EDUCATING REASON.
FROM CRAFT TO TECHNOLOGY

Bertil Rolf

ABSTRACT
In designing, developing and evaluating educational software supporting reasoning, three aspects are important. The first is the cognitive domain of practice, relating to the nature of reasoning and the idea of support. The second aspect relates to the professional and educational objectives and settings, in particular coupling reasoning software to educational modules. The third aspect relates to economics: market, distribution and user economy. Two scenarios of development for reasoning software can be extrapolated. In the first scenario, software is seen as an add-on to present day textbooks in critical thinking.

KEYWORDS
Reasoning, argumentation, software, design, teaching, learning

INTRODUCTION: TECHNICAL PROSPECTS AND EDUCATIONAL REALITIES

Science is not a collection of facts, nor a collection of theories. Science typically links facts and theories via inferential chains. Some major examples are explanations and predictions or a reduction of one theory to another. To understand science, therefore, students will have to learn scientific reasoning.
A feasible response for higher education is to change educational technology, process and organization. By designing software packages, reorganizing student learning and reshaping teacher roles, it might be possible to transform parts of the traditional art of reasoning to skills that can be taught in modern mass education. Educational software is part of such a change.

The Cognitive Domain. Reasoning As Science Or Art

Reasoning and argumentation are arts rather than application of a scientific theory. By a theory, we mean a set of axioms and all consequences following from them by means of some rule of inference.
By an "art" we refer to the exercise of precepts distinguishing better reasoning from worse. These precepts might be exercised in reasoning conduct either consciously or as a kind of implicit ("tacit") knowledge (Rolf, 1991). Developing science based software for reasoning is different from developing software based on art. A complete, formalized theory can often be programmed into software packages implementing the theory. Even with a successful science based software package good practice of decision making involves many non-formalized features related to the process of decision making e.g. getting clear about one's values, surveying all relevant alternatives, describing them adequately, establishing the structure relating the factors and getting correct indata. These features of use go beyond formalized theory. A real life user would benefit deeply not from the program in isolation but rather the complete set of methods for getting from the state of ignorance or confusion to a robust, well founded decision.

Science is not irrelevant but there is not one but several scientific disciplines that can contribute with concepts and theories relevant for developing software and education. Even so, no present scientific results exhaust principles of software packages. There is, at present, no such theory containing all principles recognized to underlie valid reasoning.

The implication both for science based and art based software is that both need to be coupled to educational modules in order to have much user value. These modules would employ software and education together in order to transform users into skillful decision makers.

Intelligence Possessing And Intelligence Enhancing Systems

A fruitful distinction for the development of reasoning software is between *intelligence possessing* and *intelligence enhancing* systems (van Gelder, 1998). The intelligence possessing systems have, at some place, formalized and programmed machinery that takes over user operations or computations and presents the user with an outcome. Intelligence enhancing systems always leave the user in command of each step in the elaboration of the argument. In this way, they act as a lever on the user's own intelligence. An example of an intelligence possessing system is an algorithm, deriving a proof sequence for a theorem of propositional calculus.

Intelligence enhancing systems give users, as it were, a lever on their natural intelligence. They do so by supplying representative diagrams or by facilitating the user's working process. These representations present information about relations in the form of visual relations that can be taken in at a glance (Larkin and Simon, 1987). By externalizing and standardizing of what once were mental operations of a user, the software facilitates reasoning processes (cf. Hutchins, 1995). It is known, for instance, that users are prone to a number of errors in taking decisions when considering a multitude of cues in complex interaction. By supporting user processes of analysis and synthesis, it is believed that software supporting reasoning will facilitate stepwise processes of forming opinions or taking decisions in complex issues (Dawes, 1988).

The distinction between intelligence possessing and intelligence enhancing systems is one of degree rather than of kind. The user of intelligence possessing systems needs ability to construct structures underlying beliefs and decisions and she will have to be able to select good models and transform real life facts into computational form that the algorithm can recognize. Some of the intelligence enhancing systems make use of elementary algorithms for representing structure or for filtering away bad arguments.

Not all reasoning can be supported through intelligence possessing systems. The field of reasoning goes far beyond what is computationally possible. Intelligence possessing systems can therefore not represent all kinds of reasoning. For instance, legal reasoning where the pros and the cons have to be balanced is a field where no plausible formalization or algorithms have been proposed (Schum, 1994).
The two kinds of software are related to different processes of design and development. Intelligence possessing systems are based on an axiomatic theory that is formalized and programmed. Such development can, in principle, be performed without much user tests. Programming largely consists in building an algorithm that implements the theory.

The development of intelligence enhancing systems is based on user studies of one form or another. Teaching tradition, using reasoning diagrams, has been a major source of influence. The diagrams take the form of nodes, representing theses, and connections, representing logical relations. Together, they form a tree or semi-tree that represents a standpoint where one or several conclusions are based on a collection of premises. The diagrams facilitate reasoning. They externalize and visualize interrelations between the buildings blocks of a complex standpoint. Furthermore, the process of reasoning is also broken down to a manifest step-by-step procedure that an actor or a team can perform.

The basic idea of such diagrams is threefold. First, an inner mental representation is replaced by an external, visual representation. Second, inner operations of inferring and concluding are represented by intersubjectively observable moves on a screen. Third, group collaboration and teacher interaction is facilitated by the use of a common symbolism for representing logical units (premises and conclusions in the form of nodes) and logical relations (connections between nodes).

There is no fixed set of principles for design and implementation support for visualization or for collective reasoning processes. Design and development, therefore, are largely experimental, based on observations and feedback from users. The process is iterative. Such development presupposes access to qualified users facing qualified tasks that are hard to solve without software use. In practice, such software development can hardly be conducted outside educational contexts where a teacher assigns tasks to students.

General versus Domain Specific Reasoning Software

Another division of software types is between general and domain specific software. In the 1960's it was believed that Artificial Intelligence could deliver general problem-solving software, independently of the domain under consideration. These beliefs were over-optimistic in that field. Instead, software for supporting inferences is now tied to particular domains. Reasoning in medicine, geology or chess can be supported by software employing assumptions drawn from the respective domains. The programming of Bayesian networks can perhaps change the balance back towards general software packages, but on the whole, it is an open matter.

Intelligence enhancing software still tends to be domain independent. Such software derives from teaching traditions and there, it is generally believed that argumentation, reasoning and critical thinking are domain independent skills for human thinkers. The assumption is more of a tacit presupposition than an openly defended thesis. Capacity for reasoning in various domains seems to be correlated. Those who perform well in one domain perform well in other domains, irrespective of their professional domain. Groups of philosophers perform well in causal reasoning across domains (Kuhn, 1991).

In programming some of the cognitive precepts and capacities underlying reasoning practice, the aim is not merely to "translate" a subset of existing practice into software. Cognitive artifacts do not merely enhance user capacities to solve previous tasks. They also extend or redefine the tasks at hand. The interesting unit is not the software package in isolation but the set of user plus artifact. Together as a unit, they accomplish more or perform better than one of them would do in a setting without the other part (Norman, 1993).

This does not imply that patterns of reasoning are stored in the form of general theorems, syllogisms or schemata. There is considerable evidence that in reasoning, people have difficulties recognizing and applying general forms or schemata of reasoning (Gigerenzer and Hug, 1992). Reasoning, if it is exercised as a general capacity, is probably rather generalized from old examples to new examples.

A Survey of Two Elementary Level Software Packages

Below, I will compare some of the features of software packages that are around. I will focus on general, intelligence enhancing systems for supporting or teaching reasoning (cf. Kirschner, Buckingham-Shum and Carr, 2002). These software packages supporting reasoning and argumentation are based on intellectual methods where the use of diagrams plays a major role. The inner operations of intelligent performance in reasoning are externalized by visual representation, permitting robust and retraceable working procedures (cf. Hutchins, 1995). The interfaces of the two programs are shown below.

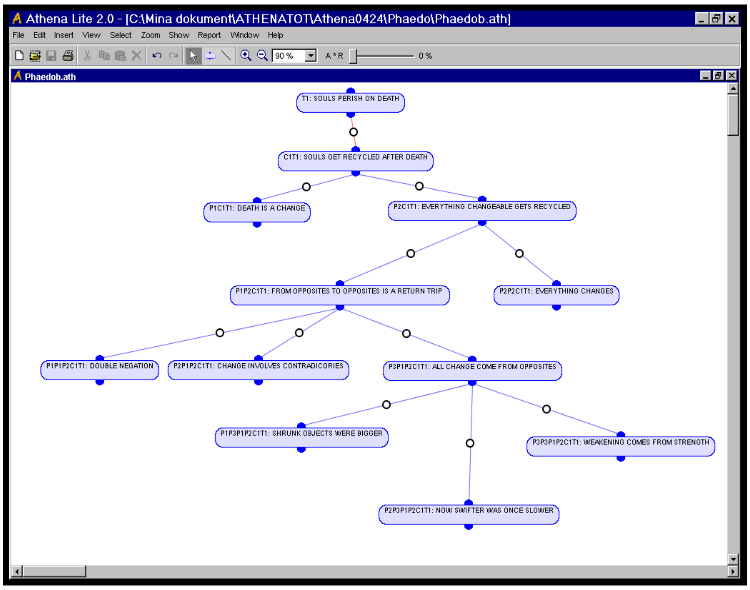

Figure 1. Athena interface. Software available at http://www.athenasoft.org

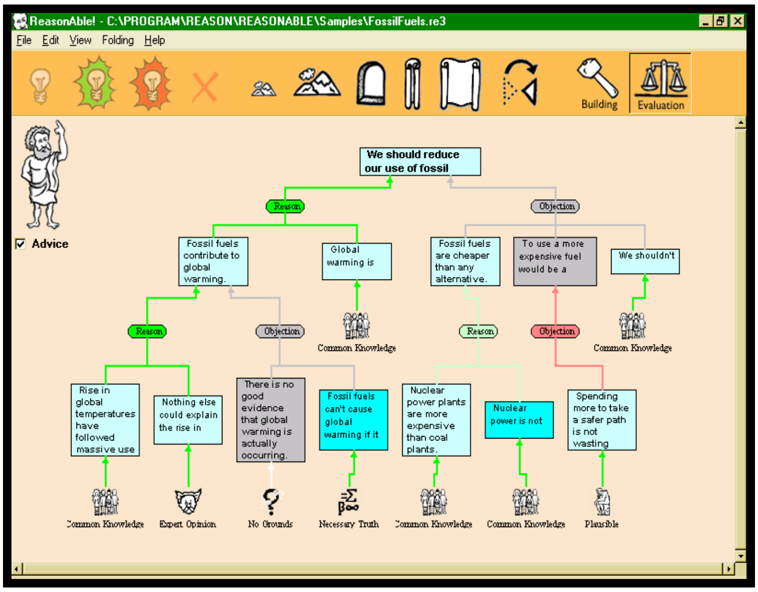

Figure 2. Reason! Able interface. Software available at http://www.goreason.com/

The similarities of these programs are striking, although they were produced in diverse context. These similarities owe to their heritage in reasoning diagrams, cultivated in the traditions of reasoning, argumentation and critical thinking (Naess, 1966; Toulmin, 1988; Scriven, 1976). The textbooks and their connected teaching methods have been a source of inspiration for the programs. There is evidence that such is indeed the case (Reimann, 2000; Donohue, van Gelder, Cummings and Bissett, 2002).

The Professional and Educational Embeddings of Reasoning Software

All programs of this type are based on the assumption that beliefs and decisions based on analysis plus synthesis have better grounds than those reached by a holistic policy for forming beliefs and taking decisions. There is fairly strong evidence that holistic policies fare badly when actors form beliefs and take decisions. In comparison, the process of analysis and synthesis would seem to improve the rationality of standpoints. Some method for analyzing complexes into factors and combining them will tend to do better than intuitive overall judgments in most expert domains. Although some of the pessimism of the early years, e.g. of Nisbett & Ross (1980), has been criticized (Gigerenzer, 1991), there remains considerable negative findings about the unguided reasoning capacities of men (Samuels, Stitch and Tremoulet, 1999).

A plausible conjecture is what we may call *the embeddedness conjecture.* It is not the software itself that creates learning effects but the tasks for which it is used. These tasks are defined by the education or social setting of which the use of the software is a part. This conjecture is a generalization of a finding by Veerman who has performed a study on reasoning software for secondary education named Belvedere. His study indicates that student learning depends on the social task and interaction rather than on software alone or teacher influence. Argumentative activities depend above all on task design and task characteristics (Veerman, 2000; Veerman, Andriessen and Kanselaar, 1999).

Most of the qualities of the user's activities are determined by educational tasks and the aptness of the software tools, less by factors that can be influenced after the predefinition of the software interface and capacities and the tasks for which it is used. It is probably advantageous to consider the skills of reasoning and argumentation as "mildly" situated. We expect capacities for reasoning to generalize from some social contexts to others. In this sense, capacities for reasoning might not be *strongly* situated, i.e. tied only to the same type of social setting (van Gelder, 2000).

Transferability

A second educational assumption has to do with *transferability*. It is, obviously, of great importance that courses promoting reasoning and argumentation transfer so as to give beneficial effects in other courses or in professional activities. In spite of attempts to verify such effects from traditional methods of teaching critical thinking, none has so far been found. It is unclear whether, and to what extent, reasoning skills are improved through courses particularly designed for that purpose, e.g. courses in logic or in critical thinking. After reviewing evidence of effects of courses in critical thinking (CT), Tim van Gelder concludes:

> Currently it is difficult to make a convincing case …that CT courses are of any substantial benefit. On one hand there are various studies indicating no significant benefit from CT instruction. On the other, there are some studies which do appear to find some benefit. … The belief, common among CT teachers, that CT courses are better for improving CT than formal logic courses does not appear to be supported by the available evidence, such as it is….An important question, which is left unresolved by these studies, is whether CT courses harm their students. It appears possible that typical CT courses actually reduce CT performance (van Gelder, 2000).

College education has a general effect on reasoning skills, but it is an open matter whether courses particularly designed for the furthering of such skills have the intended effect or even the opposite effect. Whether courses in formal logic have any effect is similarly an open matter.

At present, there is considerable need for articulating what the educational goals for teaching reasoning are. We also need to consider which conjectures are made about transfer from educational tasks solved by students to future tasks facing them. I can see four types of transfer assumptions:

- Transfer from analysis taught to analysis exercised.
- Transfer from synthesis (production) taught to synthesis (production) exercised.
- Transfer from analysis taught to synthesis (production) exercised.
- Transfer from synthesis (production) taught to analysis exercised.

First, we can consider assumptions about transfer from analysis taught to analysis exercised. In critical thinking, students are generally presented with tasks where they analyze and evaluate actual or hypothetical examples. In the background often hovers some kind of "consumer information" perspective behind the courses. If students are taught methods for informing themselves in the role of argument consumers, their future consumption of arguments will live up to the same standards as those taught in critical thinking. The results of van Gelder's survey throw doubt on that presupposition (van Gelder, 2000; Donohue, van Gelder, Cummings and Bissett, 2002).

Second, we might consider transfer from synthesis taught to synthesis exercised. An assumption common to classical rhetoric and to the teaching of speech is that the capacities for finding, structuring, elaborating on good arguments is a general, domain independent capacity. There is some indication that this kind of transfer can occur in a setting where students practice realistic, complex tasks producing arguments, keyed to role performance (Voss, Wiley and Carretero, 1995). Such teaching where role-play is integrated into structured feedback has been found to be an effective method for the training of practical skills (Argyle, 1986).

A third kind would relate to a transfer from analysis taught to synthesis exercised. This would be the case if we assume that the teaching of critical thinking, mainly consisting of analysis, carries over to the improvement of future argument production in professional roles. I know of no studies of this kind of transfer. But we may consider such transfer a risky conjecture, given the absence of evidence even of transfer from analysis to analysis. Furthermore, real life success in argument production involves situational awareness and the enaction of roles in socially dynamical contexts. It is hard to see if or how analysis of written texts could carry over to oral performance under such conditions.

Finally, transfer could relate to a transfer from synthesis taught to analysis exercised. There is some evidence that such transfer occurs (Allen, Berkowitz, Hunt and Louden, 1999). At first, this might seem surprising, given the difference between production taught and analysis exercised. But most qualitative interaction in professional argumentation involves attempts to understand others and counter their arguments. By broad and varied experience, it might be possible that analysis to some extent is acquired along with capacities for production. Furthermore, there are the general findings of Deanna Kuhn (1991) that higher education improves *general* understanding of causal reasoning.

Production Factors

Some of the criteria for evaluating educational technology draw on *market, distribution and user economy*. Developing software draws considerable investment in labor and costs. To these costs are added costs for educational development.

Suppose that we have an innovation in educational technology that saves 20% of teacher time on a given course module with 30 students and five weeks of student work. If developments costs are to be financed, some 50 courses need be given to reach break even with a program like Athena. Alternatively, if we want students to purchase the program (and modules) at a cost corresponding to textbooks, between 5.000 and 10.000 student buyers are needed to reach breakeven.

These rough figures indicate that the market cannot feed more than a few developers, given the present time-consuming methods for software development. These developers will either be large commercial actors or projects sponsored by governments or by universities for strategic reasons rather than profitability.

Division of Labour. Craft versus Industry

The arguments for educational software seem to be quite strong. For instance, in research or technical development, any piece of work builds upon works of others. Research results refer us to other research results, provoking questions, proposing hypothesis, supplying methods and so forth. Technical development is largely based on reversed engineering, deconstructing the innovations of other in order to discover the hidden workings.

But in the teachings of higher education, there is at present a deep-seated conservatism working against introduction of software. Very little of a division of labor, learning from others, seems to take place in design, development and application of educational technology. Although there are several developers and vendors of software tools, few teachers actually use other learning tools than those that are homemade (Alexander, 2002).

While the development of research and technology have an industrial character, education still carries the mark of a personal craft, carried out under myths of personal relations between students and teachers. Teachers use textbooks written by others but seldom import other material into their courses. Teachers tend to personalize their relation to educational material. This is a constraint on the use of other-produced material, methods or intellectual tools.

While I agree on the personal nature of the relation that teacher and students have to their materials, methods and software tools, I see no reason to assume or demand that that relation be the same in teacher and student. Nor do I think that the teacher needs to exercise control over the student's learning process. Students are different and their learning trajectories are different.

The resistance against educational technology has, in my view, deep roots in traditional conceptions about teacher role, responsibility and relation to students. While learning is highly personal in relation to the *content* of knowledge, I see few good reasons for personalizing teacher relations to course content. My conception of the teacher is of one who lets students loose in an intellectual learnscape, guides their tasks and efforts by feedback and controls that they have learned some of the right things to learn.

Strategies and Two Scenarios

Scientific and educational meta-traditions are bearers of the old arts of reasoning and argumentation. These meta-traditions were part of implicit learning or tacit knowledge in the old elite university. As such, they cannot be extended to large and varied student groups in modern mass higher education.

Attempts to package the arts of reasoning and argumentation in suitable forms for modern mass education is problematic. The educational results of ordinary courses in critical thinking are uncertain. Software and modern technology suggests themselves as media in a higher education where teacher time and teaching costs are scarce resources.

But although higher education in principle would need shortcuts to skills in reasoning or argumentation, such needs might not in practice channel into demands and a flourishing market. We can imagine two scenarios for the introduction of software into higher education.

The first we can call *the textbook* scenario. It extrapolates the present day, individualistic patterns of decision making. Today, teachers of higher education hold the gate to the educational market. Decision-making today is decentralized to individual, isolated teachers, laboring under considerable teaching loads. It does not seem realistic to expect major bottom up reforms of teaching methods in critical thinking. Given this decentralized decision making in labor intensive, underfinanced teaching activities, software can at best trickle down slowly along the same paths and with content equivalent to the present day channels of distribution. This is, roughly, that of textbooks in argumentation or in critical thinking. The future of software in the education of reasoning is then projectible from the introduction of textbooks in the past.

Under this first scenario, software for critical thinking might be conceptualized as an add-on to ordinary textbooks, an extra flavor as it were. The software will be integrated into present day, individualistic teaching methods. Textbooks determine the time structure of the courses. The cost of the software is also an add-on on top of the textbooks.

Such a scenario is an example of what Schön (1973) has called "dynamic conservatism". It is a step-by-step modification of present day practice, motivated by a desire not to lag behind others. No radical analysis of problems and solutions is made. Hence, it does not address the major problems of higher education – scarcity and cost of teacher time, together with large and varied student input.

In this scenario, software packages will be general and applicable in several courses, making for large groups of student users. The packages will have a low user threshold to attract many users. Their educational content will probably be standardized in order to fit present day conventional courses. This content is largely defined by present day textbooks, e.g. "Critical Thinking I" or "Argumentation" or "Reasoning for nurses/teachers/social workers" such as Govier (2001) or Kahane and Cavender (2002).

The second scenario we can call *institutional development*. It shifts decision-making in higher education to a level of teams, departments or teaching institutions. Central decisions are taken to introduce software-based education in order to save teacher resources or raise (or at least preserve) the level of educational quality. Software together with educational modules is then accommodated to fit the aims of education rather than preexisting courses, course modules or textbooks. The resulting software will not be tied to particular courses, or target specific groups of teachers, e.g. those of critical thinking. Nor will these software packages be conceived of as add on to textbooks. This second scenario yields software with far more open-ended properties than the first scenario.

Software development, together with development of educational modules, will then be a strategic concern of institutions of higher education. Teaching and software development will be vertically integrated in universities. There, close collaboration between research, technical-educational development and education of end-users is located within one organization. Again, the costs are massive and there will be a few major developers/vendors around.

The second scenario depends on economic factors such as possible savings on teacher resources, preserved quality and possible methods for transferring educational modules from producers to teachers. These possibilities are yet unknown.

Comparing Software Based Approaches to Educating Reason

In planning and evaluating software-based approaches to educating reason, a number of criteria stand out. They represent features of the software, educational modules and development strategy. These features are either openly taken decisions or tacit assumptions behind the process, the product or the projection of them into the future. Some of these basic criteria are listed below.

Cognitive Criteria

- Are reasoning and argumentation arts or applications of science or a mix of them?
- Which precepts or theorems are formalized in the software?
- Which other precepts or theorems are essential for good practice?
- Are such principles integrated into user's working processes? If so, by what means are they assumed to be integrated into final usage?
- Which steps between input and output are performed without user interference ("intelligence"). At which steps are user interference needed?
- Is the software and educational package applicable to all domains of reasoning? If not, what are the restrictions?

Educational Criteria

- What are the social embeddings of the context of reasoning and argumentation? What are the relations between students and students-teachers?
- What are the student tasks performed in education?
- How are general precepts represented in education? Which of them are explicitly taught and which are taught via example?
- What are the assumptions of transfer from learning context to future context? Are there assumptions of transfer analysis-analysis, synthesis-synthesis, analysis-synthesis or synthesis-analysis?
- What are the software facilities supporting analysis or synthesis respectively?
- What other facilities, methods or procedures are assumed to support transfer?

Economic Criteria

- What is the product – a software package for students, teachers or institutions?
- Is future development financed and, if so, how?
- Marketing. Who are the decision makers?
- What is the assumption about the craft character of teaching reasoning?
- A textbook scenario versus a scenario of institutional development.

An evaluation of the software packages in isolation will tell us little of their past, present and future. Their success or failure depends not on technical factors but on educational factors and the strategies for meeting user demands. Such demands can be channeled in several directions and prognosis today is uncertain. Two major scenarios are plausible. One sees software as an add-on to textbooks, the other sees software as a strategic investment for universities.

Under both scenarios, software and education need to be developed under integrated concepts. The target of development is the skillful decision maker rather than a software package with such and such features. All software packages with this aim will be coupled to educational modules, steering the user process. Not everything need to be built into the software itself but various educational modules can share the burden of enabling users to reach higher levels of skill.

Second, development will have to be an iterated process where educationalists and technicians cooperate. Desirable software features can be discovered and implemented only after the software has been put to user test. It is an open matter whether improvements in operation should be implemented in the software or merely in user processes steered via instructions and education.

Third, projects developing software packages for reasoning support need to integrate educationalists and technicians. Both parties are necessary for the final integrated product. This implies a close cooperation within university departments or between university departments and software firms.

REFERENCES

Alexander, S., (2002). “Will Pedagogy Drive technology or the Reverse?”, Key Note Speech at Netlearning 2002, 24-27 November, 2002, Ronneby, Sweden.

Allen, M., Berkowitz, S., Hunt, S., and Louden, A., (1999). “A Meta-Analysis of the Impact of Forensics and Communication Education on Critical Thinking”, Communication Education, 48, 18-30.

Argyle, M., (1986). The Psychology of Interpersonal Behavior, Penguin, Harmondsworth, Middx., U.K.

Dawes, R., (1988). Rational Choice in an Uncertain World, Harcourt Brace Coll. Publishers, Orlando, Fla., USA.

Donohue, A., van Gelder, T., Cummings, G., and Bissett, M., (2002). Reason! Project Studies, 1999-2002 (Reason! Project Technical Report 2002/1.) Dept. of Philosophy, University of Melbourne, Australia.

Gigerenzer, G., (1991). How to Make Cognitive Illusions Disappear: Beyond 'Heuristics and Biases.', European Review of Social Psychology, 2, 83-115.

Gigerenzer, G., and Hug, K., (1992). Domain-Specific Reasoning: Social Contracts, Cheating and Perspective Change, Cognition, 43, 127—171.

Govier, T., (2001). A Practical Study of Argument, 5th ed., Wadsworth, Thomson Learning, London, U.K.

Hutchins, E., (1995). Cognition in the Wild, The MIT Press, Cambridge, Mass., USA.

Kahane, H., and Cavender, N., (2002). Logic and Contemporary Rhetoric, 9th ed., Wadsworth, Thomson Learning, London, U.K.

Kirschner, P.A., Buckingham-Shum, S.J., and Carr, C. S., (eds.), (2002). Visualizing Argumentation, Springer Verlag, New York, N.Y., USA.

Kuhn, D., (1991). The Skills of Argument, Cambridge University Press, Cambridge, U.K.

Larkin, J., and Simon, H., (1987). ”Why a Diagram is (Sometimes) Worth Ten Thousand Words”, Cognitive Science, 11, 65-99.

Naess, A., (1966). Communication and Argument. Elements of Applied Semantics,Universitetsforlaget, Oslo, Norway.

Nisbett, R., and Ross, L., (1980). Human Inference: Strategies and Shortcomings of Social Judgment, Prentice-Hall, Englewood Cliffs, N.J., USA.

Norman, D., (1993). Things That Make Us Smart, Perseus Books, Mass., USA.

Reimann, P., (2000). “Fostering Critical Thinking about Environmental Issues With „Mind Tools’”, in: Tankar om lärande och IT - en forskningsöversikt, Skolverket, Stockholm, Sweden.

Rolf, B., (1991). Profession, tradition och tyst kunskap (Profession, tradítition and tacit knowledge), Nya Doxa, Nora, Sweden.

Samuels, R., Stitch, S., and Tremoulet, P. (1999). Rethinking Rationality: From Bleak Implications to Darwinian Modules, in I. E. LePore and Z. Pylyshyn (eds.), What Is Cognitive Science?, Blackwell Publ., Oxford, U.K.

Schön, D., (1973). Beyond the Stable State, Penguin, Harmondsworth, Middx., U.K.

Schum, D., (1994). The Evidential Foundations of Probabilistic Reasoning, Northwestern University Press, Evanston, Ill., USA.

Scriven, M., (1976). Reasoning, McGrawHill, New York. N.Y., USA.

Toulmin, S.E., (1988). The Uses of Argument, 9th ed., Cambridge University Press, Cambridge, U.K.

van Gelder, T., (1998). ”Penicillin for the Mind? Reason, Education and Cognitive Science", Preprint 1/98, Dept. of Philosophy, Univ. of Melbourne, Australia.

van Gelder, T., (2000). The Efficacy of Undergraduate Critical Thinking Courses. A Survey in Progress, Report, Dept. of Philosophy, Univ. of Melbourne, Australia.

Veerman, A.L., (2000). Computer-Supported Collaborative Learning Through Argumentation, Print Partners Ipskamp, Enschede, The Netherlands.

Veerman, A.L., Andriessen, J.E.B., and Kanselaar, G., (1999). Collaborative Learning Through Computer-Mediated Argumentation, in C. Hoadly and J. Roschelle (eds.), Proceedings of the Third Conference On Computer Supported Collaborative Learning, Standord University, Palo Alto, Calif., USA.

Voss, J.F., Wiley, J., and Carretero, M., (1995). ”Acquiring Intellectual Skills”, Annual Review of Psychology, 46, 155-181.

Professor Bertil Rolf
Blekinge Institute of Technology
Box 520
S-372 25 Ronneby
Sweden.
Email : bertil.rolf@bth.se.