SMART SOFTWARE FOR WATER RESOURCES PLANNING AND MANAGEMENT

Kurt Fedra
Advanced Computer Applications
International Institute for Applied Systems Analysis
A-2361 Laxenburg, Austria

ABSTRACT. Water resources planning and management has a considerable history in the use of computers. By covering a problem domain that combines a solid foundation in the physical sciences, and a large amount of data, with strong elements of socio-economic and political considerations, water resources planning and management is also an ideal application area for the latest advanced information technology. Advanced information technology provides the tools to design and implement *smart* software where, in a broad sense, the emphasis is on the man-machine interface. Integration, interaction and visualization are three key concepts that are discussed in detail, using a number of operational software examples.

Integration implies that in any given software system for real-world applications, more than one problem representation form or model, several sources of information or data bases, and a multi-faceted, problem-oriented user interface ought to be combined in a common framework to provide a useful and realistic information base. Integration also means that tools such as expert systems components can be embedded in, for example, simulation models and user interfaces in general, providing the means for smart systems behavior. The increasing complexity of such composite tools, in turn, requires a well-structured and modular approach, supported by technological developments such as powerful workstations supporting bit-mapped color graphics and cost-effective high-volume mass storage, distributed computing and networking support, or object-oriented software.

Interaction is a central feature of any effective man-machine system: a real-time dialogue allows the user to define and explore a problem incrementally in response to immediate answers from the system; fast and powerful systems with modern processor technology can offer the possibility to simulate dynamic processes with animated output, and they can provide a

NATO ASI Series, Vol. G 26
Decision Support Systems
Edited by D. P. Loucks and J. R. da Costa

high degree of responsiveness that is essential to maintaining a successful dialogue and direct control over the software. Obviously, raw computer power, but also the availability of appropriate input and output devices such as high-resolution color screens or mouse pointers, are key requirements here.

Visualization provides the band-width necessary to understand large amounts of highly structured information, and permits the development of an intuitive understanding of processes and interdependencies, of spatial and temporal patterns, and complex systems in general. Many of the problem components in a real-world planning or management situation are rather abstract: representing them in a symbolic, graphical format that allows visual inspection of systems behavior and, in general, symbolic interaction with the machine and its software, is an important element in friendly and easy-to-use computer-based systems.

Several examples of smart software systems are introduced below, ranging from surface to groundwater flow and transport models, and finally rule-based or expert systems for water resources applications. Some basic principles and building blocks of smart software are discussed with these examples.

INTRODUCTION

Software and computer-based tools are designed to make things easier for the human user. In practice, only very few programs do that. They make things possible that would not be possible without the computer, but they rarely make it easy on the user. Smart software is designed to do just that: make things easy for the user. Rather than adding yet another level of complication, it should free the user from the more mundane, time-consuming, as well as error-prone tasks associated with data processing and analysis, such as compiling and formatting information, ensuring completeness, consistency, and plausibility of inputs, and then extracting the relevant part of the output from the often huge amounts that computer programs tend to generate.

There are several important aspects that need to be addresses. First, there is a tradeoff between the efficiency and ease of use and the flexibility of a system. The more options