Gaines, B. R. and Shaw, M. L. G. (1999). Embedding formal knowledge models in active documents. Communications of the ACM 42(1), 57-63.

# Embedding Formal Knowledge Models *in* Active Documents

***Creating problem-solving documents.***

Brian R. Gaines and Mildred L.G. Shaw

Computational symbolic models are powerful tools for making ideas explicit because they support the formal definition of all the terms in an argument, and these formal definitions may be used to compute the consequences of a set of arguments. In particular, once ideas are fully explicit in symbolic form they may be used as part of an information system with the guarantee that the same reasoning is being applied to all cases. A typical example of where this is desirable is in the implementation of the procedures of a corporation, for example the credit-control rules of a bank or the financial authorities of a company.

STEVE ADLER

However, the formal nature of computational symbolic models is also problematic in that people find it difficult to make their reasoning processes explicit, and the models developed may be difficult to comprehend and validate. In knowledge acquisition research and applications the transition from the informal expression of knowledge that is natural to people to the formal expression of knowledge that is required for computation is recognized as a major problem [4]. Knowledge acquisition architectures have been developed that use hypermedia technologies to combine the forms of representation that are natural for people, generally media such as text, images and movies, with forms of representation that are both computational and comprehensible by people, generally semantic networks representing symbolic models in visual languages [9].

The product of knowledge acquisition is generally thought to be an expert system of some kind, and these are associated with their own interfaces to support advice giving and explanation. However, it is attractive to make the product a humanly-understandable *document* that integrates informal multimedia knowledge representation with the

formal computational structures derived from it. Embedding an operational symbolic model in a document that annotates and explains it supports human users in their understanding of the model, its application and its ongoing development. For example, Wolfram's *Mathematica* supports active mathematical notebooks in which mathematical equations can be manipulated within a document, and Claerbout has developed a system for scientific active documents, which embed the raw data and the computational analysis used to enable the reader to dynamically reproduce the results [3].

This article presents a formal visual language for representing symbolic models as semantic networks that are both comprehensible by people and that compile to a computational form supporting automatic inference. The tools that support the language enable definitions and facts expressed in it to be embedded in word processing documents in an active form in which they can be edited, annotated through surrounding text and hypertext links, and exported to associated inference engines. Inferences can be returned in the visual language so that they can be examined as additions to the original model, or, in complex examples, can be returned through an advice system that answers specific queries. Examples are given of basic inferences, the form of the language, its application to complex problem solving, and the porting of the technology involved to the Web.

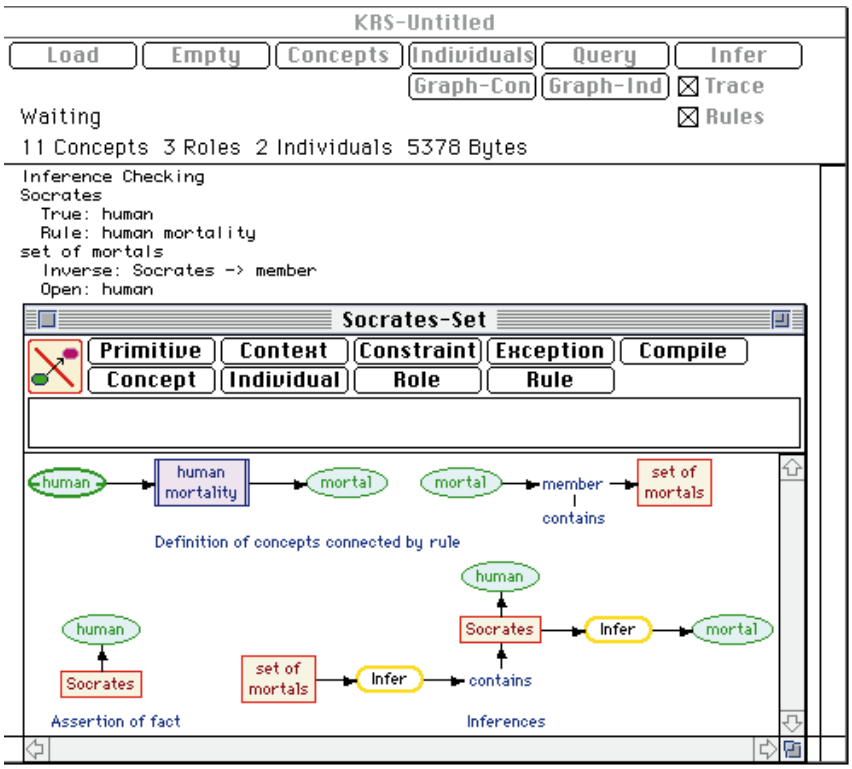


**Figure 1.** Symbolic inference from simple knowledge model

## Visual Modeling and Inference with Semantic Networks

Visual presentation of symbolic knowledge has been an attractive feature of semantic networks since their inception because the expression of such knowledge in the formal language used by expert systems is usually not very comprehensible to nonprogrammers. Early examples of semantic nets were criticized because nodes, arcs and their labels could be used ambiguously, and diagrams were subject to differing interpretations [2]. However, as formal semantics for symbolic knowledge representation languages such as CLASSIC [1] became available, it proved possible to define visual languages that had the appearance of semantic networks but translated unambiguously into knowledge representation formalisms [6].

Figure 1 shows at the lower left the window of an interactive editing tool for semantic networks in front of the window of an inference engine that can reason from a network and export its inferences back as graphic additions to the network. The example used is the classic syllogism that all human beings are mortal and this is expressed as a rule, "human mortality," whose premise is any individual satisfying the concept "human" and whose conclusion asserts that such an individual also satisfies the concept "mortal." This is expressed by the graph at the top of the window that represents the two concepts as oval shapes and links them through the rule in a rectangular box.

To make the example more interesting, "human" is defined as a *primitive* concept, that is, one whose definition is essentially incomplete, whereas "mortal" is fully defined (at the top right of the lower window) as any individual that is a "member" of the "set of mortals." Concepts are represented by ovals. Primitive concepts have inset horizontal lines on each side. Individuals are represented by rectangles,

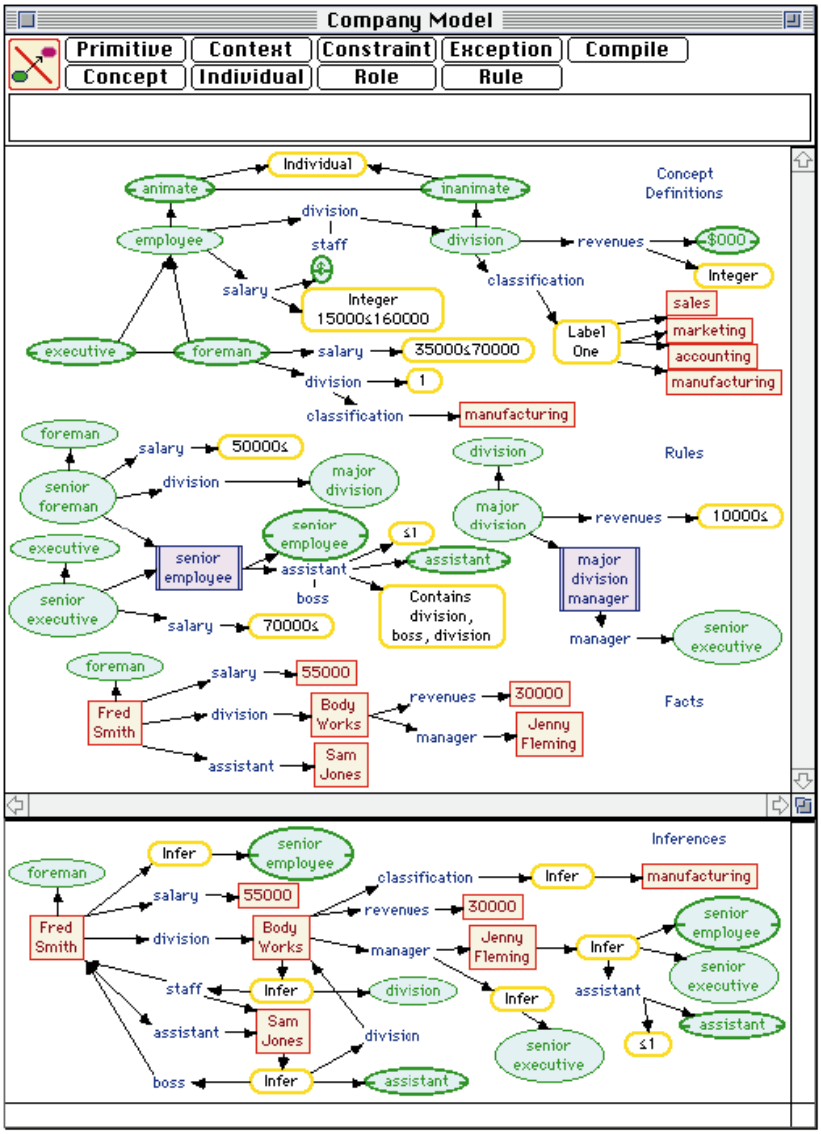


**Figure 2.** Knowledge model for a company with inferences made

and rules by rectangles with inset vertical lines on the left and right. Roles (attributes and relations) are represented by unboxed text. The line joining the roles "member" and "contains" indicates the roles are inverse, that if *x* is a member of *y* then *y* contains *x*.

The fact that "Socrates is human" is asserted at the lower left by a graph that links the individual Socrates with the concept human, asserting that the individual is an instance of the concept. When this network is exported to the inference engine running in the background it notes that it is true that Socrates is human and asserts the human mortality rule to infer that Socrates is also mortal. It also uses the inverse relation to infer that the set of mortals contains Socrates. It exports this as the graph shown at the bottom right, adding the inferences to the original graph. This simple example illustrates the basic principles that apply to more complex applications, that symbolic models can be represented as graphs and that deductive inference results in additions to these graphs.

The text in the background window traces the inference. The final line is interesting because it notes that it is "open" whether the set of mortals is human. This indicates that adding further information might prove this possibility true or false, and keeping track of open rules is important in guiding knowledge acquisition and explanation. In this case the possibility arises because the type of the individual "set of mortals" has not been specified. It would be appropriate to define a top-level ontology in which, for example, "entity" is defined as a primitive concept disjoint from "set," human inherits from entity, and the set of mortals is an instance of set. It would then be true that set of mortals could not be an instance of human, and the inference would no longer be open. This illustrates how common confusions in reasoning, such as those about the status of sets, can be exemplified and resolved appropriately.

Figure 2 shows a more complex example where the upper window describes some corporate concepts, business rules and facts, and the lower one shows the inferences returned. At the top of the window the primitive concepts "animate" and "inanimate" are defined, linked by a line that defines them to be disjoint such that no individual can be both animate and inanimate. The concepts "employee" and "division" inherit from animate and inanimate, respectively, so they are also disjoint. An employee is defined to have an attribute "division" which must be filled by individuals of type "division" and has the inverse attribute "staff," and an attribute of type "salary," which is of type "$" and filled by an integer between $15,000 and $160,000. A division is defined to have an attribute "revenues" of type "$000" which must be filled by an integer, and an attribute "classification" which must be filled by one of "sales," "marketing," "accounting," or "manufac-

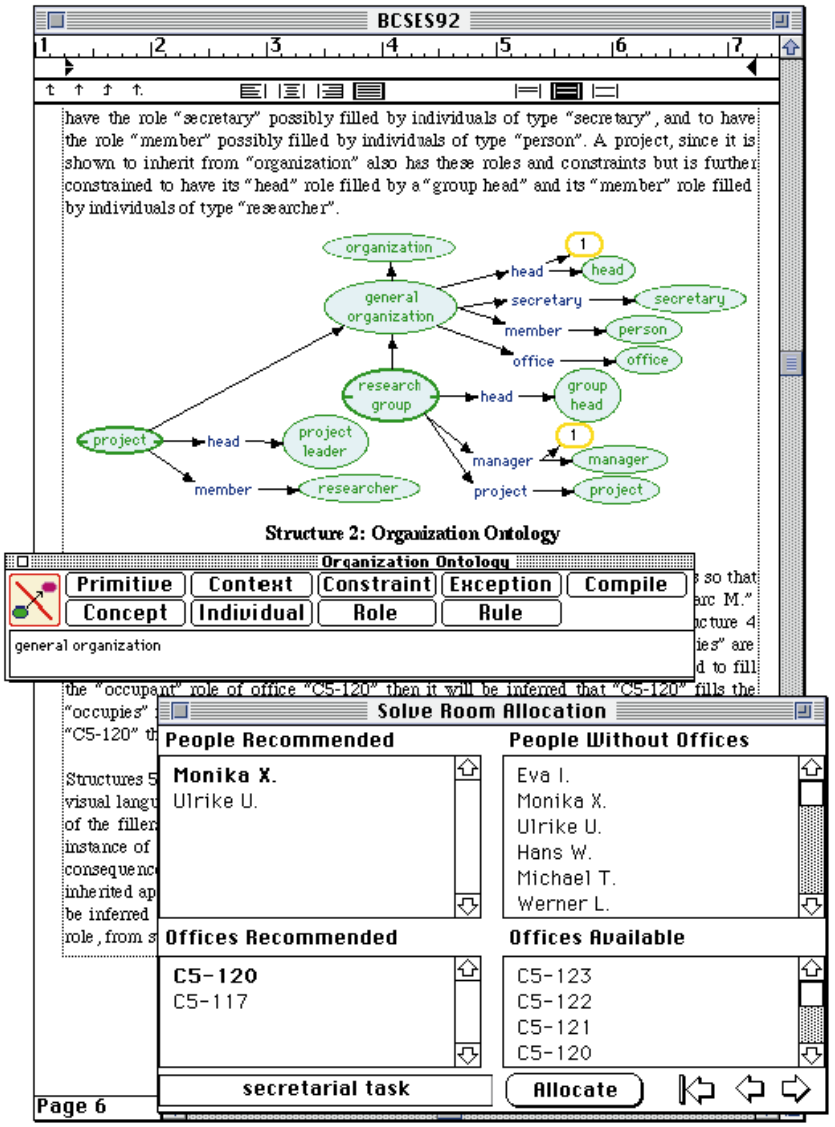


**Figure 3.** Active document with embedded expert system

turing." Below employee two disjoint subtypes of employee are defined, "executive" and "foreman," that inherit the definition of employee with foreman having the additional constraint that the salary is between $35,000 and $70,000 and that the division is filled by one individual whose classification is manufacturing.

These definitions establish precisely what the corporation for which procedures are represented means by the use of the terms "employee," "division," "executive," "foreman," and so forth. For example, if it is asserted that an individual is a foreman then it can be inferred that the person is an employee with a salary in the range of $35,000–$70,000 in a manufacturing division. The definitions act as constraints upon the entities in the corporate world and can, for example, be used to check that the defined constraints are satisfied by all the corporate entities—that someone promoted to foreman is given a salary consistent with the company norms and that the term "foreman" is not used as part of a job definition outside the manufacturing division.

Across the center of the upper window under the definitions are defined business rules using the same notation as in Figure 1 but with substantially more complex premises and conclusions. On the left, a "senior foreman" is defined as a foreman with a salary greater than or equal to $50,000 in a major division. On the right a "major division" is defined as a division with revenues greater than or equal to $10,000,000. On the left being a senior foreman is one premise for a "senior employee" rule that asserts that an individual satisfying the premise is a "senior employee" and has at least one assistant who is an "assistant" in the same division as his boss. An alternative premise to this rule is that the individual is a senior executive, which is defined as an executive with a salary greater than or equal to $70,000. On the right the "major division manager" rule asserts that any individual who is the manager of a major division is a senior executive.

At the bottom of the window beneath the rules are asserted some facts about individuals associated with the corporation. "Fred Smith" is a foreman with a salary of $55,000 in the "Body Works" division with an assistant "Sam Jones." The Body Works division has revenues of $30,000,000 and its manager is "Jenny Fleming."

The inferences that can now be made about these individuals follow the same pattern as those for the previous example but are substantially more complex and may not all be obvious at first glance since several inferences link together to develop compound conclusions. The bottom window in Figure 2 shows the graph returned by the inference engine. Using the rules shown, it has recognized that Fred Smith is a senior foreman (which also requires recognizing that Body Works is a major division) and inferred that he is a senior employee and his assistant Sam Jones is an assistant in the Body Works division with boss Fred Smith. It has inferred that Body Works has the classification manufacturing (because

**Figure 4.** Knowledge acquisition through the Web

Fred Smith is in that division and is a foreman) and its staff includes Fred Smith and Sam Jones. It has inferred that Jenny Fleming is a senior executive and senior employee and has at least one assistant.

All these inferences result from a few definitions, rules, and facts. In real applications there are many more definitions, rules and facts, and the volume of inferences rapidly becomes beyond human capacity. Definitions and rules are normally separated into coherent modules associated with different aspects of the corporation's operation and the facts usually reside in a corporate database. The results of inference can be used in a number of ways. One obvious use is to derive additional facts, such as Jenny Fleming being a senior executive. This might trigger additional rules relating, for example, to car parking or use of special lunch facilities. Constraints are also derived, such as her salary being at least $70,000. Another use of the system would be to indicate that such a constraint has been violated, that is, to provide an automatic audit of whether or not company policies are being followed or violated.

Corporate policies are dynamic and change on a continuing basis. The visual language makes it easy to edit the policies, and the inference engine can be used in a differential mode to report the changes resulting from such editing. The separation of the definitions and rules from the facts makes it simple to run "what-if" scenarios in which the implications of potential changes can be analyzed.

## Active Documents

In the past a corporation's rules would typically be encoded in a number of *procedure manuals*, and it is attractive to continue this practice by embedding the formal definitions in the visual language in an active document that enables them to be associated with corresponding text that explains and exemplifies them. The document technology needs to be such that both text and semantic networks can be edited and the networks can be used directly for inference.

We have developed an active document technology that provides users with a word processor supporting typographic text, embedded graphics and other typical word-processing features, together with hypertext linking and embedded active components including the semantic network system already described [10]. Figure 3 shows a screen from an article that was published in the proceedings of the British Computer Society Expert Systems Conference [8]. The top part of the screen shows a typical word processor 'ruler' for tabs, indents, and justification with a page of text below it. The diagram near the top of the page is a semantic network using the same application as that in Figures 1 and 2 but now embedded in the document. It is editable just as if it were in its own window. Double-clicking in a concept such as "general" organization brings up the floating palette shown just below the diagram that provides the same editing features as those previously available at the top of the window.

Twelve semantic networks embedded in the published article together define a solution to a room-allocation problem that was presented as a challenge problem to the knowledge acquisition community [7]. The article was printed as camera-ready copy for the conference proceedings and was also made available in digital form through the Internet, together with the document processor and an interface allow-

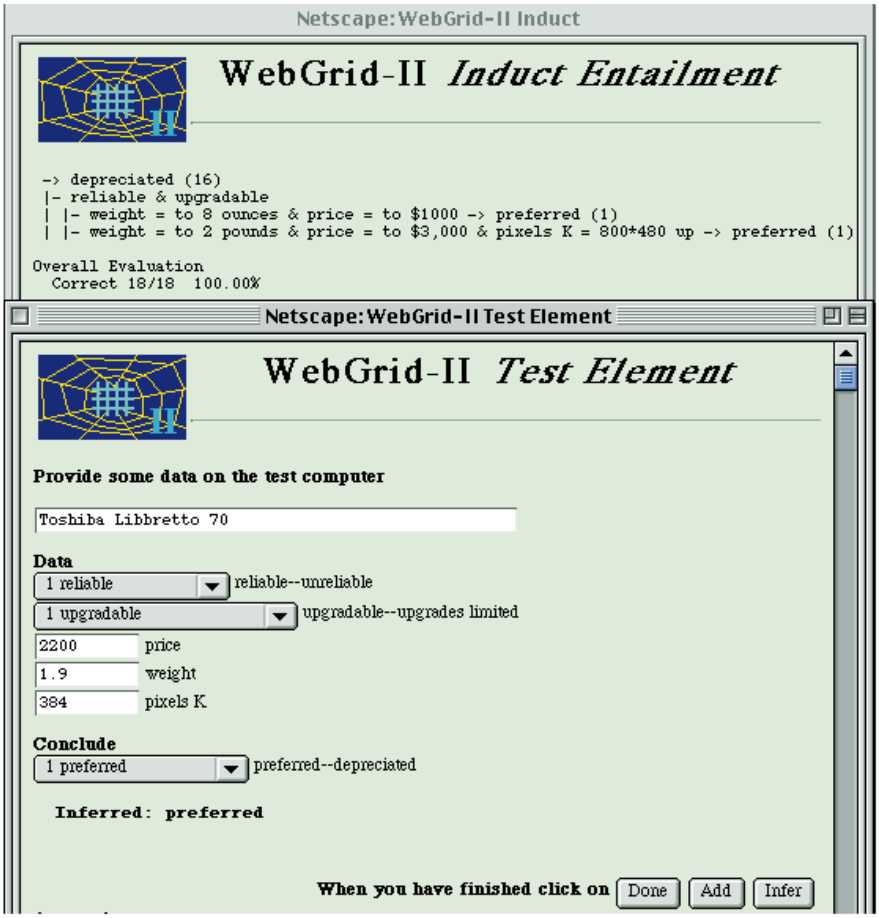


**Figure 5.** Expert system inference through the Web

ing the document to be interrogated as the knowledge base of an expert system solving the room-allocation problem. The window at the bottom right of Figure 3 shows this expert system interface interacting with the semantic networks in the document. The agenda item "secretarial task" has been activated and is recommending that one of two secretaries is allocated to one of two offices. The user selects a person and an office and clicks on the "Allocate" button to allocate the person an office and continue the inference process to derive the next recommendation.

## Knowledge Modeling in Active Documents on the Web

The Web has become a major resource for supporting multimedia active documents and the capability of developing and sharing knowledge structures collaboratively over the Internet further enhances the utility of the technologies discussed in this article. Figure 4 is a screen from a suite of knowledge acquisition applications based on the inference system described operating through the Web [11]. The user is developing corporation procedures for the approval of personal computer purchases. The active document presents multimedia data on three cases and, using the repertory grid triadic elicitation methodology, asks the user to enter a term that distinguishes which document the user wants. In this way knowledge is acquired from experts on the relevant attributes in the domain under investigation together with their values for critical cases to generate a dataset that may be used for inductive knowledge modeling.

The upper screen in Figure 5 shows rules inferred from a set of example cases, that purchases should be depreciated unless the computer is "reliable" and "upgradable" and its "weight" is up "to 8 ounces" and its "price" is up "to $1000" or its "weight" is up "to 2 pounds," its "price" is up "to $3000," and its screen "pixels" are "800*400 up." The inference engine illustrated in the examples in Figures 1 through 3 also operates through the Web as part of the knowledge acquisition system, and the lower screen in Figure 5 shows it being used with the rules on a new case. If the conclusions from the test case are incorrect the user can change them, add the revised case to the set already elicited, and develop a revised knowledge model. As shown previously the system implements open-world reasoning in which it can arrive at definite conclusions from data where some attributes are unknown, and can track which inferences are still open to become more specific with further data.

The screen shown in Figure 5 uses only the user interface widgets available through HTML level 2 forms and these cannot support the interactive graphic drawing necessary for the semantic network applications shown in Figures 1 through 3. However, the semantic network application can be used as an auxiliary server on the Web to dynamically generate the diagrams as clickable images embedded in a document, and to derive inferences from them and return them as Web documents. The concept mapping tool has also been made available as a plug-in for the Netscape browser to support interactive editing of the maps on the Web [12], and it has also been ported to run as a downloadable Java applet on the Web [5].

## Lesson Learned

Symbolic models represented in semantic networks provide a media component for formal knowledge representation with a similar status to media components for informal knowledge representation such as text, diagrams, photographs, and movies. They may be incorporated in documents as active components that can be viewed, edited and linked to other components, and these documents can be used as knowledge bases for expert systems using the symbolic models.

The knowledge-based active document technology has been used in the implementation of a procedures manual for a major corporation, and has a range of practical applications in organizational modeling to provide a convenient way of representing organizational rules and constraints in such a way that they are readily understood, updated and used in the organization's information systems. The technology ports to the Web and integrates naturally with Web-based collaborative information systems.

In terms of lessons learned, we hoped originally that the visual language tools illustrated would be useful to end users in developing symbolic models without requiring expert system analysts or knowledge engineers. However, as with any formal language, the skills of correct expression and of understanding the consequences of making a statement require specialist expertise, and we have found that end users usually need expert support in expressing their knowledge. More positively, they generally readily learn to understand expressions in the visual language terminology which provides an excellent medium for discourse. For example, it is often obvious when relevant concepts have not been used or have been incorporated incorrectly in the model.

We have also found that system modeling in the visual language scales well, and that large and complex models may be built through stepwise refinement of the components of an initial approximate model. C

## Links

WebGrid is accessible at tiger.cpsc.ucalgary.ca, and articles related to the applications described in this article may be found at ksi.cpsc.ucalgary.ca/articles/.

{ Symbolic models represented in semantic networks provide a media component for formal knowledge representation. }

## References


1. Borgida, A., Brachman, R.J., McGuiness, D.L., and Resnick, L.A. CLASSIC: A structural data model for objects. In *Proceedings of 1989 SIGMOD Conference on the Management of Data.* (1989), 58–67.
2. Brachman, R.J. On the epistemological status of semantic nets. In *Associative Networks: Representation and Use of Knowledge by Computers,* N.V. Findler, Ed. Academic Press, NY, 1979, 3–50.
3. Claerbout, J. Electronic documents give reproducible research a new meaning. In *Expanded Abstracts of 62nd Annual International Meeting of Society for Exploratory Geophysics.* (1992), 601–604.
4. Ericsson, K.A. and Smith, J., Eds. *Toward a General Theory of Expertise: Prospects and Limits.* Cambridge University Press, NY, 1991.
5. Flores-Mendez, R.A. Java concept maps for the learning web. In *Proceedings of ED-MEDIA'97: World Conference on Educational Multimedia and Hypermedia.* (Charlottesville, Virginia, 1997).
6. Gaines, B.R. An interactive visual language for term subsumption visual languages. In *IJCAI'91: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence.* (San Mateo, Calif., 1991), 817–823.
7. Gaines, B.R. A situated classification solution of a resource allocation task represented in a visual language. *Int. J. Human-Computer Studies 40*, 2 (1994), 243–271.
8. Gaines, B.R. and Shaw, M.L.G. Documents as expert systems. In *Research and Development in Expert Systems IX: Proceedings of British Computer Society Expert Systems Conference.* M.A. Bramer and R.W. Milne, Eds. Cambridge University Press, UK (1992), 331–349.
9. Gaines, B.R. and Shaw, M.L.G. Integrated knowledge acquisition architectures. *J. Intelligent Info. Syst. 1,* 1 (1992), 9–32.
10. Gaines, B.R. and Shaw, M.L.G. Open architecture multimedia documents. In *Proceedings of ACM Multimedia '93.* (1993), 137–146.
11. Gaines, B.R. and Shaw, M.L.G. Knowledge acquisition, modeling and inference through the World Wide Web. *Int. J. Human-Computer Studies 46*, 6 (1997), 729–759.
12. Kremer, R.A. and Gaines, B.R. Embedded interactive concept maps in web documents. In *Proceedings of WebNet96*, H. Maurer, Ed. (Charlottesville, VA, 1996), 273–280.



**Brian R. Gaines** (gaines@cpsc.ucalgary.ca) is Killam Memorial Research Chair and Director of the Knowledge Science Institute at the University of Calgary in Alberta.

**Mildred L.G. Shaw** (mildred@cpsc.ucalgary.ca) is Industrial Software Engineering Research Chair and Director of the Software Engineering Research Network at the University of Calgary in Alberta.


Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

© 1999 ACM 0002-0782/99/0100 $5.00