

Artificial Intelligence and Education

Proceedings of the 4th
International Conference on
AI and Education,
24-26 May 1989,
Amsterdam, Netherlands

D. Bierman, J. Breuker and J. Sandberg, Editors

IOS

1989

Amsterdam • Springfield, VA • Tokyo

ELAB - An Electronics Laboratory

Heinz-Dieter Böcker, Jürgen Herczeg

*Inst. f. Informatik,
Univ. Stgt., Herdweg 51,
D-7000 Stuttgart 1, F.R.G.*

Michael Herczeg

*ANT Nachrichtentechnik GmbH,
Gerberstr. 33,
D-7150 Backnang, F.R.G.*

Abstract

This paper describes ELAB, a knowledge-based, exploratory design and learning environment for four-poles, electronic circuits, consisting of resistors, capacitors and coils. Various kinds of four-poles may be constructed by the user and experimented with. The system contains a simulator that is augmented by a component containing knowledge about the simulation process and the application domain. The system supports quantitative analysis as well as qualitative investigation of the behavioral properties of user designed four-poles by providing adequate simulation parameters, answering questions about the application domain, and giving examples and partial solutions for problems posed by the user. The system makes heavy use of a direct manipulation interface and employs several types of visualization techniques.

Human - Problem-Domain - Communication

The *desktop metaphor* [14] is just a specific instance of the more general class of *human - problem-domain - communication (HPC)* situations as defined by [9]. It summarizes some surface characteristics of typical office workers' problem-domains and is among the best understood HPCs. We believe the term HPC in many situations to be more appropriate as a fundamental research paradigm than *human-computer-communication (HCC)*. A user interacting with a computer is primarily not interested in the interaction process. The interaction serves purposes and goals. It is a goal-oriented problem solving activity addressing application dependent problems of specific problem domains. The ultimate goal of users is to inspect and manipulate the objects of the application they are interested in. The HPC paradigm has many similarities with the idea of 3-D *virtual worlds* discussed by Brooks [3]. Both ideas treat the computer as a medium that is virtually transparent. With respect to the communication process. Both paradigms entail the idea to replace system-oriented descriptions by problem oriented descriptions [13,16] that are meaningful to the user.

Working environments with characteristics slightly different from the office situation are laboratories of any kind (e.g., electronics, physics or music laboratories). Laboratories provide exploratory environments within which one can study problems and pursue goals of learning and research. Laboratory environments simulated on a computer screen and software-based construction kits share a lot of properties, especially when used for educational and tutorial purposes: standardized parts and modules, defaults, some deliberately introduced inefficiency, protective shields to prevent disaster, etc.

Recently, the paradigm of construction kits has become popular as a conceptual base for building human-computer interfaces. Bill Budge's *pinball construction set* [5] has

triggered a series of systems and design environments (e.g., [8,10,11]) that put the user into the active position of the designer within some problem domain. D. Norman points out one of the great advantages of kits like the pinball construction set [15]: It is a world in which illegal operations are basically impossible. An operation may not achieve a desired result but there is no such thing as an illegal operation. Whether an action achieves the desired result is simply controlled by visual inspection.

The ELAB system combines ideas from the construction kit as well as the HPC paradigm. It is designed to serve as a learning environment within which a student can explore a problem domain, the world of electronic circuits. The system makes heavy use of visualization techniques to mimic a real laboratory; however, it also employs deep and detailed knowledge representations to decide when to use which techniques.

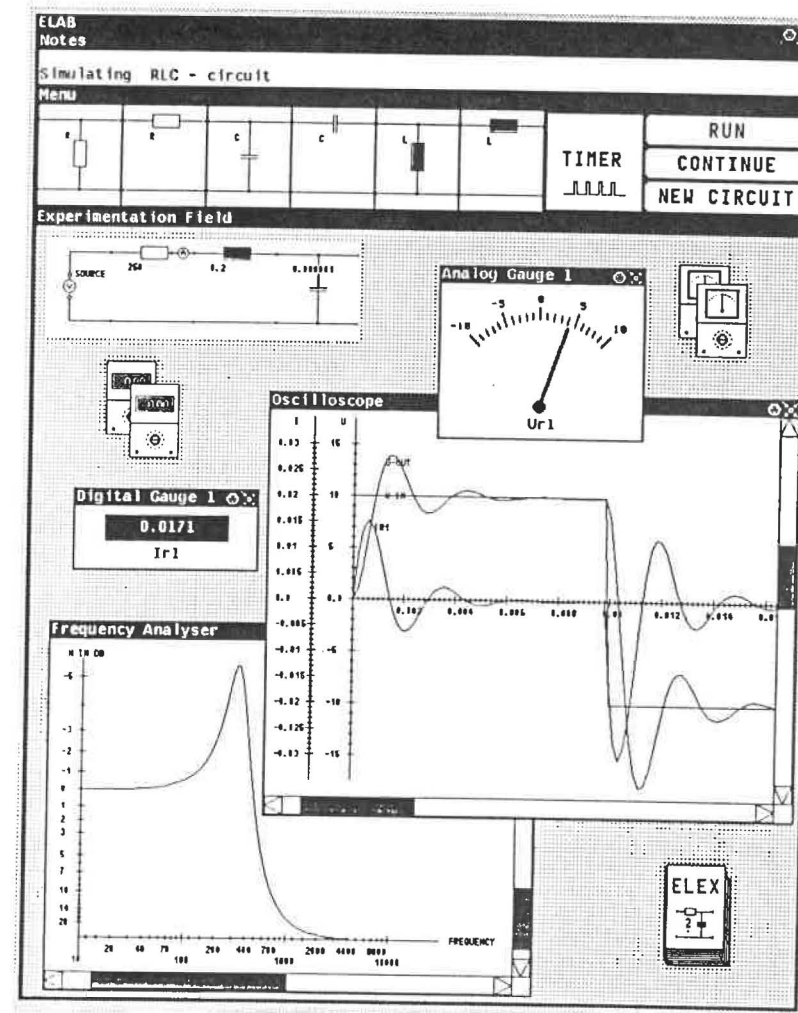
Direct Manipulation Electronics Simulation

The field of electronics has been the focus of intelligent computer-assisted instruction systems from the very beginning. The SOPHIE system [4] is among the most influential systems at all. It watches the user trouble-shooting a specific complex electronic circuit, applying various knowledge representation and inference techniques. SOPHIE communicates with the user solely by means of natural language. This kind of interaction is quite appropriate for providing explanations, but rather awkward for tasks like changing parameters of circuit components or even performing measurements. Another influential system, THINGLAB [2], applies constraint propagation and relaxation techniques to the simulation of electronic circuits, explicitly representing, however, only rather low level knowledge of the application domain. With ELAB, the construction and analysis of circuits may be performed on a computer in basically the same way as in a real laboratory. The user directly manipulates an experimentation field with objects to be found in electronics laboratories (see Figure 1)

- electrical devices (resistors, capacitors, coils),
- voltage source (function generator),
- measuring instruments (oscilloscope, frequency analyzer, analog and digital gauges).

The user may interactively create an arbitrary complex circuit out of the basic building blocks, run simulations of it, and have the system display the results with the help of various instruments: digital and analog gauges, oscilloscopes, and frequency analyzers. In order to create a circuit and run a simulation the user selects, moves, connects, and modifies graphical objects on the screen. The desktop metaphor has been transformed into a *lab metaphor*: the visualization and direct manipulation of an electronics laboratory on a computer screen.

ELAB handles the application domain of electronic circuits built from elementary four-poles consisting of resistors, capacitors or coils. These circuits, though containing only rather simple components (ELAB does not provide active elements like transistors to experiment with), quickly become complex enough that even a human electronics expert is unable to predict their behavior qualitatively, let alone quantitatively. Some of the most interesting circuits to be built with ELAB are voltage dividers, filters (e.g., high-pass, low-pass, bandpass), and resonant circuits. The type of the input signal (e.g., sine- or rectangle-shaped oscillations or pulses) and initial states (e.g., a charged capacitor) of the devices may further increase the complexity of a circuit's behavior. The simulation component provides *transient* and *frequency domain analysis* of circuits represented in a knowledge base.



In the upper part there is a window for textual notes and instructions for the user. Below we have a graphical menu for the elementary four-poles a circuit may be built of and some buttons to control the simulation process. The main part holds the experimentation field containing the circuit connected to the voltage source and various instruments to analyze the circuit's behavior.

Figure 1: ELAB — A Simulated, Direct Manipulation Electronics Laboratory

The various instruments used to analyze the behavior of the circuit are modeled after real hardware instruments with increased functionality. The oscilloscope, for example, may simultaneously protocol as many different signals as the user wishes to see. Its coordinate system is variable, and it displays not only periodical signals like most hardware oscilloscopes do, but also arbitrarily short pulses and varying oscillations. The analog gauges have interactively modifiable scales, hands, and labels, the digital gauges easily adapt to arbitrary ranges.

The actual creation of a circuit composed of *icons* representing its components is done by using the mouse as a pointing and dragging device. *Pop-up-sheets* are used to change the attributes of the components of the circuit and the settings of the instruments, *softbuttons* to start or continue the simulation. Before or after the simulation, the instruments, also represented as icons or, when active, as *graphic windows*, may be connected to the components of the circuit to measure voltage and current of the devices or, in case of the frequency analyzer the transfer function of the four-pole. The interaction with the system is largely determined by the intentions and actions of the user. The whole screen serves as a design space within which the effects of actions taken by the user are rarely local to specific regions. Therefore, a full sample session with the system cannot be illustrated in a few snapshots.

Supporting the Design Process

Systems like ELAB that model a complete application domain turn out to be difficult to operate although they use direct manipulation as their prime interaction technique. HPC and (direct manipulation) solve the problem of interaction with the objects of a simulated electronics laboratory, but there are still lots of details that are difficult to understand and (even more difficult to adjust to appropriate values. To run meaningful experiments within a simulated electronics laboratory requires a lot more than the ability to hook graphically represented objects together and push the run button to start the simulation. Combining parts into circuits involves knowledge about types and properties of circuits; the highly specialized vocabulary developed within electrical engineering is the level of abstraction

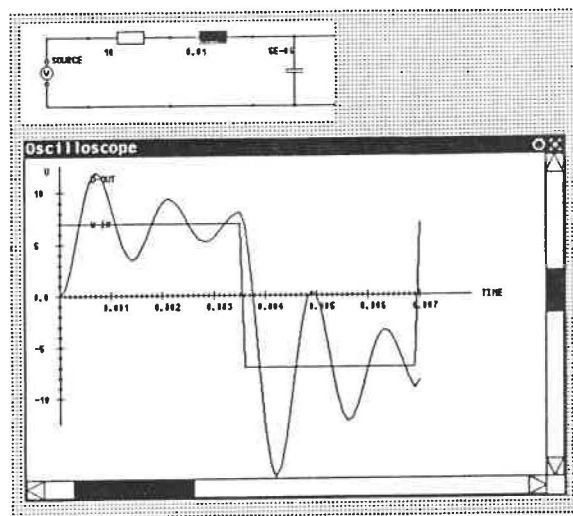


Figure 2: Transient Analysis of a Resonant Circuit Controlled by the Simulation Expert

and communication that the user as well as the system have to be familiar with. At least two kinds of knowledge are involved: knowledge about the application domain, e.g., specific four-poles, and knowledge about how to operate a simulated laboratory, e.g., how to set up and run meaningful simulations.

To overcome some of the problems encountered in communicating with the objects of the complex application domain we augmented ELAB by an expert component. Among

other things, this component monitors the user's actions, controls simulation parameters, and adjusts the displays of the measuring instruments to guarantee a good overview of the sampled information; e.g., it may be asked to operate the oscilloscope, a rather complete tool in itself.

Functionally, the expert component consists of three major parts that roughly correspond to the different kinds of knowledge utilized by the system: the simulation expert, the laboratory expert, and the electronics expert.

The *simulation expert* manages the access to the circuit knowledge base, sets up the simulation parameters appropriately (e.g., how fine or coarse a simulation to run over time), detects abnormal situations (e.g., short circuits), and finally runs the actual simulation; we will not describe it in more detail since this knowledge is not directly accessible to the user. The decisions, however, made by this component may always be overwritten by the user if for some reasons they are felt to be inappropriate for the situation at hand. Figure 2 shows the simulation result of the transient analysis of a resonant circuit exclusively managed by the *simulation.expert*, revealing the typical behavior of this particular circuit.

The *laboratory.expert* supports the ELAB user while setting up experiments and ensures that the system is always in a consistent and reliable state. It also directs the attention of the user to critical situations that are obviously not intended, e.g., if the user starts

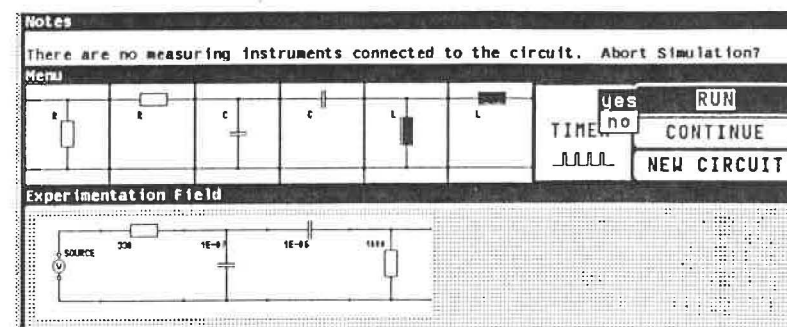


Figure 3: Critical Situation Detected by the Laboratory Expert

the simulation of a circuit without having connected any measuring instruments. In this case the laboratory expert proposes to abort the simulation (see Figure 3). To detect situations like these a continuous communication is required between the expert and the components of the laboratory environment, e.g., the measuring instruments. The expert applies heuristics to decide whether and when the actions of the user are to be interrupted. It uses monitoring techniques typically applied by active help systems; consequently, the communication incorporates elements of a mixed-initiative dialogue [6].

The *electronics expert* is a separate on-call system component that may be asked for help or assistance by the user. The range of assistance it provides to the user relates to:

- explanations about various electronics concepts or about specific electrical circuits either built by the user or by the expert. This is accomplished by allowing the user to browse through a network of electronics concepts (see Figure 4). Explanations may also be given for the tools contained in the laboratory, e.g., the frequency analyzer. Each piece of information has to be requested by the user, e.g., by pointing to the item in the expert window he is interested in. In addition, some

explanations are augmented by visual feedback for a better understanding of the explained concept, e.g., the 3dB-frequency is drawn into the frequency analyzer's diagram of the transfer function when the user asks for an explanation (cf. Figure 4).

- statements about the qualitative dependencies between the device parameters and the specific circuit parameters' e.g., how the resistance of a specific resistor affects the circuit parameters, or how the 3dB-frequency of a filter is affected by its device parameters. This qualitative information is particularly useful to understand how a circuit works, and how its behavior is influenced by the parameters of its components (see Figure 5).
- the detection and elimination of redundancy in user built circuits; the expert may suggest to simplify complicated circuits into simpler, functionally equivalent ones, e.g., by replacing two or more resistors arranged in series by one resistor. This is particularly useful for novices in the field to recognize and understand basic structural components of a circuit.
- the automatic or semi-automatic construction of circuits with specific, user-defined qualitative and/or quantitative characteristics, e.g., the user may ask the system to build a low-pass with some predefined 3dB-frequency (see Figure 6). This is a very powerful tool, since it helps the user solving problems that cannot be accomplished by experimentation in a real laboratory. ELAB also contains some simple mechanics to tailor the system's behavior to specific users. Predefined user stereotypes, e.g., novice and expert, may be inspected, modified, and/or saved for future sessions by the individual user to fit personal preferences.

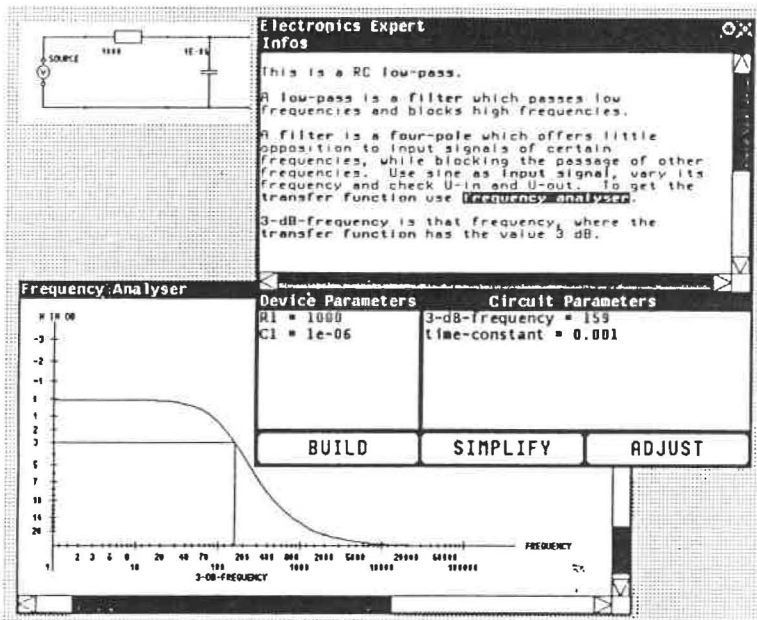


Figure 4: Browsing through a Network of Electronics Concepts

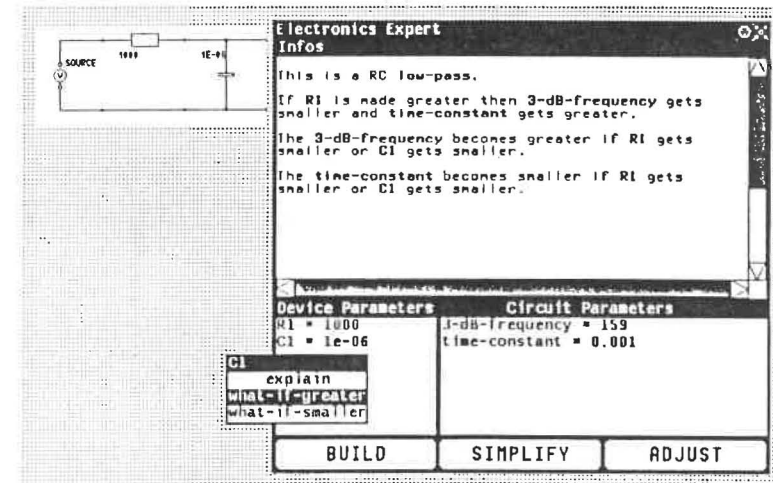


Figure 5: Qualitative Explanations of Circuit Dependencies

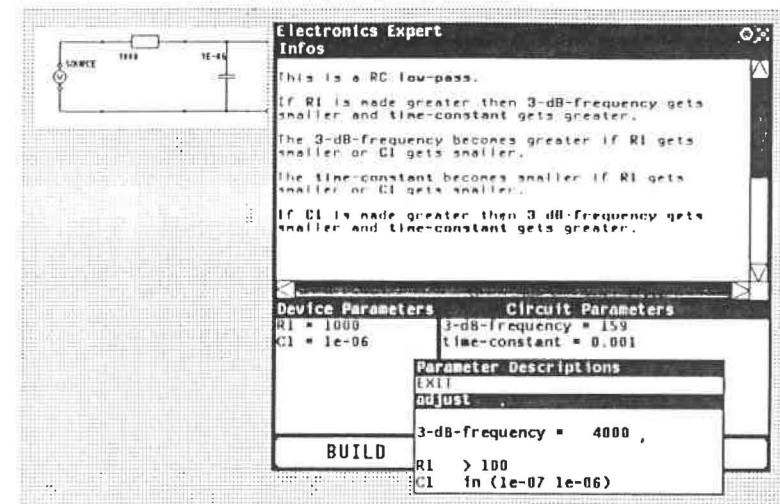


Figure 6: Adjusting the Parameters of a Circuit According to User Defined Constraints

Knowledge Representation

Most of the knowledge contained in ELAB is encapsulated within objects, classes and instances, of ObjTalk, an object-oriented extension [12] to FranzLisp. An object is described by attributes representing its state and methods representing its behavior. The fourth elementary four-pole of the circuit in Figure 3, for example, is represented by an object with the following attributes):

Type = resistor
 Orientation = parallel
 Quantity = R
 R = 1000
 Circuit = <some-circuit>
 Position = 4
 Icon = <some-icon>
 Sheet = <some-sheet>

the four-pole consists of a resistor
 the resistor is arranged in parallel
 the characteristic quantity is resistance R
 resistance in Ohms
 the circuit which the four-pole belongs to
 the four-pole is in 4th position
 icon to visualize the four-pole
 sheet to modify four-pole attributes

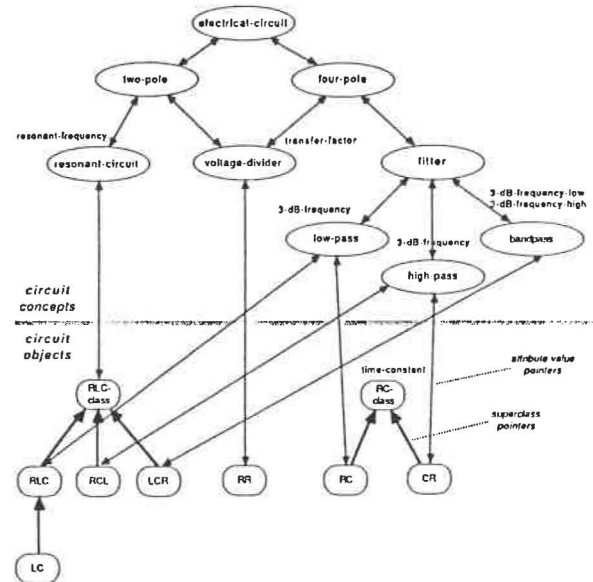


Figure 7: Object Network Representing an Extract of the Knowledge Base

Classes are organized into an *inheritance network*, instances are connected via attribute value pointers. Figure 7 shows an extract of the object network forming the knowledge base of ELAB. It consists of a collection of circuit objects and circuit concepts representing different kinds of *declarative* and *procedural knowledge* utilized by the simulation and expert components.

Circuit objects describe the simulation behavior of specific circuits, used by the simulation component when performing the transient and frequency domain analysis, as well as formal descriptions of how characteristic circuit parameters, e.g., the 3dB-frequency, are computed from the parameters of the electrical devices, i.e., resistance, capacitance and inductance. This knowledge is used by the electronics expert to derive qualitative dependencies between parameters and qualitative solutions for constraint problems, posed by the user, by means of numerical evaluation and simulation.

Circuit concepts describe qualitative attributes of electronics concepts by textual descriptions and references to related concepts. This kind of knowledge is utilized by the electronics expert to give explanations for electronics concepts related to a specific circuit. The concept low-pass, for example, has a pointer to the concept 3dB-frequency, a characteristic parameter for this kind of circuit, and pointers to the more general

concept *filter* and to specific circuit objects, such as RC- or RLC-circuit. Information belonging to a specific circuit is found by navigating through the object network starting at the corresponding circuit object following different pointers. Describing concepts by more general concepts corresponds to the principle of inheritance in hierarchical networks.

The knowledge base of ELAB may be easily extended by adding new circuit objects or concepts without having to change the simulation or expert components.

Conclusions

We have built a still evolving, prototypical system that combines knowledge representation and visualization techniques to reach a new level of human - problem-domain communication. This is achieved by adding several expert components, thus introducing domain-oriented knowledge, to an already existing laboratory environment. The experience gained with the ELAB system has demonstrated, that the usability of construction kits and exploratory environments increases by an order of magnitude when they are supported by problem-oriented knowledge enabling the user to communicate in familiar terms. Novices in the application field and electronics experts working with the system confirmed the qualification of ELAB for a tutoring system as well as a problem solving tool for this particular application domain. Using the same basic idea we have experienced a similar increase in usability with an editor for conventional music notation [1].

The ELAB system is currently implemented in FranzLisp using a window system [7] tailored to BitGraph terminals. The window system as well as the knowledge base of the ELAB system are built on top of ObjTalk. We are currently working on a reimplementations of the system using X-Windows, Common-Lisp, and CLOS.

References

- [1] Böcker, H.D., & Mahling, A. (1988). What's in a Note? In *Proceedings of the International Computer Music Conference '88* (pp. 166-174). GIMIK, Cologne.
- [2] Borning, A. (1979). *Thinglab - A Constraint-Oriented Simulation Laboratory* (Technical Report No. SSL-79-3). Palo Alto, Ca.: Xerox Palo Alto Research Center.
- [3] Brooks, F.P. (1988). Grasping Reality through Illusion - Interactive Graphics Serving Science. In E. Soloway, D. Frye, & S. B. Sheppard (Eds.), *Proceedings of CHI* (pp. 1-11). New York: ACM.
- [4] Brown, J.S., Burton, R.R., & de Kleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In D. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems* (pp. 227-282). London, New York: Academic Press.
- [5] Budge, B. (1983). *Pinball Construction Set* (Computer Program). San Mateo, Ca.: Electronic Arts.
- [6] Carbonell, J. (1970). *Mixed-Initiative Man-Computer Instructional Dialogues* (Technical Report). Cambridge, MA: BBN.

- [7] Fabian, F., & Lemke, A.C. (1985). *Wisp Manual* (Technical Report No. CU-CS-302-85). Boulder: University of Colorado (translated by V. Patten & C. Morel).
- [8] Fischer, G. (1989). Human-Computer Interaction Software: Lessons Learned, Challenges Ahead. *IEEE Software*, 6, (1), 44-52.
- [9] Fischer, G. (1987). Making Computers more Useful and more Usable. In G. Salvendy (Ed.), *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems. Proceedings of the Second International Conference on Human-Computer Interaction*, 2, (pp. 97-104). Amsterdam - Oxford - New York.
- [10] Fischer, G & Lemke, A. (1988). Construction Kits and Design Environments: Steps Toward Human Problem-Domain Communication. *Human-Computer Interaction*, 3, (3).
- [11] Fischer, G., & Morch, A. (1988). CRACK: A Critiquing Approach to Cooperative Kitchen Design. In *Proceedings of the International Conference on Intelligent Tutoring Systems* (pp. 176-185). New York: ACM.
- [12] Girgensohn, A. & Rathke, B. (1988). ObjTalk- version 15.15. Research Group INFORM, Institut für Informatik, Universität Stuttgart, inform manual edition.
- [13] Hutchins, E.L., Hollan, J.D. & Norman, D.A. (1986). Direct Manipulation Interfaces. In D. A. Norman & S. W. Draper (Eds.), *User Centered System Design: New Perspectives on human-Computer Interaction*, Hillsdale, N.J.: Lawrence Erlbaum Associates.
- [11] Mallone, T.W. (1983). How do People Organize their Desks? Implications for the Design of Office Information Systems. *ACM transactions on Office Information Systems*, 1, (1), 99-112.
- [15] Norman, D.A. (1986). Cognitive Engineering. In D. A. Norman & S. W. Draper (Eds.), *User Centered System Design. New Perspectives on Human-Computer Interaction* (pp. 31-61). Hillsdale, N.J.: Lawrence Erlbaum Associates.
- [16] Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer*, 16, (8), 57-69.