A HYBRID APPROACH TO INFORMATION AND DECISION SUPPORT SYSTEMS:
Hazardous Substances and Industrial Risk Management

K. Fedra, E. Weigkricht, and L. Winkelbauer
International Institute for Applied Systems Analysis
Laxenburg, Austria

RR–87–12
June 1987

Preprinted from *Economics and Artificial Intelligence* (forthcoming),
Pergamon Books Ltd.

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
Laxenburg, Austria

Preprinted with permission from *Economics and Artificial Intelligence* (forthcoming), Pergamon Books Ltd., pp. 169–175.

Printed by Novographic, Vienna, Austria

FOREWORD

Applied systems analysis is – or should be – a tool in the hands of planners and decision makers who have to deal with the complex and growing problems of modern society. There is, however, an obvious gap between the ever-increasing complexity and volume of scientific and technological information and tools of analysis relevant to large socio-technical and environmental systems, and the information requirements at a strategic planning and policy level.

The Advanced Computer Applications (ACA) project builds on IIASA's traditional strength in the methodological foundations of operations research and applied systems analysis, and its rich experience in numerous applications areas including the environment, technology, and risk. The ACA group draws on this infrastructure and combines it with elements of Artificial Intelligence (AI) and advanced information and computer technology. Several completely externally funded research and development projects in the field of model-based decision support and applied AI are currently under way.

As an example of this approach to information and decision support systems, a research and development project sponsored by the CEC's EURATOM Joint Research Center (JRC) at Ispra, Italy, in the area of hazardous substances and industrial risk management, is described in this paper. With the emphasis on a directly understandable problem representation, and the user interface as a key element of interactive decision support systems, it is a step toward increased direct practical usability of IIASA's research results.

BORIS SEGERSTAHL
Deputy Director
International Institute for Applied Systems Analysis

A HYBRID APPROACH TO INFORMATION AND DECISION SUPPORT SYSTEMS: HAZARDOUS SUBSTANCES AND INDUSTRIAL RISK MANAGEMENT

K. Fedra, E. Weigkricht and L. Winkelbauer

International Institute for Applied System Analysis (IIASA), A-2361 Laxenburg, Austria

Abstract: This paper describes a large-scale, model-based decision support system with embedded Artificial Intelligence (AI) technology, and a largely symbolic color graphics user interface. Designed for industrial risk assessment and the management of hazardous substances, its primary purpose is to allow the efficient use of methods of analysis and information management and to provide a powerful tool in the hands of planners, managers, policy and decision makers. This new generation of model-based decision support system should lead to a more informed, structured, comprehensive and interdisciplinary management of hazardous substances.

To facilitate access for the non-technical user, and for more experimental and explorative use, it proved necessary to build much of the accumulated knowledge on the subject areas into the user interface. Thus, the interface incorporates elements of knowledge-based expert systems that are capable of assisting any non-expert user to select, set up, run, and interpret complex technical software. By providing a coherent, integrated user interface, the interactions between different models, their data bases, and software for display and analysis become transparent and the substantive information basis for decision support can be considerably enlarged.

As a central component of this hybrid decision support tool, a heterarchical, object-oriented information system on hazardous substances has been incorporated. It provides aggregated as well as detailed information for non-specialist users and allows easy modular updating with automatic consistency checking. To cover all the different viewpoints under which one can group hazardous substances (e.g., chemical compounds, toxicity, etc.) an object-oriented heterarchy of viewpoints, groups, and subgroups, down to the elementary substances, has been set up. Each object is activated when accessed, provides its descriptive data, refers to and derives data from other objects. In addition, every object statically inherits all the descriptive data and activities from all its predecessors in the heterarchy.

Keywords: AI, expert systems, model-based decision support, knowledge engineering, heterarchical frame-based information system, industrial risk assessment, hazardous substances management

Expert Systems and Decision Support

Underlying the concept of decision support systems in general, and expert systems in particular, is the recognition that there is a class of (decision) problem situations that are not well understood by the group of people involved. Such problems cannot be properly solved by a single systems analysis effort or a highly structured computerized decision aid (Fick and Sprague, 1980). They are neither unique – so that a one-shot effort would be justified given the problem is big enough – nor do they recur frequently enough in sufficient similarity to subject them to rigid mathematical treatment. Due to the mixture of uncertainty in the scientific aspects of the problem, and the subjective and judgmental elements in its socio-political aspects, there is no wholly objective way to find a best solution.

One approach to this class of under-specified problem situations is an iterative sequence of systems analysis and learning generated by (expert or decision support) system use. This should help shape the problem as well as aid in finding solutions. Key ingredients, following Phillips (1984), are the *Problem Owners*, *Preference Technology* (which helps to express value judgements, and formalize time and risk preferences, and tradeoffs amongst them), and *Information Technology*, (which provides substantive background information, data, and models).

There is no universally accepted definition of *decision support systems*. Almost any computer-based system, from data base management or information systems via simulation models to mathematical programming or optimization, could support decisions. The literature is overwhelming (see Fedra and Otway, 1986, for a recent discussion). Approaches range from rigidly mathematical treatment, to applied computer sciences, management sciences, or psychology. Decision support paradigms include *predictive models*, which give unique answers but with limited accuracy or validity. *Scenario analysis* relaxes the initial assumptions by making them more conditional, but at the same time more dubious. *Normative models* prescribe how things should happen. Alternatively, *descriptive or behavioral models* supposedly describe things as they are.

Most recent assessments of the field, ana in particular those concentrating on more complex, ill-defined, policy-oriented and strategic problem areas, tend to agree on the importance of interactiveness and the direct involvement of the user, resulting in new layers of feedback structures. The *information system model* is based on a sequential structure of analysis and decision support. In comparison, the *decision support model* implies feedbacks from the applications, e.g., communication, negotiation, and bargaining onto the information system, scenario generation, and strategic analysis.

Often enough, however, the problem holder (e.g., a regulatory agency) is not specialized in all the component domains of the problem (e.g., industrial engineering, environmental sciences, toxicology, etc.). Expertise in the numerous domains touched upon by the problem situation is therefore as much a bottleneck as the structure of the decision problem. Building human expertise and some degree of intelligent judgement into decision supporting software is one of the major objectives of AI. Only recently, the area of *expert systems* or *knowledge engineering* has emerged as a road to successful applications of AI techniques (e.g., Fedra & Otway, 1986).

The system discussed here combines several methods of applied systems analysis, operations research, planning, policy sciences, and AI into one integrated software system (Fedra, 1985, 1986) and provides direct and easy access to these largely formal and complex methods to a broad group of users.

A Hybrid Approach to Model-based Decision Support

The software system described here envisions technical experts as its users, as well as decision and policy makers, and in fact, the computer is seen as a mediator and translator between expert and decision maker, between science and policy. The computer is thus not only a vehicle for analysis, but even more importantly, a vehicle for communication, learning, and experimentation.

The two basic, though inseparably interwoven elements, are to supply factual information based on existing data, statistics, and scientific evidence, and to trace the likely consequences of new plans.

The selected approach for the design of this software system is eclectic as well as pragmatic. We use proven or promising building blocks, and we use available modules where we can find them. We also exercise methodological pluralism: any "model", whether it is a simulation model, a computer language, or a knowledge representation paradigm is only valid within a small and often very specialized domain. No single method can cope with the full spectrum of phenomena, or rather points of view, called for by interdisciplinary and truly applied science.

Application and problem-oriented rather than methodology-oriented systems are most often *hybrid systems*, where elements of AI technology are combined with more classical techniques of information processing and approaches of operations research and systems analysis. Here traditional numerical data processing is supplemented by symbolic elements, rules, and heuristics in the various forms of knowledge representation.

There are numerous applications where the addition of a quite small amount of "knowledge" in the above sense, e.g., to an existing simulation model, may considerably extend its power and usefulness and at the same time make it much easier to use. Expert Systems are not necessarily purely knowledge driven, relying on huge knowledge bases of thousands of rules. Applications containing only small knowledge bases of at best a few dozen to a hundred rules can dramatically extend the scope of standard computer applications in terms of application domains as well as in terms of an enlarged non-technical user community.

The Problem Area: Management of Hazardous Substances and Industrial Risk

The effective management of hazardous wastes calls for:

- a minimization of waste generation by process modification and recycling;
- the conversion to non-hazardous forms;
- finally, a safe disposal of whatever is left.

In addition to hazardous wastes, there is a large number of commercial products that are also hazardous. Their production, transportation, and use – before they enter any waste stream – is also of concern. Industrial production processes that involve hazardous raw materials, feedstocks, or interim products, which may reach the environment after an accident, causing direct health risks to man, are also considered.

The problems of managing hazardous substances are neither well defined nor reducible to a small set of relatively simple subproblems. They always involve complex tradeoffs under uncertainty, feedback structures and synergistic effects, non-linear and potentially catastrophic systems behavior – in short, the full repertoire of a real-world mess. The classical methods of Operations Research and Control Engineering, that require a complete and quantitative definition of the problem from the outset, are certainly insufficient.

While only the combination of a larger set of methods and approaches holds promise of effectively tackling such problems, the subjective and discretionary human element must also be given due weight. This calls for the direct and interactive involvement of users, allowing them to exert discretion and judgement wherever formal methods are insufficient.

Under contract to the CEC's Joint Research Centre (Ispra), IIASA's Advanced Computer Applications Project is developing an interactive computer-based decision support and information system. Recognizing the potentially enormous development effort required and the open-ended nature of such an undertaking, we propose a well-structured cooperative effort that takes advantage of the large volume of scientific software already available. A modular design philosophy permits the development of individual building blocks, which are valuable products in their own right, and to interface and integrate them in a flexible, easily modifiable framework.

Every scenario for simulation or optimization, defined interactively with this system, must ultimately be assessed, evaluated, and compared with alternatives in terms of a list of criteria. These criteria include economic, technical, environmental, resource use, and finally sociopolitical considerations. Clearly, only a small subset of these criteria may be expressed in numerical terms. Most of them require the use of linguistic variables for a qualitative description. Using fuzzy set theory, qualitative verbal statements can easily be combined with numerical indicators for a joint evaluation and ranking. In the system design, the use of programming languages like LISP or PROLOG gives the user freedom to manipulate symbols and numbers within a coherent framework.

Model Integration and the User Interface

From a user perspective, the system must be able to assist in its own use, i.e., explain what it can do, how it can be done, and where a result comes from. The basic conceptual elements of this menu-driven system are the following:

- *the interactive user interface* that handles the dialog between the user(s) and the machine; this is largely menu-driven, that is, at any point the user is offered several possible actions which he can select from a menu of options provided by the system;
- *a task scheduler or control program*, that interprets the user request – and, in fact, helps to formulate and structure it – and coordinates the necessary tasks (program executions) to be performed; this program contains the "knowledge" about the individual component software modules and their interdependencies;
  the control program can translate a user request into either:
  - a data/knowledge base query;
  - a request for "scenario analysis"

  the latter will be transferred to
- *a problem generator*, that assists in defining scenarios for simulation and/or optimization; its main task is to elicit a consistent and complete set of specifications from the user, by iteratively resorting to the data base and/or knowledge base to build up the *information context or frame* of the scenario. A scenario is defined by a delimitation in space and time, a set of (possibly recursively linked) processes, a set of control variables, and a set of criteria to describe results. It is represented by
- *a set of process-oriented models*, that can be used in either simulation or optimization modes. The results of creating a scenario and either simulating or optimizing it are passed back to the problem generator level through a
- *evaluation and comparison module*, that attempts to evaluate a scenario according to the list of criteria specified, and assists in organizing the results from several scenarios. For this comparison and the presentation of results, the system uses a
- *graphical display and report generator*, which allows selection from a variety of display styles and formats, and in particular enables the results of the scenario analysis to be viewed in graphical form. Finally, the system employs a
- *system's administration module*, which is largely responsible for housekeeping and learning: it attempts to incorporate information gained during a particular session into the permanent data/knowledge bases and thus allows the system to "learn" and improve its information background from one session to the next.

It is important to notice that most of these elements are linked recursively. For example, a scenario analysis will usually imply several data/knowledge base queries to provide the frame and necessary parameters transparently. Within each functional level, several iterations are possible, and at any decision breakpoint that the system cannot resolve from its current goal structure, the user can specify alternative branches to be followed.

The system must, however, on request "explain" where a result comes from and how it was derived, e.g., from the data base, inferred by a rule-based production system, or as the result of a model application.

The simulation models of the production system can be configured to describe the comprehensive life-cycle of hazardous substances. The major components of the simulation system are (Figure 1):

- the industrial production sector,
- use and market,
- waste management, including treatment and disposal,
- the cross-cutting transportation sector, and
- man and the environment.

Each of these major components is represented by several individual models, covering a variety of possible approaches and levels of resolution. Each element of the simulation system can be used in isolation, or it is linked with several others as pre- or post-processors into increasingly larger (sub)systems models. None of the complexities of the system's integration are obvious to the user: the style of the user interface and interactions with the system are always the same at the user end.

AI technology is embedded into this integrated software system at various levels, in several modules. They range from heterarchical, frame-based data bases(see next section) to rule-based pre-processors and input generators for classical numerical simulation models, rule-based heuristic feasibility and consistency checking of interactive input, to symbolic simulation and intelligent parsers for language-oriented input. The emphasis, clearly, is on a broad set of problem- and knowledge-representation techniques, integrated into one coherent framework.

Heterarchical, Object-oriented Information System on Hazardous Substances

Whenever any of the simulation models is used, it is used for a given substance, substance group, or mixture of substances and substance groups. The classification of substances and substance groups, and the linkage between these groups and the physical, chemical, and toxicological properties of the substances are of critical importance.

With about 70,000 to 100,000 chemical substances on the world market, and about 1000 added to this list every year, any attempts at a complete or even comprehensive coverage within the framework of this project are illusory. Rather, we must provide information about a ***representative subset*** with an access mechanism that accounts for the ill-defined structure resulting from all the chemical nomenclature, trivial and trade names, and attribute-oriented cross-cutting groupings (e.g., oxidizing substances, water soluble toxics, etc.).

The starting point for any attempts at classification is thus not organic chemistry or environmental toxicology, but a reflection on likely ways to formulate a problem. Entry points for substance identification are therefore ***type of use*** (e.g., agricultural chemical: pesticide) or industrial origin, i.e., production process or type of industry, implying an industrial waste stream (e.g., metal plating, pesticide formulation; a listing of 154 industrial waste streams that contain hazardous components is included in the EPA's WET model approach (ICF, 1984a,b)) rather than chemical taxonomy.

Hybrid Knowledge Representation*): Due to the diverse nature of the information required, we have chosen a hybrid approach to data/knowledge representation, combining traditional data base structure and management concepts (e.g., relational data bases) with knowledge representation paradigms developed in the field of AI. While most of the "hard" and often numerical or at least fixed-format data are organized in the form of relational data bases the knowledge bases again use a hybrid representation approach.

*) This section is based on Fedra (1985).

Hybrid Knowledge Representation implies that within our information system, multiple representation paradigms are integrated. A knowledge base might therefore consist of term definitions represented as frames, object relationships represented in predicate calculus, and decision heuristics represented in production rules.

Predicate Calculus (Barr and Feigenbaum, 1981) is appealing because of its general expressive power and well-defined semantics. Formally, a predicate is a statement about an object:

((property_name) (object) (property_value))

A predicate is applied to a specific number of arguments, and has the value of either TRUE or FALSE when applied to specific objects as arguments. In addition to predicates and arguments, Predicate Calculus supplies ***connectives*** and ***quantifiers***. Examples for connectives are AND, OR, IMPLIES. Quantifiers are FORALL and EXISTS, that add some inferential power to Predicate Calculus. However, for the purpose of building up a representation structure for more complex statements about objects, Predicate Calculus representation becomes very complicated and clumsy, therefore in our system it has been integrated only to represent internal facts used by the inference module.

In ***Object-oriented Representation*** or ***frame-based knowledge representation*** (Minsky, 1975; Roberts and Goldstein, 1977; Bobrow and Winograd, 1977), the representational objects or ***frames*** allow descriptions of some complexity. Objects or classes of objects are represented by ***frames*** which form a hierarchy in which each object is a member of a class and each class is a member of a superclass (except the top-level classes). A frame consists of ***slots*** which contain information about the attributes of the objects or the class of objects it represents, a reference to its superclass and references to its members and/or instantiations, if it is a frame that represents a class. Frames are defined as specializations of more general frames, individual objects are represented by ***instantiations*** of more general frames, and the resulting connections between frames form ***taxonomies***. A class has attributes of its own, as well as attributes of its members. An object ***inherits*** the member attributes of the class of which it is a member. The inheritance of attributes is a powerful tool in the partial description of objects, typical for the ill-defined and data-poor situations the system has to deal with.

A third major paradigm of knowledge representation are ***production rules*** (Davis and King, 1977; Weigkricht and Winkelbauer, 1986): they are related to predicate calculus. They consist of rules, or ***condition-action pairs***: "if this conditions occurs, then do this action". They can easily be understood, but have sufficient expressive power for domain-dependent inference and the description of behavior.

To combine the benefits of an object-oriented approach with those of condition-action pairs, a heterarchical frame structure for the chemical data and knowledge bases is being developed in CommonLisp, i.e., an object can be a member of ***several*** classes and each class can belong to several superclasses, and by adding "rule abilities" to a special slot called ***actions***, i.e., this slot does not store information but performs ***procedural tasks*** (Winograd, 1975) which are defined as condition-action pairs. A detailed description of the heterarchical frame-structure is given below.

Heterarchical Structure for Information Management: Our approach foresees the use of a basic list of about 500 ***substances*** (or individual substances, i.e., entities that do not have any subcategories), constructed as a superset of EC and USEPA lists of hazardous substances. In parallel we have constructed a set of ***substance classes*** which must have at least one element in them. Every ***substance*** has a list of properties or attributes; it also has at least one ***parent substance class*** in which it is a member. Every member of a group inherits all the properties of this group. In a similar structure, all the groups are members of various other parent groups (but only the immediate upper level is specified at each level), where finally all subgroups belong to the top group ***hazardous substances***.

While attributes of individual substances are, by and large, numbers (e.g., a flash point or an LD_{50}), the corresponding attribute at a class level will be a range (flash point: 18-30°C) or a symbolic, linguistic label (e.g., toxicity: very high).

The structure outlined below also takes care of unknowns at various levels within this classification scheme. Whenever a certain property is not known at any level, the value from the immediate parent_class (or the composition of more than one value from more than one immediate parent_class) will be substituted. The structure is also extremely flexible in describing any degree of partial overlap and missing levels in a hierarchical scheme.

Frame Syntax: Each class-frame consists of the following six slots:

- **Explanation:** verbal information about the current frame, concerning the substance class which is represented by the frame, its attributes, the default values and/or indirect references and the position of the frame in the heterarchical structure;
- **Superclasses:** references to the classes to which the current frame belongs;
- **Description:** attributes with values and/or procedural attachments (i.e., procedures which calculate the values or refer to them or both) which describe the substance class represented by the current frame;
- **Subclasses:** references to the classes which belong to the current frame;
- **Instances:** references to the instances (i.e., substances) of the substance class represented by the current frame;
- **Actions:** condition-action pairs, where the actions of an action part are carried out if the frame receives a message which matches the corresponding condition pattern.

The **formal description** of a frame is as follows:

```
(Class    classname
          (Explanation    (<Verbal Information>))
          (Superclasses   (<List of Classnames>))
          (Description    ((<Slotname>-1 <Filler>-1)
                           (<Slotname>-2 <Filler>-2)
                           ...
                           (<Slotname>-n <Filler>-n)))
          (Subclasses     (<List of Classnames>))
          (Instances      (<List of Substances>))
          (Actions        (If <Condition Pattern>
                           Then <Action Part>)))

<Verbal Information> =   a list of words

<List of Classnames> =   classname |
                         classname <List of Classnames>

<List of Substances> =   substance |
                         substance <List of Substances>

<Slotname> =   attribute of the represented class

<Filler> =   (Class classname) |
             (Value value) |
             (default (Lisp s-expression)) |
             ($if-needed (Lisp s-expression)) |
             ($if-added (Lisp s-expression)) |
             ($if-changed (Lisp s-expression)) |
             ($if-deleted (Lisp s-expression))

<Condition Pattern> =   (<Pattern>-1
                         <Pattern>-2
                         ...
                         <Pattern>-m)

<Pattern> =   constant | ?variable | #

<Action Part> =   (Lisp s-expression)
```

As an example, two class-frames from the heterarchical knowledge base structure for phenols are given below:

```
(Class aromatics:
 (Superclasses    (Object))
 (Description     (attribute-1 .....)
                  (attribute-2 .....)
                  ...
                  (attribute-n .....))
 (Actions         (If (List your members)
                   Then (prog (ask self subclasses)
                              (ask self instances))))
 (Subclasses      (aromatic_hydrocarbons
                   aromatic_heterocyclics))
 (Instances       NIL))

(Class
hydrocarbons_substituted_with_two_chlorines:
 (Superclasses
    (aromatic_hydrocarbons_double_substituted
     chlorinated_phenol
     mixed_chlorinated_aromatic_hydrocarbons))
 (Descriptions  (attribute-o+1 .....)
                (attribute-o+2 .....)
                ...
                (attribute-p .....))
 (Subclasses    NIL)
 (Instances     (2,4-Dichlorophenol
                 2,6-Dichlorophenol)))
```

Information Retrieval: To retrieve information the user may directly enter the name of a substance or of a substance class, or he may specify value ranges (numerical and/or symbolic) for one or more substance (class) attributes. The information system transforms this specifications into messages for the top-level classes (also called viewpoints). On receiving these messages the frames which represent the viewpoints are activated and check if they are selected by the user's specification. If they are, they proceed to create messages for their subframes which again perform their matching operations and create messages, and so on. This recursive procedure does not need to search through the whole structure because it is ***directed*** by the procedural knowledge supplied in each frame (rules in the ***Actions*** slot and references in the ***$if***-slot) and supported by information-inheritance. This procedure results in a substructure of valid substance classes which represents the systems view of the user's level of expertise, guiding the user to the more detailed information, if he agrees to proceed with the interaction. An example of how this information is displayed is given in Figure 2.

Updating the Knowledge Base: Updating is quite similar to the retrieval of information. First, the substructure which will be affected is localized by an interframe message sending/receiving sequence. Then the updates of the attribute values are entered and checked if they are consistent within the selected substructure by using the ***$if-added*** and the ***$if-needed*** slot fillers together with the rules of the ***Actions*** slot which deals with consistency tests. The same procedure is used for the next higher aggregation levels until the whole structure has been proved to be consistent with the new and/or changed attribute values.

This information system is a central module of the framework system and is one example of AI technology embedded in the system.

Data Bases, Simulation, and Optimization

The integrated software system as described above can be used in a variety of ways. The simplest and most straightforward use of the system is as an ***interactive information system***. Here the user "browses" through the data and knowledge bases or asks very specific questions. As an example, consider the substances data base (Fedra et al., 1986) which can also be used from any of the impact models; the necessary substance-specific parameters are automatically retrieved and made available to the calling model.

The second mode of use is termed ***scenario analysis***. Here the user defines a special situation or scenario (e.g., the release of a certain substance from a facility), and then traces the consequences of this situation through modeling. The system will assist the user in the formulation of these "What if..." questions, largely by offering menus of options, and ensuring a complete and consistent specification.

The scenario analysis mode can use any or all models in isolation or linked together; the selection and coupling of models is automatic. The evaluation and comparison of alternatives is always performed in terms of a subset or all of a list of criteria, including monetary as well as symbolic, qualitative descriptors (Fedra, 1984). The use of certain models is implied by the selection of indicators and criteria that are chosen to describe a scenario's outcome.

Two time domains for scenario analysis with different problems addressed are supported: the models can either be used to simulate medium- to long-term phenomena, with a characteristic time scale of years, or short-term events, i.e., accidents, with a characteristic time scale of days. Switching from one mode to the other, with the necessary aggregation or disaggregation is possible. Similar to this switching in the time domain, a change in the space domain must also be supported. There is of course a close linkage between time and space scales, in that most short-term phenomena like spills or accidents are relevant on a local to regional scale, whereas long-term phenomena like continuous routine release of hazardous substances will usually be considered on a regional to national scale.

Scenario analysis may be either straightforward simulation, or a combination of simulation and optimization techniques (Figures 3, 4). In the latter case, the user does not have to specify concrete values for all control variables defining a scenario, but rather specifies allowable ranges as well as a goal structure. In the optimization mode, our system becomes a decision support system proper.

Using techniques such as reference points in multi-objective problems (Wierzbicki, 1983), an appropriate framework allows one to modify expectations interactively. The user can redefine objectives and constraints in response to first results. Alternatively, discrete optimization can be used as a post-processor for the results of simulation. The human evaluator is therefore directly incorporated in the optimization process.

All these refinements of the basic information and simulation system however must not complicate the user's interactions with the system. Ease of use, and the possibility of obtaining immediate, albeit crude and tentative, answers to problems the machine helps to formulate in a directly understandable, attractive and pictorial format are seen as the most important features of the system.

References

Barr, A. and Feigenbaum, E.A. [eds.] (1981) The Handbook of Artificial Intelligence. Vol. 1. Los Altos, CA. William Kaufmann, Inc. pp. 163-171.

Bobrow, D.G. and Winograd, T. (1977) An overview of KRL, a knowledge representation language. Cognitive Science 1:3-46.

Davis, R. and King, J.J. (1977) An overview of production systems. In E. Elcock and D. Mitchie [eds.], Machine Intelligence 8. Chichester, England. Ellis Horwood. pp.300-332.

Fedra, K. (1984) Interactive Water Quality Simulation in a Regional Framework: A Management Oriented Approach to Lake and Watershed Modeling. Ecological Modelling, 21 (1983/1984) pp.209-232.

Fedra, K. (1985) Advanced Decision-oriented Software for the Management of Hazardous Substances. Part I: Structure and Design. CP-85-18. International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria.

Fedra, K. (1986) Advanced Decision-oriented Software for the Management of Hazardous Substances. Part II. A Prototype Demonstration System. CP-86-10, International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. 98p.

Fedra, K. and Otway, H. (1986) Advanced Decision-oriented Software for the Management of Hazardous Substances. Part III. Decision Support and Expert Systems: Uses and Users. CP-86-14, International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. 44p.

Fedra, K., Mutzbauer, G., Weigkricht, E. and Winkelbauer, L. (1986) Advanced Decision-oriented Software for the Management of Hazardous Substances. Part IV: Substances Data and Knowledge Base. CP-86-xx. International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. Forthcoming.

Fick, G. and Sprague, R.H.jun.[eds.] (1980) Decision Support Systems: Issues and Challenges. Proceedings of an International Task Force Meeting, June 23-25, 1980. IIASA Proceedings Series, Pergamon Press, Oxford. 189p.

ICF (1984a) The RCRA Risk-Cost Analysis model. Phase III Report. Submitted to the Office of Solid Waste, Economic Analysis Branch, U.S. Environmental Protection Agency. March 1, 1984. ICF Incorporated, International Square, 1850 K Street, Northwest, Washington, D.C.

ICF (1984b) The RCRA Risk-cost Analysis model. Phase III Report. Appendices. Submitted to the Office of Solid Waste, Economic Analysis Branch, U.S. Environmental Protection Agency. March 1, 1984. ICF Incorporated, International Square, 1850 K Street, Northwest, Washington, D.C.

Minsky, M. (1975) A framework for representing knowledge. In P. Winston [ed.], The psychology of computer vision. New York. McGraw-Hill. pp.211-277.

Phillips, L. (1984) Decision support for managers. In: H. Otway and M. Peltu [eds.], The Managerial Challenge of New Office Technology, Butterworths, London, 246p.

Roberts, R.B. and Goldstein, I.P. (1977) The FRL Primer. MIT Artificial Intelligence Laboratory. Memo 408, July 1977. Cambridge, MA.

Weigkricht, E. and Winkelbauer, L. (1986) Knowledge-based Systems: Overview and Selected Examples. WP-86-xx. International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. Forthcoming.

Wierzbicki, A. (1983) A Mathematical Basis for Satisficing Decision Making. Mathematical Modeling USA 3:391-405.

Winograd, T. (1975) Frame representations and the declarative/procedural controversy. In Bobrow, D.G. and Collins, A. [eds.], Representation and understanding: Studies in cognitive science. New York. Academic Press. pp. 185-210.

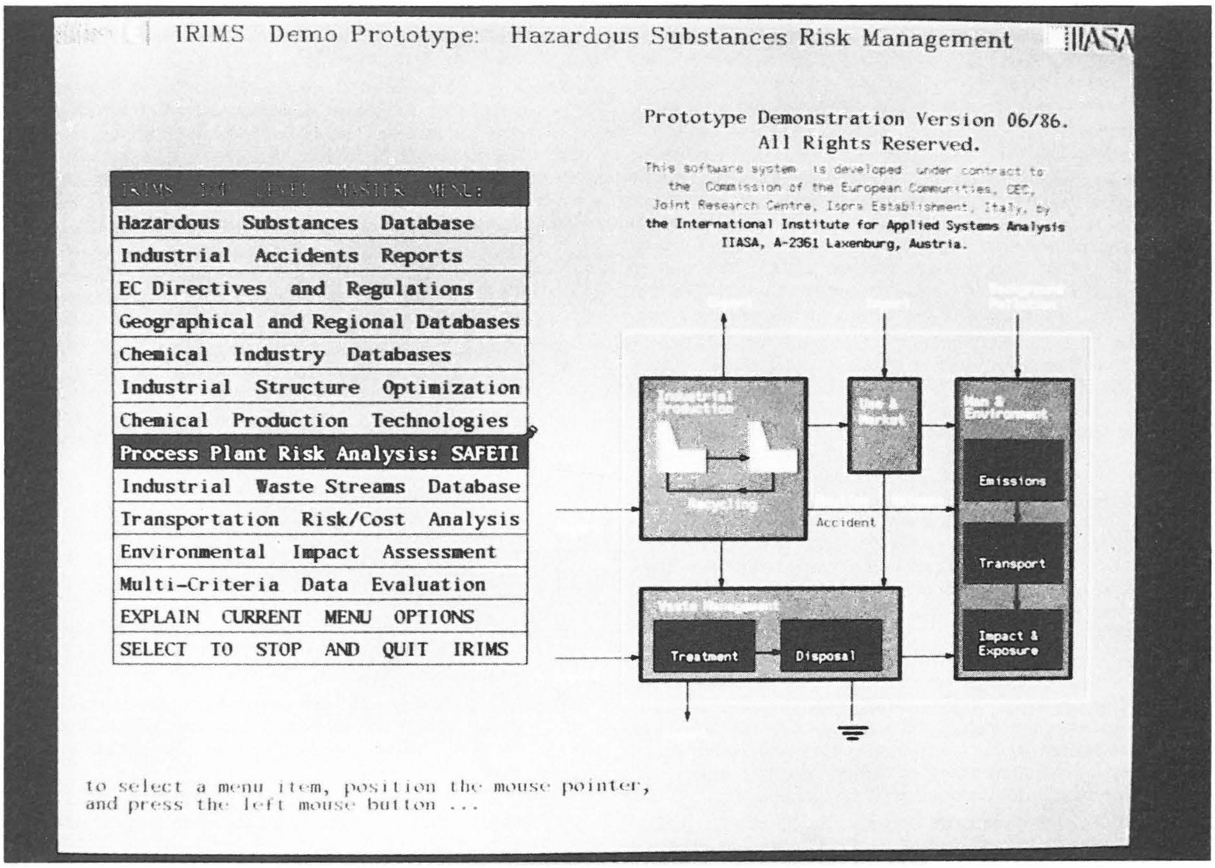

Figure 1: Top level system master menu

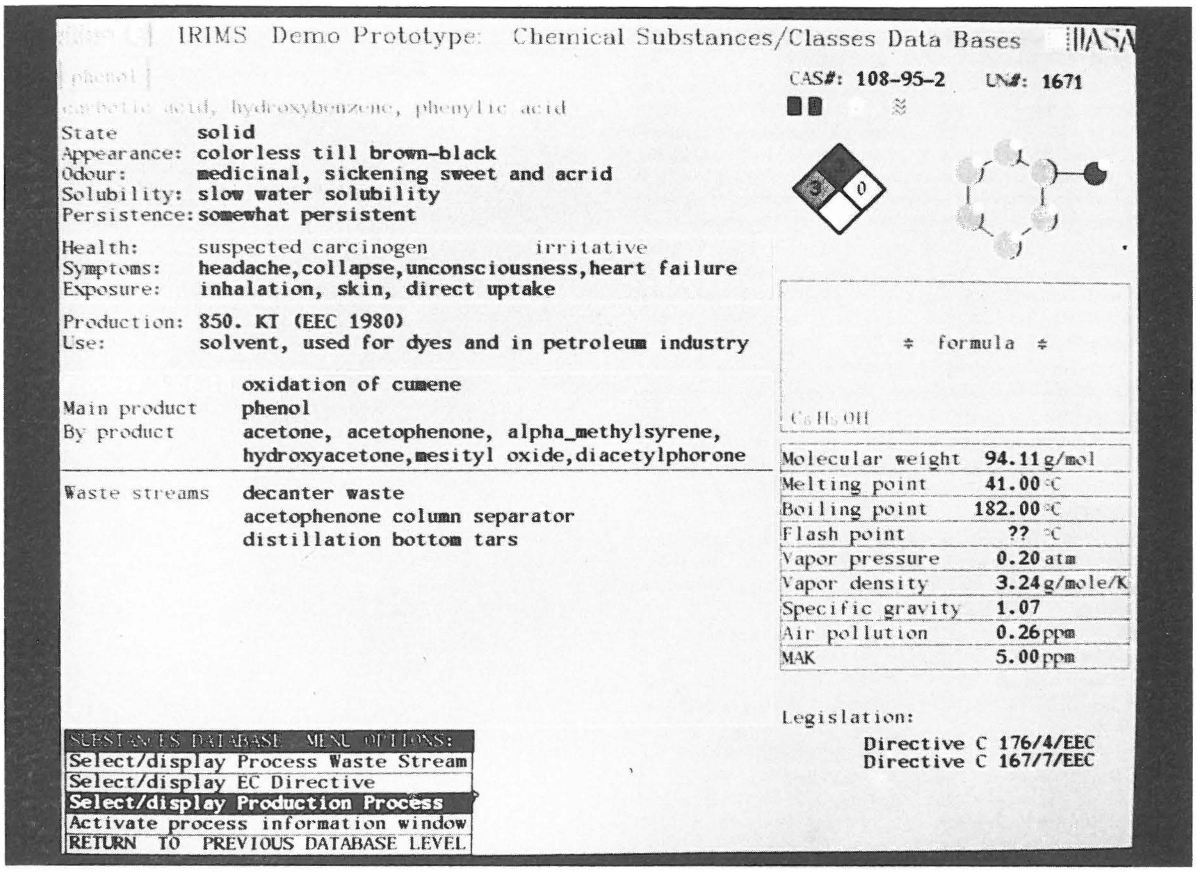

Figure 2: Summary page description for a basic substance

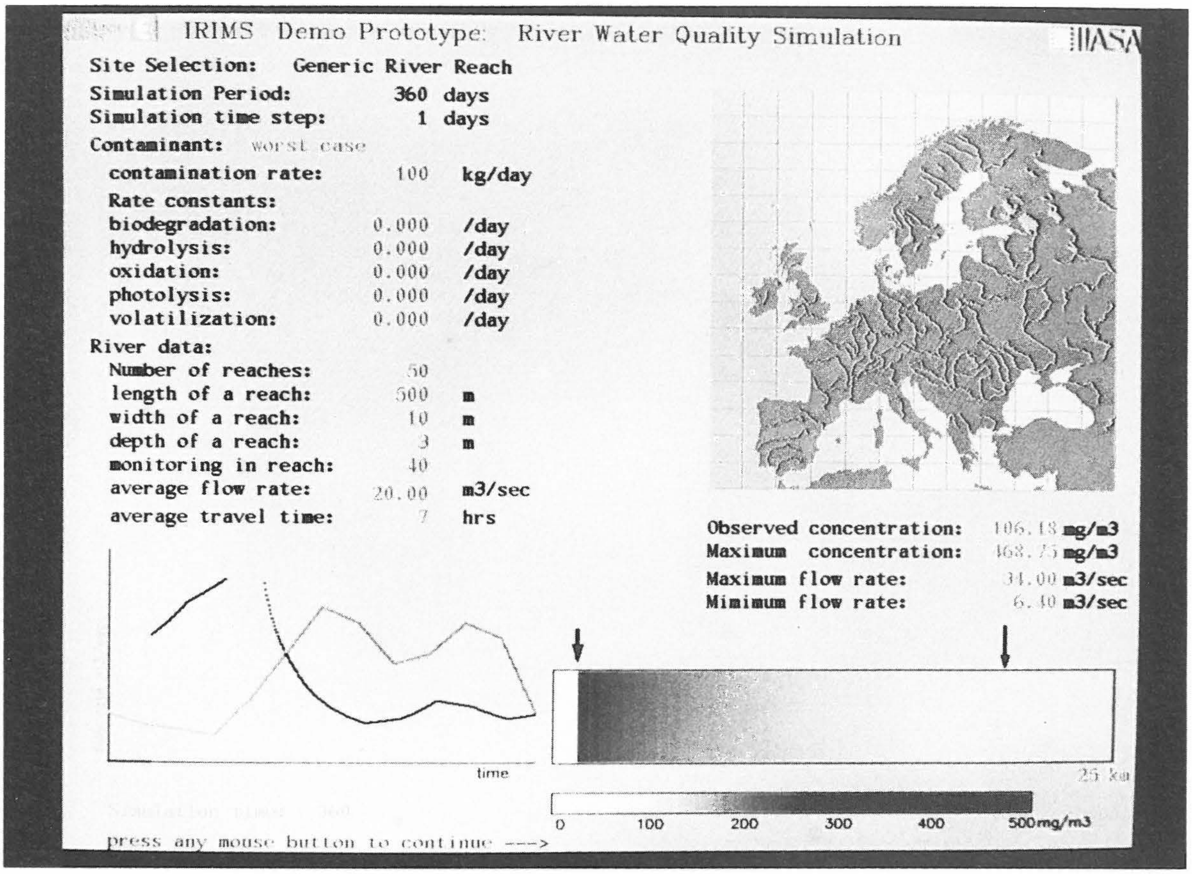

Figure 3: River water quality simulation model

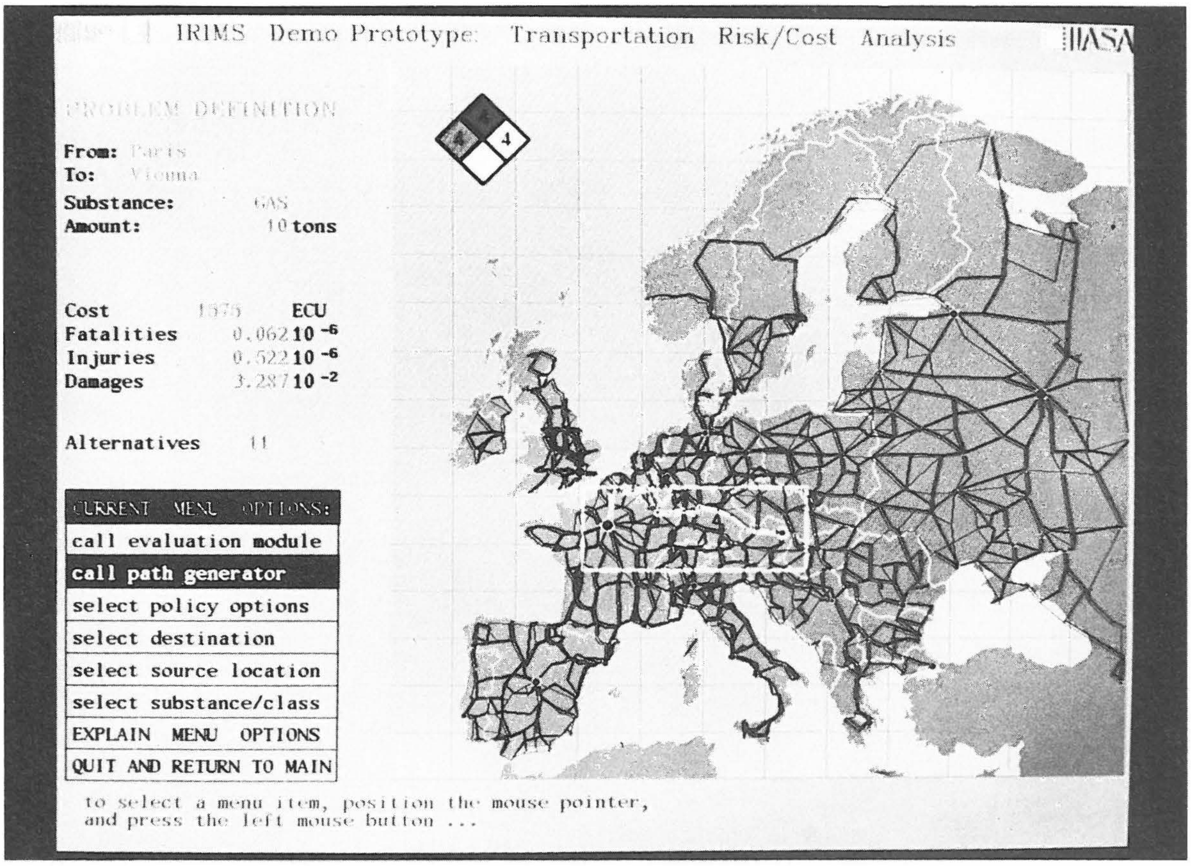

Figure 4: Transportation risk-cost analysis simulation model