
ALBERT A. ANGEHRN
INSEAD, European Institute of Business Administration
Boulevard de Constance
Fontainebleau Cedex, F-77305, France

HANS-JAKOB LÜTHI
ETH, Swiss Federal Institute of Technology
IFOR, ETH-Zentrum
Zürich, CH-8092, Switzerland

Visual interactive modeling principles and modeling by example offer concrete guidelines for developing user-centered DSS generators that exploit the advantages of visual modeling while supplying active decision support through proper integration of AI techniques. Human-computer interaction remains a central issue, suggesting further research to realize a high level of human-machine cooperation in problem solving and decision making.

Ideally DSSs should behave like human consultants supporting decision makers in expressing, structuring, and better understanding their problems. The development of such convivial, cooperative systems is a key issue of decision support. Our contribution to this research area is illustrated by a new intelligent DSS for decision making in a geographical context. The design principles underlying this system emphasize system usability and suggest a new way for enhancing human-computer cooperation in DSSs. A highly visual, object-oriented modeling environment allows end-users to incrementally describe their problems, and a knowledge-based component interacts with the users by exchanging "examples" and stimulates them in generating and exploring alternatives during the decision-making process.

DSS Goals
The decision support systems field has been appropriately defined by Sol [1983] as the development of approaches for applying information systems technology to increase the effectiveness of decision makers in situations where the computer can support and enhance human judgment in the performance of tasks.
that have elements which cannot be specified in advance (p. 1).

Accordingly, in the DSS field we aim at changing the paradigm for providing computer support for managerial tasks, suggesting new ways for matching the technical world of information processing (technological perspective) with individual approaches to problem solving (human perspective) [Angehrn et al. 1990]. Aiming at a revaluation of the human perspective, computer support for managerial decisions needs to evolve

— From traditional data processing to visual information processing,
— From formal analysis to interactive modeling,
— From algorithmic manipulation to direct manipulation,
— From problem solvers to context-sensitive modeling environments, and
— From passive support systems to intelligent cooperative systems.

In our view, the main goal of a DSS is not to deliver information system technology or specific problem-solving techniques but to provide decision makers with tools for interactively exploring, designing, and analyzing decision situations in a manner compatible with their mental representations. Some basic guidelines for designing DSSs with the above-mentioned characteristics have proved applicable and appropriate in practice. In particular, two basic design principles have resulted in the development of the Tolomeo system, a visual interactive, intelligent DSS supporting a wide range of decision-making processes in the geographical context.

**Modeling Environments and Symbiotic Systems**

Functionally, a DSS is made up of three components: a language subsystem, a knowledge subsystem, and a problem-processing subsystem. The user communicates problems to the DSS using the expressive power (semantics and syntax) of the language component. Conforming to the user's needs, the problem-processing component is then responsible for activating the available functions, and generating and conveying the appropriate information to the decision maker. We took into account this previously formulated conceptual model [Holsapple and Whinston 1987; Arnoldi 1989] and the goals listed above in establishing the functional and technical requirements for each component of Tolomeo (Figure 1).

For the DSS user, the interface is the system. The language component is the most crucial factor in DSS design because it determines whether the functions of a potentially rich system are transparent to the user [Winograd and Flores 1986]. If they are not, he or she will reject the system or use it incorrectly.

This fact is often ignored or underestimated in traditional system development, where identifying and defining the problem and developing algorithms play the major roles. In Figure 2, we compare the classical method and data-centered approach—whose focus is on delivering

---

**Ideally DSSs should behave like human consultants supporting decision makers in understanding, expressing, and structuring their problems.**
functionality—with a user-centered one [Angehrn 1989b]. This user-centered approach aims at warranting system usability by first designing an environment the user can communicate and work effectively. Suitable information processing techniques (functionality) are integrated into the system only at the end of the development process. Supporting decision makers with elaborated models and techniques has value, but only usable methods are really effective! Our first design principle is: Usability prior to functionality.

The method- and data-centered approach (Figure 2, left) is still recommended for automatic planning, allowing tasks to be specified (and delegated!) precisely and facilitating the use of suitable and reliable solving techniques. When the user has to design a large part of the application, the emphasis in developing the system then shifts to analyzing the decision context, determining an appropriate language and form of communication, identifying notions, concepts, and operations that are familiar to the user and that correspond to his or her knowledge and experience [Angehrn et al. 1990].
To perform the next step of implementing an interactive modeling environment, the system designer should study two technical elements: the well-established rules for designing visual languages [Chang 1987] and object-oriented principles for supporting the modeling process [Brodie, Mylopoulos, and Schmidt 1984]. This will lead the system designer to identify the fundamental constructs that support visual interactive modeling (VIM).

We based the design of Tolomeo on three such basic constructs called modeling primitives:

— Objects, that is, concrete or abstract entities (cities, ambulances, cars, projects . . .) with a set of specific attributes (population, accidents, speed, duration, . . .),

— Relations, that is, objects describing structural or functional dependencies between other objects (connections, streets, shipments, regions, . . .), and

— Constraints on attribute values (for example, car speed less than 100 miles per hour).

These generic modeling primitives are made available to the user in a visual interactive environment. The user can incrementally describe different decision situations and design solution alternatives by introducing and manipulating on the screen the concrete objects he or she associates with different problem entities.

Furthermore, cooperative teamwork between the user and the computer-based system is possible only when both parties are active in the problem-solving process. The system itself must therefore function in response to prevailing conditions and smoothly interlace the methods it has available in the modeling process. Our second design principle is the principle of active cooperation: While the human being divides the complex planning job into tasks and assigns some problems to the machine to resolve, the system takes on the role of advisor and facilitator [Angehrn 1989a]. During the decision-making process, it stimulates, supports, and suggests the application of analytical methods and other techniques for deducing relevant information.

By following these two basic design principles—usability prior to functionality and active cooperation—we can create so-called symbiotic systems, which complement human skills through computer technique. This kind of system allows tasks to be carried out which neither man nor computer could or would want to perform alone, as described by Fischer and Gunzenhäuser [1986].

Essentially, users can now perform the following functions:

— They can analyze decision situations according to their personal styles and knowledge.

— They can build and compare various quantitative models.

— They can adapt these models to changing conditions.

— They can evaluate different aspects of their activities using a variety of different means.

**VIM and Intelligent DSS: Putting Theory into Practice**

Tolomeo is the concrete example of a system that embodies a user-centered approach. It supports visual interactive modeling, and it is based on the idea of modeling by example (MbE) as a means of en-
INTELLIGENT DSSs

hancing cooperation between the user and the system. Tolomeo is a DSS generator, that is, an open system (like a spreadsheet) that can be applied to a broad range of decision situations that can be modeled in a geographical context, that is, described using a geographical map as the main communication medium. Typical examples of such decision problems are the analysis of geo/demographic data in marketing or political decisions (Tolomeo can be employed as a GIS, Geographic Information System) and more complex tasks that involve the use of sophisticated modeling and solving techniques, such as routing vehicle fleets, allocating resources (ambulances, power plants, or production sites), and analyzing the consequences of changing the location of a firm (a study we have conducted for the marketing department of a large Swiss newspaper).

An Example: Modeling Communication Networks

To illustrate the type of support delivered by Tolomeo during a decision process, we describe its use in planning and analyzing a communication network. This is only an example; Tolomeo can be applied even in less structured decision situations that involve the evaluation of a variety of criteria (economical, political, environmental, and other factors) that have nothing in common with this example. Furthermore, even in this small example we do not start from a well-defined problem specification. This aspect underlines the often neglected reality that well-structured problems usually appear only as partial components in a decision process and are seldom available a priori in an objective form. They only come to light in the flow diagram process stemming from a decision maker's subjective feeling.

To design and analyze a communication network, the planner can describe the task within a short time using the Tolomeo system. This description or model of the situation reflects the planner's view of the problem at a specific phase of the decision process (Figure 3). Apart from the background (a map of the planning area), the model consists basically of two main components: nodes representing existing sending and receiving stations that are connected by transmission channels (the elements of the network through which messages can be transmitted from one node to another); and switching centers, which the planner is interested in introducing to serve as collecting points for message distribution. In the planner's model, the switching centers can also be interconnected by a second network.

Starting from such a small model, the planner can interactively explore different alternatives for the number of switching centers and their locations and for the design of a network connecting them. The process of generating and evaluating different plans is supported by Tolomeo visually as well as quantitatively: each element of a model can carry some dynamic information in the form of user-defined properties whose values can be typed in, extracted from a data base, or calculated by formulas as in a spreadsheet (Figure 3).

Decision Support through Visual Interactive Modeling

A system like Tolomeo can be described as having four important dimensions:
(1) End-user (decision-maker) modeling;
(2) Modeling as a concrete, visual, and
Figure 3: This model represents the planner's view of the communication network. The different model components (nodes, switching centers, networks) can be defined and manipulated graphically. Quantitative information (user-defined attributes) attached to the single object components is displayed in tabular form. For example, the table on the left hand side shows the properties of the node Basel (24 messages sent with an average transmission time of 5.2 seconds). In a second table (right), the planner can visualize the data characterizing the connection between the two nodes St. Gallen and Chur.

incremental process;
(3) Reactive system behavior; and
(4) Smooth integration of supporting functions (especially analytical methods) during the decision process.

End-User Modeling

One of the most important objectives of DSS development is the design of flexible tools that users can employ individually during their decision processes. Modeling (the incremental process of converting mental concepts into explicit and hence analyzable representations) is crucial in every formal decision process. One of the most interesting issues in DSS research is the development of concepts and tools to support decision makers in performing the modeling process by themselves.

As many of today's DSSs focus on problem solving rather than on supporting the modeling process, users are often limited by the scope of action predefined by the system designer. Systems like Tolomeo, on the other hand, supply flexible modeling environments that enable users to develop their own models interactively, starting from their individual views of the decision situations they face.

This flexibility is based on a clearly arranged set of primitives, basic modeling tools that fulfill two requirements:
— They are general enough and allow the description of a variety of real-world decision situations (expressive power).
INTELLIGENT DSSs

— They permit end-users to master them without great effort, allowing decision makers to concentrate on the task, not on the tools (usability).

In Tolomeo's modeling environment, users can perform every operation (definition, modification, representation, and organization of model components) interactively by manipulating visible and tangible objects. Using this design principle in developing man-machine interfaces permits the designer to avoid abstract structures (like command languages) and to realize a concrete modeling style based on context-specific elements and representations. As discussed by Turban and Carlson [1989] in their paper on visual decision making, this kind of environment can support users in interactively creating or modifying a model of complex decision situations, allowing them to experiment with various decision alternatives and see their impact in a graphical, even dynamic form.

Modeling as a Concrete, Visual, and Incremental Process

The modeling primitives in Tolomeo are mainly based on object-oriented concepts [Stefik and Bobrow 1986]. Hence every model can be formally represented as a collection of object-classes characterized by the entities, the properties, and the relationships (objects, formulas, and hierarchical structures) that the user describes interactively during the decision process. Nevertheless the user is not obliged to deal explicitly with an abstract model represen-
Figure 4: In analyzing a problem, users can visualize only the model components they are really interested in. For instance, the communication network can be displayed showing only the nodes and the basic network (left) or omitting the network but visualizing the switching centers as circles, and the connections between them (right).

achieved in Tolomeo supports modeling as an incremental process. The user can incrementally build, complete, and test models. This makes the human-machine interaction easy and concrete, and the process of modeling stimulates the user’s learning about the unstructured situation.

Reactive System Behavior

The effective cooperation between a user and a DSS relies on the system’s capability to complement the user’s activity with suitable feedback. In a modeling environment like that described, the reactive behavior is strongly connected to the modeling primitives the user employs in different situations. In Tolomeo, for instance, three conceptually different feedback types can be distinguished:

1. A recalculation mechanism operates on the functional relationships (formulas) the user introduces dynamically between the components of a model. This kind of feedback has three major advantages: it guarantees model consistency at every stage; it supports incremental model development (the user can always see and work with an updated model); and it allows the user to perform what-if analysis and compare alternatives interactively.

2. The visual modeling environment of Tolomeo also makes available modeling primitives called constraints. They support the introduction of semantic information into a model: using constraints, the user can explicitly and dynamically express preferences (goals) by attaching specific requirements to any model component. For example, the user could attach a constraint to the values of the nodes’ property “average transmission time” and express in this way that these values should not exceed 2.5 seconds for each node of the network.

Thus, Tolomeo’s second type of reactive behavior shows the importance of control mechanisms processing different types of user-defined constraints. Such mechanisms dynamically check the existence of constraints, test if the conditions are fulfilled, and advise the user through visual feedback. The system indicates what model components do not achieve user-defined targets and supports in this way a kind of
INTELLIGENT DSSs

visual optimization. The user can immedi-
ately see the quality of the decision alter-
atives he or she has designed and obtain
useful indications for improvements.

(3) A third important type of feedback
concerns the control of structural consis-
tency. In building models of decision situa-
tions, the user will always employ the
available modeling primitives for defining
structural relationships among different
model components. For example, in our
network model (Figure 3) the switching
centers’ locations can be structurally con-
fined to places where nodes already exist.
It is again up to the system to guarantee
consistency and to react in case one of
these implicit rules is broken by the user
during the modeling process.

Smooth Integration of Supporting
Functions

The concepts described so far, as well as
their realization in the decision support
system Tolomeo, are mainly oriented to
supporting the user’s active involvement in
the process of designing and analyzing de-
cision alternatives. But what about the sys-
tem’s activity? Apart from the reactive be-
havior described, the system remains quite
passive (reaction is not equal to action)
(Keen [1986] and Jelassi, Williams, and
Fidler [1987] discuss the emerging active
role of DSSs).

To supply active support, we extended
the system to act as a consultant able to
(1) Understand and interpret the decision
situations described by the user,
(2) Dynamically formulate sugges-
tions, and
(3) Support the ad hoc application of dif-
ferent techniques.

The main idea underlying this interac-
tive consulting activity consists in letting
the system communicate its knowledge to
the user by exchanging concrete examples
(as human experts usually do).

Note that artificial intelligence (AI) tech-
niques are used in this framework in a dif-
f erent way than in expert systems, where
the main goal is to enhance the autono-
 mous problem solving capabilities of the
system. The modeling by example (MbE)
approach [Angehrn 1989a, 1989b] exploits
AI techniques to improve the system’s ca-
pabilities in communicating with the user
at a high level. The aim of MbE is to sup-
port the decision makers’ creativity and in-
tuition without attempting to substitute for
their judgment.

The modeling by example scheme (Fig-
ure 5) has been applied in Tolomeo to sup-
port users in employing complex methods
(mathematical tools and other information
processing techniques) flexibly within vi-
sual interactive modeling environments.
One of the main benefits of this unconven-
tional, direct way of delivering support (by
example) comes from supplying the system
functionality (model- and knowledge-base)
smoothly: users do not need to interrupt
their decision processes or to reformulate
their models by switching from a descrip-
tive to a normative mode. The decision al-
ternatives the user designs and analyzes
manually in a what-if mode serve directly
as examples for the knowledge-based com-
ponent of the system (which has been im-
plemented on a Macintosh II using the
procedural language Modula2 and the de-
clarative language Prolog). The user can
simply hand over his or her alternatives to
the system and in this way activate an in-
fERENCE mechanism, which, after a dialog,
The user designs decision alternatives

Full integration into the user's model

New decision alternatives generated by the system

The user's examples start the interaction cycle

The examples activate the knowledge-based component of the system

An inference mechanism analyzes the examples and the model and generates facts (formal problem representation)

The system formulates hypotheses (logical inference process)

Dynamically, it generates suggestions (pattern-matching process)

In a dialog, the system presents suggestions and inquires about specific user preferences

The system activates the techniques stored in the methods-base managing all the technical input/output details

It generates solutions and communicates the results as new examples

Figure 5: Within Tolomeo, the active system support takes place by example. The system analyzes the model and the solution alternatives designed interactively by the user and suggests new solution alternatives after selecting and executing complex information processing techniques such as optimization algorithms and heuristics applicable to routing, allocation, and clustering problems.

usually concludes with some suggestions for generating new and "better" alternatives.

For example, the planner may want to enhance the model of the communication network (Figure 3) by introducing some sectors partitioning the planning area. Using the visual- and graphic-based modeling environment of Tolomeo, the user can introduce and specify such new model components by drawing the desired objects (in this case: four polygons) on the map displayed on the screen. The planner can then hand over these self-designed sectors as examples of what to look for and activate in this way the consulting component of Tolomeo. After analyzing the structural characteristics of the user's example, the system will suggest the use of a suitable method (for instance, a clustering technique), extract the needed input parameters from the user's model, and finally generate a new example. Such a system-generated example may, for instance, correspond to a new set of sectors clustering the planning region and showing some op-
INTELLIGENT DSSs

timality characteristics, such as a balanced distribution of the message quantities or flows in the different sectors.

As the alternatives the system generates are integrated into the user's model, they are immediately available for further explorations (for example through what-if analysis). For instance, the planner can use the new sectors in designing new locations for switching centers. Again, the user can exploit the MbE approach implemented in Tolomeo, handing over the self-designed switching centers as examples of what to look for. The knowledge-based component of Tolomeo will then select and execute an appropriate method (for instance, a center or median algorithm) for generating and suggesting new locations for the switching centers (Figure 6).

Because communication takes place through concrete examples, the decision maker does not have to battle with preparing input or representing output. The system takes over these technical tasks and allows the user to operate at a higher level. Different algorithms and solving techniques can be easily linked together, allowing an effective combination of human skills and computing power.

Conclusions

The VIM principles and the modeling-by-example approach offer concrete guidelines for developing user-centered DSSs like Tolomeo that exploit the advantages of visual modeling and supply active decision support by integrating AI techniques. In addition to Tolomeo, which we developed to support decision making in a geo/demographical context, we are developing other tools based on the same design principles but acting in different decision contexts. They will provide a broader basis for evaluating and empirically testing the principles described in this paper and for guiding the development of more convivial DSSs for specific real-world situations.

Human-computer interaction remains a central issue in the DSS domain, and further research is needed to realize a high level of human-machine cooperation in problem solving and decision making.

Figure 6: The switching centers designed by the planner (left) serve as examples for activating the consulting component of Tolomeo. The system then generates a new solution alternative—for instance the cost-optimal one—and integrates the new locations (right) graphically into the user's model.
References


