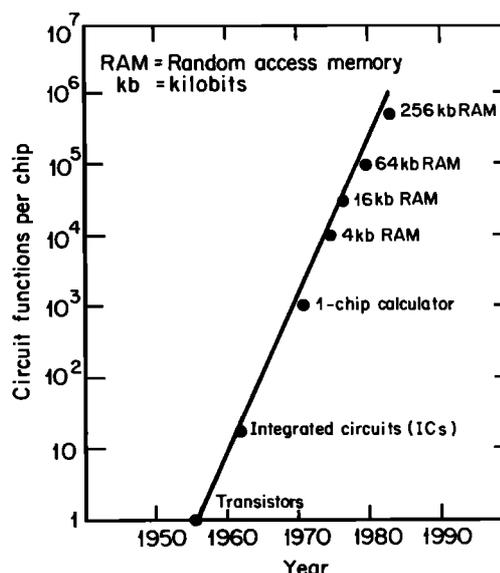


Fig. 1. One of many indications of the increase in computer technology over time.

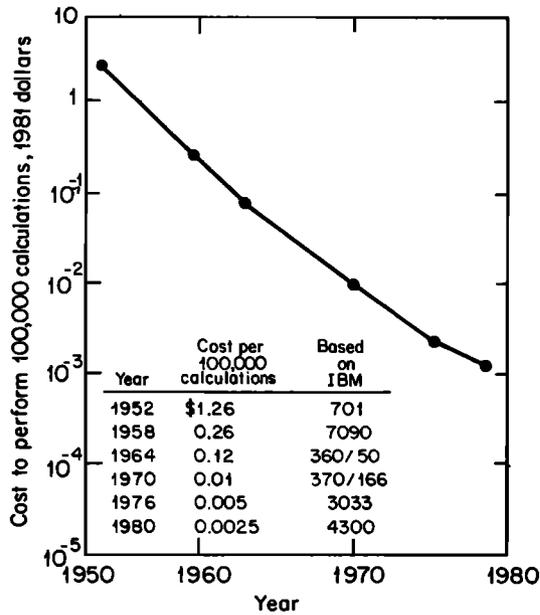


Fig. 2. One of many indications of the decrease in computer and computing costs over time.

application software. Another bottleneck is the man-machine interface in a technical sense; i.e., how can a human and a machine effectively communicate with each other given their vastly different modes and speeds of information processing. To improve this interface, there are two basic approaches: user education (getting the user closer to the machine's requirements), or machine education (getting the machine closer to the user's requirements). Both approaches have their potential, but both have limitations.

Educating users takes time and, for the higher echelons of existing decision and policymaking, this seems unlikely in any reasonable amount of time. However, one can imagine that the intermediate and lower ranks of technical staffs could develop some of the necessary skills. Also, one also can safely assume the coming generations of engineers, scientists, and professionals will have considerably more computer-oriented

skills than those found today among their elders. A top-level decision maker may easily endorse the use of a new approach, but it is the people who are actually exposed to it that determine its success or failure.

Existing software for modeling and decision-aiding spans a wide range of "user friendliness." Many large noninteractive programs (e.g., many of the Hydrologic Engineering Center (HEC) models of the U.S. Army Corps of Engineers) look friendly at best for their creators or those that use them often. Other programs, especially some written for personal microcomputers, are at a much higher level of user friendliness. The revolution in microcomputers is not only in the hardware, but also in the decision-aiding, interactive, often graphical software [Belshaw, 1982]. User friendliness is often equated with menu-driven approaches, i.e., techniques that allow computers to be used without the knowledge of any classical programming language. Almost all that is required from the user is to pick one out of several options offered by the machine, until the required action is completely specified [e.g., Yenarichs, 1982]. Clearly, this involves trade-offs between simplicity, speed of use, complexity of the interface, hardware and software cost, and the flexibility of applications.

User friendliness, however, also includes more general aspects such as semantic and syntactic consistency, graceful (and instructive) recovery from failures, and a rich assortment of input-output structures and devices, interactive graphics being the most obvious one [e.g., Foley and Van Dam, 1982]. However, a user-friendly interface requires that the underlying software, i.e., the models, are easily understood, well structured, and compatible with the mental processes used in less formal approaches to solving complex problems.

A NEW APPROACH TO MODELING

A close examination of the basic principles of computer modeling and simulation modeling in particular, as usually practiced in water resources and environmental systems analysis, will reveal a conservative bias being built into the use of the computer. Still in its infancy, "the first use of the new technology [i.e., the computer], is quite naturally to do in a slightly different way what had been done before without it"

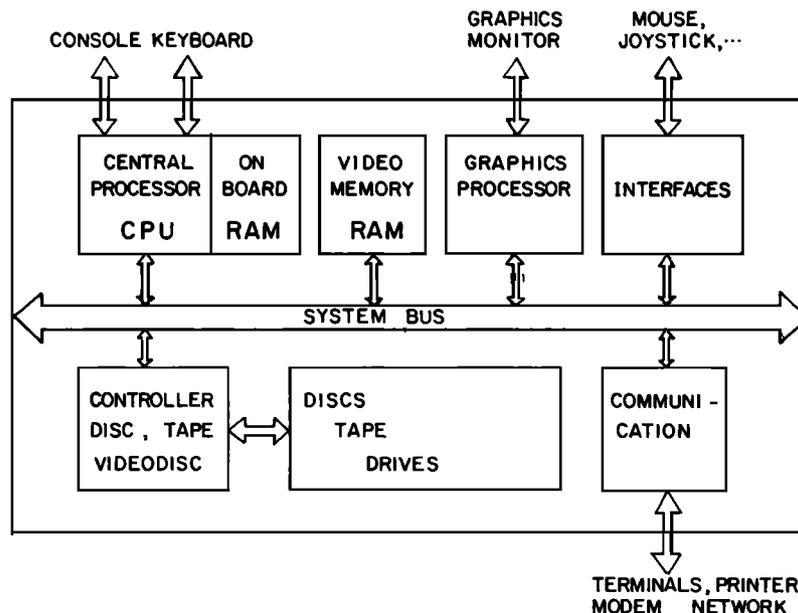


Fig. 3. Computer workstation hardware showing the basic configuration of a small self-contained unit that includes core and disk storage, a central processor, and various input and output (display) devices.

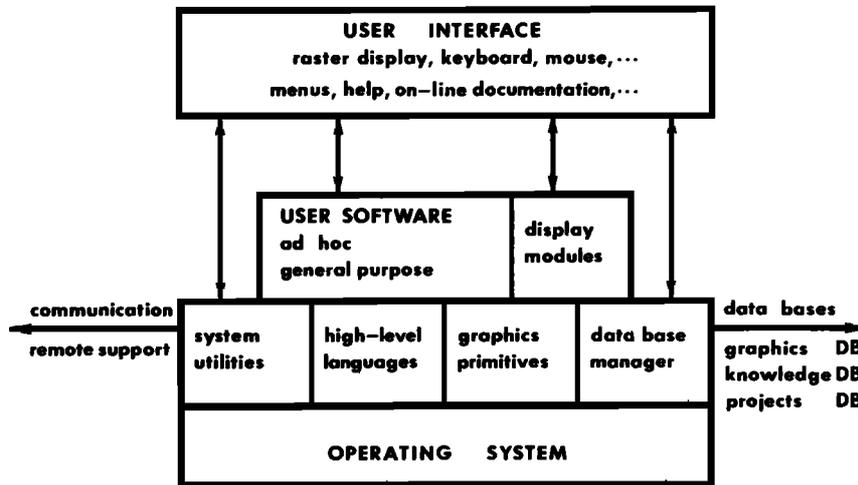


Fig. 4. Computer work station software showing the integration of operating software with application software and the user interface.

[Papert, 1980]. Obviously, many simulation or optimization models are constrained by the requirements of analytical solutions, by the ease and (mathematical) elegance of analysis rather than by the problem they purport to solve. Real-world problems are best described as “a mess” and cannot necessarily be represented by, say, linear algebra or Taylor-series expansions. For the description of real-world problems, one would have to use many more conditional constructs (with a finite and usually small set of possible conditions) representing rules, experience, and expertise, rather than (differentiable) functional relationships. These rules are of the If ... Then or While or For Each ... Do types, and are basic elements in the Artificial Intelligence field [Charniak et al., 1980].

Elements of heuristic programming allow for the incorporation of empirical knowledge and expertise, however tentative. Usually, this kind of knowledge is not only more readily available, but also more relevant as a direct reflection of the problem. The usual differentiable functional relationship, in contrast, introduces an arbitrary pseudoprecision. As a rule, we know neither the exact kind of most functional relationships nor the necessary parameters [e.g., Fedra, 1982]. While there is undoubtedly much to be learned from mathematical models and the study of their behavior, translating their results into policy advice is rarely straightforward. In real-world problem solving, usually few if any of the basic assumptions of any

mathematical model will hold. As a consequence, the model and its output, understood as a product, will at best be a distorted, and possibly misleading, approximation. Models are rightly described as caricatures of reality, so their output and answers are necessarily the caricatures of a solution.

An alternative or complementary approach could be called heuristic [Simon and Newell, 1958] and linguistic modeling [Zadeh, 1973]. Heuristic methods concentrate on problem solving and especially the mental operations useful in this process. Heuristics are not necessarily algorithms or effective procedures of computer science, but rather rules of the thumb applied to problems, especially practical problems. The most important heuristic is “the end suggests the means” [Polya, 1971]. Realizing the dominant role of interpretation and judgment in more comprehensive problems and the importance of exploratory or educational, as opposed to engineering or design, applications of models, these models start from the perceptions of a problem rather than from physical, ecological, or economic theory.

Unrestricted by algorithmic constraints, building a model or problem formalization (Figure 5) starts out from a simple narrative description of the problem situation. This is eventually translated into an arbitrary pseudocode that resembles a high-level language like FORTRAN or Pascal but avoids excessive detail by lapsing from time to time into natural

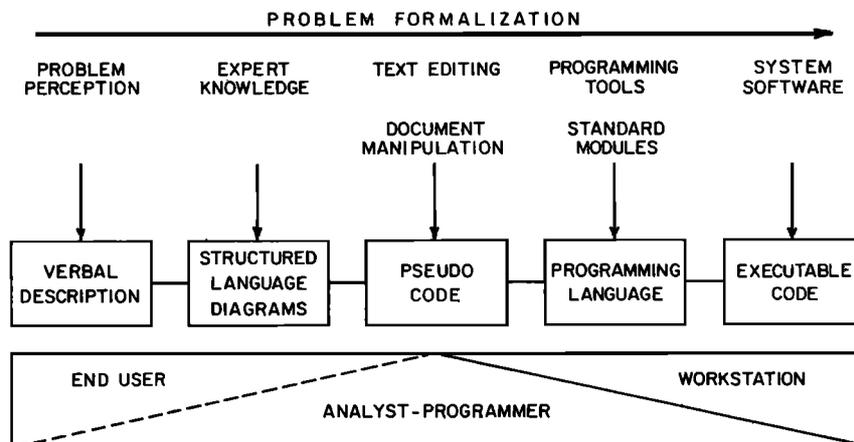


Fig. 5. Idealized process of problem formalization and model development showing the roles of analyst/programmer and end user. The actual process may involve shortcuts and reiterations of various steps.

language [e.g., *Kernighan and Blauger, 1976*]. Typically, such a description will consist of numerous badly defined variables and at best qualitative interactions. However, once a list of variables and interactions is specified, they can be defined, taking recourse to the expertise (and bias) of the people involved. Wherever applicable, scientific knowledge will of course be incorporated, but undisputed scientific evidence is usually found only at a very detailed level and at small scales. Eventually, this information can be extrapolated to larger regional levels. As a rule, however, such extrapolation will likely again require arbitrary assumptions and subjective judgments. The resulting formal descriptions will therefore have to account for all this uncertainty and ambiguity.

Translating the perception of the problem and the set of definitions and descriptions of interactions into an executable code of a computer model should be straightforward. The basic narrative forms the major part of the simulation system. Each of the description of interactions are put into the form of function or subroutine subprograms. The resulting code will be highly redundant, but easy to read, understand, and modify. Of course, each of the systems modules may be fairly sophisticated, particularly so where the effects of uncertainty and ambiguity have to be included. In these situations a variety of techniques such as fuzzy algorithms, conditional statements, and fuzzy classifiers can be used for the description of the soft elements such as many socioeconomic, institutional, or political aspects [*Kickert, 1978*]. Constructs involving stochastic variables or parameters for Monte Carlo simulation can be used to capture risk and uncertainty. Optimization and complex decision routines can be used to possibly represent behavioral aspects. The modeling process thus combines "classical," i.e., theory-based mass and energy conversion models and optimization methods, with elements of heuristic programming similar in concept to rule-based expert systems in Artificial Intelligence (see, for example, *Stansfield [1979]*), but using fuzzy sets instead of symbol lists, allowing for the direct and easy coupling of symbolic and numerical elements [*Fedra, 1983*].

#### Symbolic Interfaces for Man-Machine Interaction

Linguistic, or symbolic modeling can be described as combining the "classical" quantitative forms of description with qualitative, symbolic, natural language elements. The symbolic elements can be represented by lists of atomic symbolic elements, i.e., "names," as in the mainstream of Artificial Intelligence, and represented most clearly in the programming language List Processor (LISP). Alternatively, they can be based on *Zadeh's [1973]* concept of a linguistic variable, represented by fuzzy sets. The latter form of representation provides a convenient format of linking quantitative and qualitative information, and an elaborate set of operations allows conversion from one domain to the other. This is not only instrumental in structuring a quasi-natural language man-machine interface, but it can also be used to efficiently aggregate and summarize information and to couple models of different degrees of resolution and precision. In particular, output from stochastic models, generated in the form of distributions for the variables of interest, can readily be converted or classified into linguistic statements. These distributions can also be interpreted as fuzzy sets and are thus subject to the available set of operations defined for them [*Zadeh, 1973*].

Obviously, there is a close link between heuristic and linguistic approaches to programming: expert knowledge formulated in natural language translates almost directly into

fuzzy algorithms. These, in turn, allow the derivation of approximate solutions to inexactly stated resource management problems, including measures of the solutions precision in terms of linguistic classifiers or hedges, such as "little," "more or less," or "very" (see below).

Linguistic variables, based on fuzzy sets, can very effectively be used for classification and communication. Each of the calculated values of a model output variable can be understood as a sample from a probability distribution of possible outcomes over the ensemble of possible inputs and parameters [*Fedra, 1982*]. This reflects the numerous sources of uncertainty associated with such calculations and is quite obvious in the case of stochastic models. Such a sample is thus associated with a certain variability, or it is generated as a distribution, a set of numbers. By using fuzzy algebra [*Zadeh, 1973*], this translates into the linguistic statement describing the value of the variable in natural language. A simple example might be water quality which, for a given purpose and use, could be described in terms of "good," "fair," "bad," etc. or simply "sufficient" or "insufficient" [*Fedra, 1983*].

Figure 6 illustrates a simple graphical display method, where a model prediction for a continuous water quality variable, for example, represented as a discretized distribution is translated into a linguistic statement. Interpreting the output variable distribution as a fuzzy set, where the normalized frequencies of a given class translate into values of a membership function, the composition of this fuzzy set with the definition of quality classes along the values of the output variable (Figure 6a) results in the membership function or support

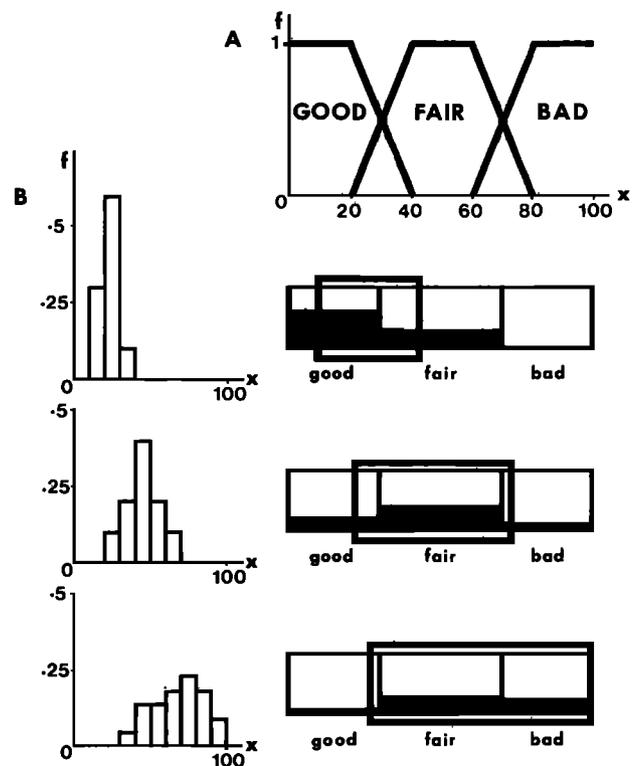


Fig. 6. Water quality classification using fuzzy sets. (a) Fuzzy relationship  $R$  between a continuous water quality variable  $x$  and three water quality classes: good, fair, and bad. (b) Examples of translating the composition of a water quality forecast of increasing uncertainty (expressed as a frequency distribution of the water quality variable  $x$ , interpreted as a fuzzy set) with the fuzzy relation  $R$  into a simple graphical symbol, i.e., the window over the quality scale. Dark areas represent support levels for the individual quality classes.

values for the individual quality classes. This method also conveys some measure of the "imprecision" of the estimate, represented by the growing extent of the window over the three-part scale (Figure 6b).

The use of linguistic variables serves several purposes:

1. It provides an easy to understand user interface for which the decision maker can input his or her subjective judgments and develop an intuitive feeling, which can be merged into a set of mental models.

2. It interfaces nicely with stochastic simulation techniques. When the variable comes as a discretized distribution from Monte Carlo models [Fedra, 1982], this distribution can again be described by a fuzzy set. With a fuzzy relationship between the variable and the evaluation or quality class, the support levels or values of the membership functions for the individual quality classes can be computed [Jowitt and Lumbers, 1982]. The result can then be displayed by indicating the respective verbal classification, a range of classes [e.g., good to moderate], or by using a simple graphical display method [Fedra, 1983].

3. It can easily be backed up by a more elaborate description in terms of probability distributions (displayed graphically) when requested.

4. It can be quite easily calibrated, even if there is only partial consensus in a group of people; in fact, it can be understood as representing partial consensus.

5. The linguistic statements can easily be combined into policy-oriented classifications, conclusions, or fuzzy inferences of the kind: if water quality is fair or better and tourism is constant or slowly increasing, then the policy is a success; otherwise it's a failure. Control and policy variables and the external assumptions in the design of a scenario can be specified in such qualitative and relational terms as "more" investment into wastewater treatment (relative to a basis case); "slowly increasing" prices in the tourism sector; or a "very small increase" in fertilizer usage.

This already sounds very much like the language of a certain level of policymaking. However, now there is a direct link provided to simulation modeling, bridging the gap between the language of the nontechnical user, or the real-world application, and the numbers of the model. Lists of such statements combined through fuzzy relations can define objectives and criteria at a relatively abstract level of conceptualization, appealing from a policy-oriented point of view. At the same time, the direct link to the numerical, strictly quantitative though stochastic methods of simulation and analysis is maintained. Obviously, this can be used for the screening of very large numbers of alternatives, as a preprocessor for the interactive approach.

Also, placing much of the critical ambiguity in the user interface is probably the most valuable aspect of the whole approach. There, in the user interface, ambiguity is obvious, easy to inspect and criticize, and it provokes judgment. Linguistic variables are certainly a very effective vehicle for communication, and their symbolic nature allows them to be readily translated into graphical symbols (e.g., color or video attributes like intensity or blinking) as well. All these elements may be used and combined freely and without any restrictions, wherever deemed necessary. Models are structured entirely as a result of recursively formalizing the perception(s) of a problem; as a consequence, this process is iterative, and several rounds of discussion may be required before a consistent model can be built.

There are several important advantages of the approach

just described. First, since the model description starts out in a common language narrative form, the code will already contain its documentation, assuming we use the computer to organize the information to be structured from the very beginning. The original textfiles containing the first description of the systems elements are the comments of the final program. More importantly, the method allows combining the classical, scientific-type of relevant information, i.e., quantitative data, with the much larger body of "soft" qualitative information, expertise, and experience. For most problem situations there exists a body of qualitative knowledge, of expertise, and experience, far in excess of those few quantitative measurements and observations that may be available.

The challenge, clearly, is to include this qualitative type of useful and valuable information and combine it with the more traditional type of quantitative data. One possibility is by using verbal classification within a normalized domain of a variable of interest and use fuzzy set operations to derive an approximate estimate. For example, the relationship between, say, lake water quality and the number of tourists to be expected on the beaches is certainly fuzzy and at best only approximately understood [Feenberg and Mills, 1980]. We can now define a multiplier "effect," normalized to the range of zero to one, to express the adverse effects of poor water quality on the number of visitors. Expert opinion or user interaction is used to "calibrate" the relationship, i.e., define the support values for the individual effect classes and the water quality-effect relationship.

The method makes well-structured and modular programming very easy, and most economic. The top-down approach lends itself most naturally to a high degree of structuring [e.g., Stevens et al., 1974]. Due to its modularity and self-contained full documentation, the code is easy to maintain and easy to modify. The latter is certainly most important, since frequent redefinitions of the problem are a common symptom of progressive learning. As a consequence, most of the code is of an ad hoc nature. However, the modular scheme also facilitates the use of prefabricated tools and elements (e.g., Myer [1975]; Kernighan and Blaugar [1976]; Loucks et al. this issue(b), Figures 3 and 6).

#### WORK STATION ARCHITECTURE

The above outline suggests the combination of numerous and various tools for analysis, realizing that no single approach or technique can satisfactorily cope with the richness and variety of most real-world problems. This multitude of tools requires a common organizing framework, implemented on the computer, and provided by the software development support environment of modern operating systems [e.g., Chesson, 1975; Dolotta et al., 1978; Ritchie and Thompson, 1978]. Such a highly modular approach again has to be highly structured [e.g., Wirth, 1973; Stevens et al., 1974] and hierarchically organized, allowing for compatible data bases and a smooth transition from one model to another or from one level of detail to another. Even high levels of complexity, thus organized, can be made manageable [Myer, 1975; Kernighan and Blaugar, 1976].

Hierarchical structuring with a common denominator also allows for the selective detailed examination of any element, sector, or region, while the remainder of the overall system is treated at the level of minimum resolution. Such functional or spatial "zooming capabilities" makes the approach extremely flexible when working with a diverse group of users representing various sectors and regions related in a certain prob-

lem. While maintaining the framework and thus feedback of the overall system, much detail for selected elements can be obtained without resorting to immense model systems with all their technical and communicational constraints [Kleijnen, 1976].

Consider a comprehensive study like the more recent Policy Analysis of Water Management for the Netherlands (PAWN) [Goeller *et al.*, 1983]; the range of problems addressed is prototypical for many more comprehensive policy analyses, although the subproblems and the emphasis will of course be different for different regions. For each of the problems considered there exist several approaches and in fact models, since all these problems, with some generalization, are more or less generic problems. By standardizing the input-output structure of these models, but without too much affecting their inner workings, they can be transformed into compatible, exchangeable, and readily available building blocks. A common modeling framework, based on numerous interface routines that connect the individual models, data bases, and the user interface, allows one to combine models of vastly different structure, emphasis, dimensionality, and resolution. Thus the vast body of available software can be used and incorporated into a comprehensive analysis.

To be useful, such a collection of building blocks has to be well organized. It is essential to have an on-line catalog of available software modules, including a detailed description of these program elements, that can be searched by key words. The importance of such well-organized model documentation is well recognized, although systematic attempts at model comparison and consistent documentation are rare [e.g., Bachmat *et al.*, 1980]. In a standardized framework for model development, or rather synthesis, consistent style is an important element of user friendliness.

#### A Display-Oriented User Interface

We have already commented about the way models are often built with a conservative bias and that this bias is the result of past limitations in hardware and software. These limitations no longer exist. The same holds true for the way many people interact with computers [e.g., Kay, 1977; Guedj *et al.*, 1980]. The standard user interface seems to be designed only for teletypes. This punch card-oriented style of communication, restricted to alphanumeric formats, geared toward the batch-processing environment of the past, is hardly suitable for a truly interactive approach and dedicated work stations.

Bit-mapped graphics systems with multiple window capabilities or more than one parallel output device allow the structuring of complex displays. This can greatly increase the amount of information communicated and at the same time also enhance the ease of understanding [Meyrowitz and Moser, 1981] (e.g., Figure 7). Mixtures of alphanumeric, symbolic, and graphical elements, using such familiar backdrops as maps or flow-chart representations of systems, can be very effective; they do, however, require a considerable amount of design effort [Loucks *et al.*, this issue(b)]. Consequently, in designing the model representations, the style of the display, or the visual part of the user interface, has to be considered from the very beginning. Again, hierarchical structures are essential and well supported by, e.g., the Association for Computing Machinery (ACM), Special Interest Group on Graphics (SIGGRAPH), Core or Graphic Kernel System (GKS), Graphics Standard [e.g., Foley and Van Dam, 1982].

As an example, consider a regional map which for each

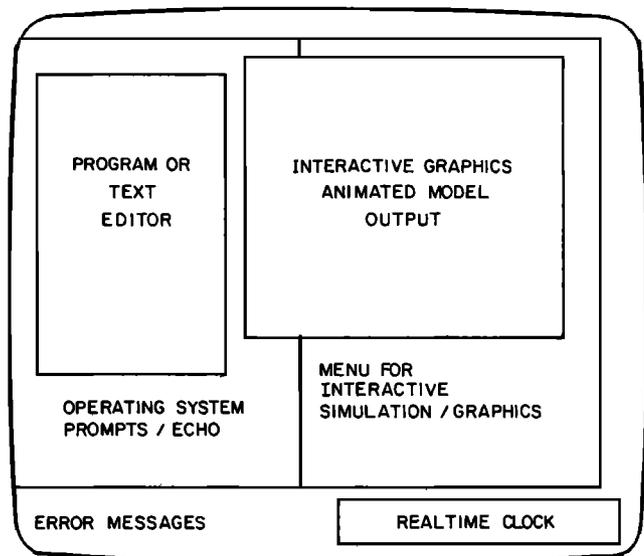


Fig. 7. Multiple window display showing possible organization of a single video screen. Six partly overlapping windows correspond to various processes, levels of interaction, and operator information. Alternatively, more than one display unit can be used.

community contains certain symbols (such as pie or bar charts) for each major source of pollution, wastewater treatment facility, or point of use. These symbols, together with identifying text, indicate the status of these nodes [Fedra, 1983, this issue]. They are selectively modified either during editing of the policy variables or the simulation of the system's behavior. With the basic background map created within one window (there could be more than one window on the screen), and placing each community's set of symbols into a segment, the individual communities can now be selectively updated in random sequence and arbitrary scales, using a generalized loop structure which only has to be supplied the appropriate segment identification.

The new paradigm of man-machine interaction is based on personal, i.e., nonshared, micro/minicomputers and interactive graphics as the standard means for user interaction. It offers a multitude of "windows" corresponding to multiple simultaneous activities [e.g., Yenarichs, 1982]. The contents of these windows or virtual terminals each reflect the progress of some activity like editing a file, output from a simulation program, or the result of a data base search. Obviously, these parallel processes allow a much broader bandwidth of interaction than traditional methods. The ease and speed with which the user can manipulate the parallel executing processes and their corresponding windows is a major advantage over more traditional time-sharing techniques. The inherent multidimensionality and multiple-process orientation free the user from the restrictions of linear, sequential textual communication inherited from the batch environment, its tools and languages. Examples of advanced user interfaces are given in the works by Bobrow [1968], Bolt [1979, 1980], Herot *et al.* [1980], or Meyrowitz and Moser [1981].

#### IMPLEMENTATION

Clearly, to be acceptable and attractive the user interface must talk, and understand, the language of the users. If the user is not the same person that put the software together, which will rarely be the case, a close cooperation is required. Many of the disciplines involved in environmental planning

have their own very specific jargon. Since these technical languages do vary from group to group, it is essential to be flexible in the language-oriented part of the user interface, allowing for easy modifications of "translations"; users should not be forced to learn a new language to interact with the machine and in particular not any highly formal symbolic languages.

The software interface needs to be customized, designed for, with, and sometimes even by the user (see Figure 6). The approach, using a set of tools in a compatible framework, is customized by applying it to a given problem, and within the context of a given institutional structure. To allow such a close and truly interactive operation, easy access is a key provision. In practice, it means putting the machine together with the necessary software right on the user's desk and writing software for maximum user ease.

This new paradigm envisions an integrated man-machine system [compare *Sheridan and Ferrel*, 1981], where instead of learning to fly an aircraft or to control a nuclear power plant the user learns to manage the environment with the help of an appropriate computer-driven simulator. The role of the analyst thus changes considerably from what it has been in the past. Instead of delivering the usual set of hefty volumes of reports as the result of an assignment, the new "product" is now an integrated tool kit, together with the on-the-job training in its use. Although this approach is demanding and may require considerable user involvement, it assures a useful product. The design, implementation, training, and first application of the hardware-software combination are inseparable elements of this new analyst-client relationship. What we are proposing and working on ourselves [*Fedra*, 1983; *Loucks et al.*, this issue(a, b)] seems worth reiterating. Instead of delivering "the optimal" solution or a set of solutions to choose from, what is delivered is a customized set of tools and techniques that will allow the user to define, analyze, synthesize, and evaluate solutions to his or her problems. No longer is the client required to specify and precisely quantify his criteria a priori; he or she can explore the problem and develop objectives and criteria as part of the ongoing learning process. The methods of analysis are now made part of the decision-making process, within the context of the institutional structure in which decisions are made.

#### DISCUSSION

Environmental engineers have always been very much concerned, and properly so, with identifying and designing one or more preferred alternatives. The main questions often asked have been how to do things efficiently. However, as much as we must continue to focus on how to do something best, we must increasingly ask, "What should we do, and why?". It is the search for an enlarged set of alternatives rather than the optimization of a few more or less successful ones that becomes increasingly important. This shift in emphasis also causes a shift in the relevant objectives. The softer the problem, the more important are perceptions and values, the more emphasis is on the political and social versus the technical and engineering aspects of the problem.

The recognition of genuinely conflicting but legitimate interests in coexisting elements of society or societies as a whole is surely a precondition to conflict resolution and compromising. What we are proposing is a framework for analysis and communication that facilitates the sorting out of points of difference over semantic questions by using a common system of reference, a common meta language, display oriented and

composed of numerical quantitative and symbolic qualitative elements. The agreement to disagree about certain values, objectives, or criteria relevant to a problem situation, and a conflict of interest is thus the first step toward conflict resolution. A common set of knowledge describing the problem situation is again a necessary prerequisite for that. Providing and enlarging this common knowledge base is the primary objective of our modeling efforts.

It seems worthwhile to reiterate the potential role of the interactive computer work station concept in this setting: it is meant to support the planning and decision-making process, not to replace it. This support consists of making available relevant information on the environmental and technological systems in question, and the likely consequences of any action or policy considered, in a fast, reliable (at least in the sense of repeatable and open for criticism), and easy to comprehend way. The work station can free planners and policymakers from the laborious and often disruptingly time-consuming tasks of noninteractive data manipulation and analysis. The easy and fast organization of information and evidence should allow for a more creative, playful, and brainstorming atmosphere. An attractive, powerful, and responsive tool should invite and stimulate a more experimental, innovative attitude of "anything goes" in the sense of *Feyerabend's* [1978] criticism of methodological constraints. The basic idea is to provide a tool that will permit a dramatic increase in the number of alternatives that can be examined, and in fact, first of all, to be invented. It can trace the consequences deemed important by simulation at an otherwise prohibitive level of detail or breadth and coverage. Finally, it can assist informed judgment by appropriately aggregating and displaying alternatives to be compared.

The basic elements of the work station concept are a system of data bases, including a "knowledge" data base, that contain ecological and engineering or economic handbook-type information; project data bases with all the project-specific information from meteorological records to regional statistics, maps, engineering drawings, etc.; and simulation data bases that handle the modeling results. Application programs constitute a kit of tools for a flexible tinker toy approach to systems modeling, simulation, and optimization. These tools are hierarchically organized at various levels of aggregation. Finally, the toolbox has to contain assorted methods for evaluation and display, i.e., organizing the information available and generated during the analysis under interactive control. The work station concept is supposed to provide an educational framework, a LOGO-turtle environment [*Papert*, 1980] for environmental systems analysis.

An important element is the direct interaction. What we are proposing is the development and the transfer of tools and skills rather than "solutions." We want to build the modeling approach into the decision-making process and its institutional framework. This will require close attention to the importance of customized design, on-site implementation, on-the-job training, and continuing support and maintenance.

"Our society's growing reliance on computer systems that were initially intended to 'help' people make analyses and decisions, but which have long since both surpassed the understanding of their users and become indispensable to them, is a very serious development. ... decisions are made with the aid of, and sometimes entirely by, computers, whose programs no one any longer knows explicitly or understands. Hence no one can know the criteria or the rules on which such decisions are based" [*Weizenbaum*, 1976].

That situation need not be the case. The way we think models of even complex systems should be constructed, using relatively simple modules, devising a well-structured system of the modules and thoroughly documenting the model system, will help in effective model use. Remember that there are numerous examples of situations not involving computers at all, where decisions are made based on rules that are not known explicitly or fully understood. Most instinctive behavior falls into this category.

In complex problem situations it is next to impossible to sort out the effect of any individual decision from among the multitude of confounding influences. The success and effectiveness of computer technology and any modeling approach can only be described in terms of its use, its acceptance, and contribution in the practice of planning and decision making. The measure of success is the contribution to a learning process, stimulating the introduction of new concepts and points of view and new perceptions of the problems.

#### REFERENCES

- Anonymous, Robots take Detroit, *Automot. Ind.*, 162(1), 1984.
- Bachmat, Y., J. Bredehoeft, B. Andrews, D. Holtz, and S. Sebastian, *Groundwater Management: The Use of Numerical Models*, Water Resour. Monogr. Ser., vol. 5, pp. 1-27, AGU, Washington, D. C., 1980.
- Belshaw, P. N., Microcomputers and management science, *Interfaces*, 12(6), 105-106, 1982.
- Bobrow, G. D., Natural language input for a computer problem solving system, in *Semantic Information Processing*, edited by M. Minsky, pp. 1-440, MIT Press, Cambridge, Mass., 1968.
- Bolt, R. A., Spatial data-management, report to DARPA, MIT Architect. Mach. Group, Cambridge, Mass., 1979.
- Bolt, R. A., 'Put-that-there:' Voice and gesture at the graphics interface, *Comput. Graphics*, 14(3), 262-270, 1980.
- Branscomb, L. M., Electronics and computer: An overview, *Science*, 215(4534), 755-760, 1982.
- Brennan, J. J., and M. K. Molloy, Microcomputers, *Interfaces*, 13(1), 28-39, 1983.
- Brown, G. E., Jr., Can systems analysis and operations research help Congress?, *Interfaces*, 12(6), 119-125, 1982.
- Charniak, E., C. K. Riesbeck, and D. V. McDermott, *Artificial Intelligence Programming*, pp. 1-323, Lawrence Erlbaum Associates, Hillsdale, N. J., 1980.
- Chesson, G. L., The network UNIX system, *Oper. Syst. Rev.*, 9(5), 60-66, 1975.
- Dolotta, T. A., R. C. Haight, and J. R. Mashey, UNIX time-sharing system: The programmer's workbench, *Bell Syst. Tech. J.*, 57(6), 2177-2200, 1978.
- Fedra, K., Environmental modeling under uncertainty: Monte Carlo simulation, *Rep. RR-83-28*, Int. Inst. for Appl. Syst. Anal., Laxenburg, Austria, 1983.
- Fedra, K., Interactive water quality simulation in a regional framework: A management oriented approach to lake and watershed modeling, *Ecol. Model.*, 21, 209-232, 1983.
- Fedra, K., A modular interactive simulation system for eutrophication and regional development, *Water Resour. Res.*, this issue.
- Feenberg, D., and E. Mills, *Measuring the Benefits of Water Pollution Abatement*, pp. 1-181, Academic, New York, 1980.
- Feyerabend, P., *Science in a Free Society*, pp. 1-221, New Left Books, London, 1978.
- Foley, J. D., and A. Van Dam, *Fundamentals of Interactive Computer Graphics*, pp. 1-664, Addison-Wesley, Reading, Mass., 1982.
- Friedman, R. M., *The Use of Models for Water Resources Management, Planning and Policy*, Office of Technology Assessment, U.S. Congress, Washington, D. C., August, 1982.
- Geoffrion, A. M., Can MS/OR evolve fast enough?, *Interfaces*, 13(1), 10-25, 1983.
- Goeller, B. F., Policy analysis of water management for the Netherlands—Volume I, Summary report, *Rep. RAND/R-2500/1-NETH*, Rand Corp., Santa Monica, Calif., 1983.
- Gomory, R. Y., Technology development, *Science*, 220, 576-580, 1983.
- Guedj, R. (Ed.), *Methodology of Interaction*, pp. 1-410, North-Holland, Amsterdam, 1980.
- Herot, C. F., et al., A prototype spatial database management system, *Comput. Graphics*, 14(2), 63-70, 1980.
- Jowitt, P. W., and J. P. Lumbers, Water quality objectives, discharge standards and fuzzy logic, Optimal allocation of water resources, *Proceedings of the Exeter Symposium*, July 1982, *IAHS Publ. 135*, IAHS, Washington, D. C., 1982.
- Kay, A. C., Microelectronics and the personal computer, *Sci. Am.*, 237(3), 230-244, 1977.
- Kernighan, B. W., and P. J. Blauger, *The Elements of Programming Style*, pp. 1-168, McGraw-Hill, New York, 1978.
- Kernighan, B. W., and P. J. Blauger, *Software Tools*, pp. 1-338, Addison-Wesley, Reading, Mass., 1976.
- Kickert, W. J. M., *Fuzzy Theories on Decision Making*, pp. 1-181, Martinus Nijhoff Social Sciences Division, Leiden, The Netherlands, 1978.
- Kleijnen, J. P. C., Computers and operations research: A survey, *Comput. Oper. Res.*, 1, 327-335, 1976.
- Loucks, D. P., J. R. Stedinger, and U. Shamir, Research in water resources and environmental systems modeling, *Nat. Res. Forum*, 8(3), 219-240, 1984.
- Loucks, D. P., J. Kindler, and K. Fedra, Interactive water resources modeling and model use: An overview, *Water Resour. Res.*, this issue(a).
- Loucks, D. P., M. R. Taylor, and P. N. French, Interactive data management for resources planning and analysis, *Water Resour. Res.*, this issue(b).
- McGowan, R. P., The use of computers in government: A look at municipal fire services, *Comput. Environ. Syst.*, 7, 155-162, 1982.
- Myer, G. J., *Reliable Software Through Composite Design*, pp. 1-159, Petrocilli/Charter, New York, 1975.
- Myers, W., CAD/CAM, The need for a broader focus, *Computer*, 15(1), 105-117, 1982.
- Meyrowitz, N., and M. Moser, BRUWIN: An adaptable design strategy for window manager/virtual terminal systems, paper presented at the Proceedings of the 8th Annual Symposium on Operating Systems Principles (SIGOPS), Pacific Grove, Calif., Dec. 1981.
- Papert, S., *Mindstorms, Children, Computers, and Powerful Ideas*, pp. 1-230, Basic Books, New York, 1980.
- Polya, G., *How to Solve It*, pp. 1-253, Princeton University Press, Princeton, N. J., 1971.
- Riley, M. J., *Management Information Systems*, 2nd ed., pp. 1-425, Holden Day, Oakland, Calif., 1981.
- Ritchie, D. M., and K. Thompson, The UNIX time-sharing system, *Bell Syst. Tech. J.*, 57(6), 1905-1929, 1978.
- Rogers, P. O., Urban planning policy analysis and administration, *Discuss. Pap. D79-4*, Dept. of City and Reg. Plann., Harvard Univ., Cambridge, Mass., 1979.
- Sebestyen, I., The videodisc revolution, *Rep. WP-81-160*, Int. Inst. for Appl. Syst. Anal., Laxenburg, Austria, 1981.
- Sheridan, T. B., and W. R. Ferrel, *Man-Machine Systems: Information, Control, and Decision Models of Human Performance*, pp. 1-452, MIT Press, Cambridge, Mass., 1981.
- Simon, H. A., and A. Newell, Heuristic problem solving: The next advance in operations research, *Oper. Res.*, 6, 8, 1958.
- Sprague, R. H., and E. D. Carlson, *Building Effective Decision Support Systems*, Prentice-Hall, Englewood Cliffs, N. J., 1982.
- Stansfield, J. L., Developing support systems for information analysis, in *Artificial Intelligence: An MIT Perspective*, edited by P. H. Winston and R. H. Brown, pp. 288-314, MIT Press, Cambridge, 1979.
- Stevens, W. P., G. J. Myers, and L. L. Constantine, Structured design, *IBM Syst. J.*, April, 1974.
- Weizenbaum, J., *Computer Power and Human Reason*, pp. 1-30, W. H. Freeman, San Francisco, Calif., 1976.
- Winston, P. H., and R. H. Brown (Eds.), *Artificial Intelligence: An MIT Perspective*, vol. 1, pp. 1-492, MIT Press, Cambridge, Mass., 1980.
- Wirth, N., *Systematic Programming*, pp. 1-169, Prentice-Hall, Englewood Cliffs, N. J., 1973.
- Yenarichs, L., Menu driven real-time system relieves data acquisition headaches, *Electron. Prod. Mag.*, 1, 49-53, 1982.
- Zedah, L. A., Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. SMC-3*, 1, 28-44, 1973.

K. Fedra, International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria.

D. P. Loucks, School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853.

(Received March 17, 1984;  
accepted March 21, 1984.)